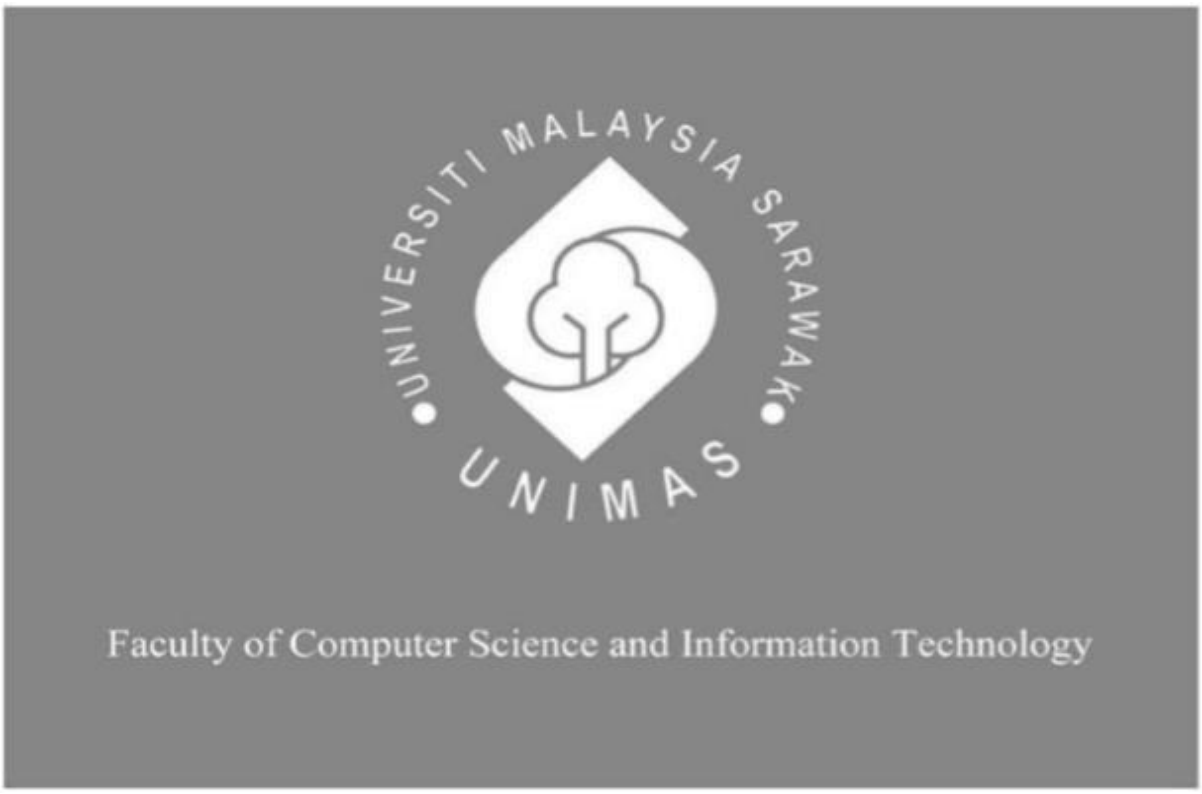


Automatic Data Extraction from Blood Test Report Using Microsoft Azure AI Document Intelligence

Ho Jia Wei

79545

Bachelor of Computer Science with Honours

(Multimedia Computing)

2024

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE Automatic Data Extraction from Blood Test Report Using Microsoft Azure AI Document Intelligence

ACADEMIC SESSION: SESSION 24/25

HO JIA WEI

(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☐	CONFIDENTIAL	(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)
☐	RESTRICTED	(Contains restricted information as dictated by the body or organization where the research was conducted)
☑	UNRESTRICTED	

(AUTHOR'S SIGNATURE)

Permanent Address
No.6 Jalan 4/44,
46050 Petaling Jaya,
Selangor

Date: 23/6/2025

Validated by

(SUPERVISOR'S SIGNATURE)

Ts. Dr Lim Phei Chin
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

Date: 23 June 2025

Note * Thesis refers to PhD, Master, and Bachelor Degree
** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

Table of Content

ACKNOWLEDGMENT

The successful outcome of this project required guidance and assistance from many people, and I am extremely privileged to have got this all along with the accomplishment of my final year project. I would like to take this opportunity to express my heartfelt gratitude to everyone who has supported and guided me throughout my final year project.

I respect and thank Ts. Dr Lim Phei Chin, for giving me an opportunity to do my final year project under her supervision and providing all support and leadership which made me complete the project on time. I am extremely thankful to her for providing such nice guidance despite having a busy schedule giving lectures.

Furthermore, I would like to acknowledge the contributions of my course coordinator, Professor Dr. Wang Yin Chai, Professor of Faculty of Computer Science & Information Technology, Universiti Malaysia Sarawak (UNIMAS), who have been a constant source of encouragement and motivation throughout my project work. His unwavering support and guidance have been a source of encouragement and inspiration.

Lastly, I heartily thank my project examiner, Ts. Dr Sarah Flora Anak Samson Juan, Deputy Dean of Industry and Community Engagement at Faculty of Computer Science & Information Technology, Universiti Malaysia Sarawak (UNIMAS), for her feedback and comments during this project.

ABSTRACT

The healthcare sector often struggles with manual data extraction from blood test reports, which is time-consuming, error-prone, and inefficient. To address this challenge, this study presents the development of a mobile application for automatic text extraction from blood test reports using Microsoft Azure Document AI Intelligence. The application integrates document processing service to streamline the digitization process and enhance data accuracy. The use of Rapid Application Development (RAD) methodology makes the application progresses through iterative stages to ensure user-centered and flexible design. The proposed system uses Azure AI Document Intelligence's customizable models for extracting structured and unstructured data from diverse blood test report formats. Extracted data is validated by users before being stored in an Azure SQL database for seamless data management and retrieval. Additional features include data visualization through tables and line charts and user authentication for security. This application aims to automate the data extraction process, reducing manual workloads and minimizing human error. By standardizing and organizing extracted blood test report data, the system facilitates more accurate diagnoses and efficient healthcare management. Preliminary results indicate the solution’s potential to revolutionize medical record digitization, improve data accessibility, and enhance patient care in healthcare settings.

ABSTRAK

Sektor kesihatan sering menghadapi cabaran dalam proses manual untuk mengekstrak data daripada laporan ujian darah yang tidak cekap. Untuk menangani cabaran ini, kajian ini membentangkan pembangunan aplikasi mudah alih untuk pengekstrakan teks automatik daripada laporan ujian darah menggunakan Microsoft Azure Document AI Intelligence. Aplikasi ini mengintegrasikan perkhidmatan pemprosesan dokumen untuk mempermudah proses pendigitalan dan meningkatkan ketepatan data. Penggunaan metodologi Rapid Application Development (RAD) membolehkan aplikasi ini untuk melalui fasa iteratif bagi memastikan reka bentuk yang fleksibel dan mengutamakan keperluan pengguna. Aplikasi ini dapat menggunakan model yang boleh diubahkan oleh Azure AI Document Intelligence untuk mengekstrak data berstruktur dan tidak berstruktur daripada pelbagai format laporan ujian darah. Data yang diekstrak akan disahkan oleh pengguna sebelum disimpan dalam pangkalan data Azure SQL bagi pengurusan dan pemulihan data yang lancar. Ciri-ciri tambahan aplikasi ini termasuklah visualisasi data melalui jadual dan carta garis serta pengesahan pengguna untuk keselamatan. Di samping itu, aplikasi ini bertujuan mengautomasi proses pengekstrakan data, mengurangkan beban kerja manual, dan meminimumkan kesilapan manusia. Dengan menyeragamkan dan menyusun data laporan ujian darah yang diekstrak, sistem ini memudahkan diagnosis yang lebih tepat dan pengurusan penjagaan kesihatan yang lebih cekap. Hasil awal menunjukkan potensi penyelesaian ini untuk merevolusikan pendigitalan rekod perubatan, meningkatkan kebolehcapaian data dan memperkukuh penjagaan pesakit dalam persekitaran penjagaan kesihatan.

List of Tables

List of Figures

List of Abbreviations

RAD	Rapid Application Development
EMR	Electronic Medical Records
ML	Machine Learning
OCR	Optical Character Recognition
NLP	Natural Language Processing
WER	Word Error Rates
CER	Character Error Rates
NER	Named-Entity Recognition
ESR	Electronic Source Record
CNN	Convolutional Neural Networks
IDP	Intelligent Document Processing
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
URL	Uniform Resource Locator
IDE	Integrated Development Environment
API	Application Programming Interface

Chapter 1: Introduction

1.1 Background

Every health care organization deal with form-like documents such as medical report in daily workflows. These documents recorded the result of the conducted health test and the details such as patient personal information, health test laboratory name, and the health parameters. According to Cowie et al. (2017), medical reports are used for tracking the patient's health condition and provide patient care enhancement. Hence, many health care organizations had collected medical reports from their patients for the doctors to have a quick review on their patients' health conditions. These medical reports are from various test laboratories which have different layout and writing structure. The unstandardized medical report format had caused a huge challenge to the digitization of collected medical reports as it requires more complicated programming algorithm in data identification. This had burdened the clinicians as manual data extraction process are labour intensive and highly time-consuming (Gokhale et al., 2021). The different formats and layout will result in higher human error during the manual data extraction.

As the process of transforming unstructured data into structured data was unavoidable to create electronic medical records (EMR), the idea of automatic blood test data extraction using document processing service was proposed. Currently, there are several available document processing services provided by well-known technology company such as Microsoft Azure, Amazone Web Services and Google, which applied machine learning to train models for data extraction. Based on Xu et al. (2021), they had noted that these document processing services had leveraged technology such as machine learning (ML) and optical character recognition (OCR) to streamline the data extraction process.

The application of OCR techniques in data extraction has revolutionized how unstructured data is processed and managed in healthcare. OCR allows for the automated reading and

interpretation of text from images and scanned documents which had reduced the need of manual input. As highlighted by Subramani et al. (2020), OCR plays a critical role in transforming non-standardized medical documents into machine-readable formats to enable seamless digitization of medical data extraction. Furthermore, OCR-powered systems can identify and extract key details from diverse medical report layouts regardless of variations in structure and design.

The advancements in OCR technology had enhanced its capabilities in recognizing and categorizing health metrics accurately. The integration of natural language processing (NLP) and ML in OCR solutions are equipped with features such as the ability to handle poor-quality images, making them suitable for use in real-world medical scenarios. These advancements had addressed challenges such as errors caused by inconsistent formatting by ensuring higher accuracy in data extraction. By leveraging document processing service in healthcare, clinicians can save time, minimize human errors, and focus more on patient care. Thus, document processing services offers a promising solution for automating data extraction and digitizing medical records efficiently. Review of related work will be done in Chapter 2 to provide a deeper understanding of the use of document processing services in medical data extraction.

1.2 Problem Statement

The healthcare industry continues to face significant challenges in digitizing medical reports to establish comprehensive and reliable Electronic Medical Records (EMR) (Kaihlanen et al., 2020; Melnick et al., 2020). While various solutions have been developed to assist clinicians and nurses in recording patient information, the necessity for manual data entry persists as a major barrier to the widespread adoption of these solutions. This dependency on

manual processes prolongs the digitization workflow and introduces a higher likelihood of human errors. These errors undermine the accuracy and reliability of patient records, resulting in inconsistencies and incomplete data that can hinder effective patient care and decision-making (Holmes et al., 2021). Without addressing these issues, healthcare providers risk compromising the quality and safety of their services.

As noted by Holmes et al. (2021), manual data entry is labour-intensive and especially error-prone in scenarios involving high patient volumes or intricate medical information. The inaccurate recordings of critical health parameters could result in serious repercussions including incorrect diagnoses or inappropriate treatments, ultimately compromise patient safety. Moreover, healthcare organizations often receive medical reports from multiple laboratories, each employing different formats, layouts, and terminologies. For example, Figure 1.1 and Figure 1.2 had illustrated the haematology test results from two different laboratories which are Pantai Premier Pathology and B.P. Clinical Lab. These two test reports had demonstrated the variability in their layouts and formats. This diversity had further complicated the data entry process then created additional burdens for clinicians and administrators who must ensure accuracy and standardization while processing these varied formats.

HAEMATOLOGY

Full Blood Count				
HAEMOGLOBIN	血红蛋白	14.2	g/dL	(13.0- 18.0)
Red Blood Cell	红血细胞	4.70	x10^12/L	(4.50- 6.50)
RDW	红细胞分布宽度	12.6	%	(<14.3)
PCV	红细胞压积	45	%	(40 - 54)
MCV	平均细胞体积	96	fL	(76 - 96)
MCH	平均细胞血红蛋白	30	pg	(28 - 34)
MCHC	平均红细胞血红蛋白浓度	32	g/dL	(30 - 36)
WHITE BLOOD CELL	白血细胞	7.6	x10^9/L	(4.0 - 11.0)

Figure 1.1: Haematology Test Result of Pantai Premier Pathology

HAEMATOLOGY EXAMINATION	(血液檢查)			
Total RBC	(紅血球計數)	: 5.3	x10^12 /L	(M 4.5-6.0 F 4.0-5.5)
Haemoglobin (Hb)	(血紅素)	: 110	gm/L	(M 125-175 F 115-155)
PCV	(血球比容量)	: 0.36	L	(M 0.40 - 0.50 F 0.37 - 0.45)
MCV	(平均紅血球容積)	: 68	fl	(82-98)
MCH	(平均紅血球血色蛋白)	: 21	pg	(27-33)
MCHC	(平均紅血球血色蛋白濃度)	: 310	g/L	(310 - 350)
RDW	(紅血球體積分布寬度)	: 16.2	%	(11.0 - 16.0)
Total WBC	(白血球計數)	: 9.8	x10^9 /L	(4-11)

Figure 1.2: Haematology Test Result of B.P. Clinical Lab

Although digitization tools leveraging Optical Character Recognition (OCR) and Machine Learning (ML) technologies have emerged, their adoption in the healthcare sector remains limited. Many existing solutions are lack of flexibility to adapt to the diverse formats of medical reports and often exhibit inconsistencies in data extraction accuracy. These shortcomings are particularly pronounced when dealing with complex medical terminologies or the conversion of measurement units in blood test reports. Such limitations underscore the urgent need for automated solutions capable of efficiently extracting structured data from unstructured medical documents while maintaining high levels of accuracy and reliability.

Given the critical importance of accurate medical records in ensuring quality healthcare, there is a pressing need for an automatic blood test data extraction solution that leverages document processing services. The proposed project aims to address this need by developing an application capable of automating the extraction of raw text from blood test reports using OCR technology and a trained ML model. This raw text will then be processed through specialized algorithms to identify and organize key health metrics such as Platelet Count, Hematocrit Value, and other vital parameters. By incorporating these advanced technologies, the application will streamline the digitization process, reduce manual workload, and improve data consistency, thereby offering healthcare organizations a reliable tool for managing and analyzing blood test data effectively. This innovation has the potential to enhance patient care

and contribute to more efficient healthcare operations while addressing the existing gaps in medical data management.

1.3 Scope

The mobile application of automatic data extraction from blood test report is specifically designed to handle blood test reports written in the English language. This focus ensures precise extraction and accurate interpretation of text, given the challenges associated with multilingual document processing. The application's primary functionality revolves around extracting data from haematology sections of blood test reports, where it will identify and retrieve health metrics or parameters as specified in **Appendix A: Blood Test Report Health Metrics.** By narrowing the scope to haematology, the project aims to refine its functionality and deliver accurate results in this critical area of medical diagnostics. The application also aims to demonstrate its capability as a proof of concept for scalable solutions that are potentially expanding to other test types in the future.

Furthermore, the mobile application is built to depend entirely on services provided by advanced document processing platforms such as Microsoft Azure. These services will facilitate efficient data extraction and structured data generation using OCR and ML capabilities. The reliance on Microsoft Azure ensures that the application can leverage its robust algorithms for key-value pair extraction to handle complex medical reports.

1.4 Aims and Objectives

The overall objectives of this project are as follows:

1. To study current document processing techniques.

2. To implement document processing techniques on the mobile application development of automatic data extraction from blood test report.
3. To evaluate the application performance using Character Error Rate and Confidence Score calculated by Microsoft Azure Document Intelligence.

1.5 Significance of Project

The significance of this project lies in addressing the challenge in the healthcare sector which is efficiently digitizing blood test reports. By leveraging state-of-the-art document processing services such as Microsoft Azure Document Intelligence, Amazon Textract, and Google Document AI, the proposed mobile application will offer a robust solution for automating data extraction from blood test reports. These services had powered by many data extraction techniques such as advanced machine learning algorithms and optical character recognition (OCR) technologies to provide the foundation for accurately identifying and extracting essential health metrics from blood test report including the blood cell counts, haemoglobin levels, and platelet counts.

This mobile application aims to eliminate the time-consuming manual input process that also prone to human errors, thereby improving the reliability of digitized blood test records. The extracted data will be standardized into a uniform format to provide clean report for data visualisation. This feature is particularly significant for healthcare providers who need to analyze patient health trends or compile data for research purposes. Additionally, standardization will enable clinicians and researchers to draw more precise conclusions and insights from the extracted data, contributing to improved diagnostic and treatment outcomes.

Moreover, the performance of the application will be evaluated using metrics such as CER and Confidence Score, which provides a quantitative measure of the accuracy and

reliability of the extracted data. By ensuring high confidence scores with low CER, the project addresses critical concerns regarding the usability and trustworthiness of automated data extraction systems in medical settings. Ultimately, this project holds the potential to revolutionize how medical data is processed, contributing to advancements in healthcare delivery, research, and patient management.

1.6 Expected Outcome

The outcome of this project is the development of a functional mobile application prototype that significantly enhances the efficiency and accuracy of medical data extraction processes. This application will be designed to handle diverse formats of blood test reports, including scanned images and PDF files. By utilizing the document processing service, the application will extract textual content with high precision to ensure that medical terminologies and abbreviations are correctly interpreted and categorized.

Once the textual data is extracted, the application will process and organize it into structured formats, such as tables, which is suitable for further analysis. This structured format is intended to facilitate seamless integration with analytical tools, enabling healthcare providers to make data-driven decisions efficiently. The categorization of extracted data will include flagging critical conditions. Thus, this application will be a crucial tool for clinicians.

Chapter 2: Literature Review

2.1 Background

Chapter 2 provides a comprehensive review of the literature relevant to the development of an automated system for Blood Test Data Extraction. This chapter explores the foundational technologies, current data extraction techniques, and existing applications that underpin the proposed solution. It is structured into several sections where each section addressing critical aspects of the project and establishing the groundwork for the proposed methodology.

The chapter starts with Section 2.1, which outlines the scope and organization of the literature review. This section serves as a roadmap for summarizing the topics covered in subsequent sections and their relevance to the project objectives. Section 2.2 describes the review of related work by examining the core technologies essential to medical data extraction. The advancements in Natural Language Processing (NLP) by highlighting the techniques and tools used for processing unstructured text in healthcare industry will be discuss. Besides, this section will also focus on the Optical Character Recognition (OCR) technique which provides insights into its evolution and application in document digitization. The hybrid approaches that combine NLP, OCR and other data extraction techniques to enhance data extraction accuracy and efficiency will be address.

Section 2.3 presents a review of current data extraction techniques. This section evaluates various methodologies including rule-based systems, machine learning models, and deep learning approaches. The strengths and limitations of each reviewed technique for data extraction will be emphasized in this section. Section 2.4 reviews existing data extraction applications to understand their functionality and limitations. Subsections 2.4.1 to 2.4.5 examine popular applications such as Microsoft Lens, Adobe Scan, TapScanner, Doc Scanner,

and CamScanner. Each application is analyzed for its performance and feature set particularly in the context of data extraction.

Section 2.5 provides a critical discussion based on the findings from Sections 2.3 and 2.4. Subsection 2.5.1 evaluates current data extraction tools by identifying gaps and opportunities for innovation. Subsection 2.5.2 examines the application features of existing data extraction applications by assessing their relevance to the basic functions needed for image to text data extraction and identifying areas for improvement in automated data extraction.

Finally, Section 2.6 summarizes the insights gained from the literature review and introduces the proposed solution. This section highlights the capabilities of the proposed solution including document processing service which powered by the integration of advanced OCR, NLP and other data extraction techniques, and its potential to streamline blood test data digitization.

2.2 Review of Related Work

Kormilitzin et al. (2020) had introduced Med7, a clinical named-entity recognition (NER) model designed to extract medication-related entities (e.g., drug names, dosage, and frequency) from unstructured electronic health records (EHRs). The model achieved high accuracy with a lenient F1 score of 0.957 when pre-trained on MIMIC-III data and demonstrated improved transferability to UK-based mental health records after fine-tuning (F1 = 0.944). This study highlighted the importance of domain adaptation in clinical NLP models for achieving robust performance across diverse healthcare datasets. However, the categories which are naturally underrepresented had brings impact to the accuracy of the NER model. The skewed medical records had made it more challenging to train a robust model to accurately identify the annotated entities.

Karthikeyan et al. (2022) proposed an OCR post-correction methodology leveraging the RoBERTa deep learning model to enhance text accuracy in scanned medical documents, addressing issues such as illegibility and domain-specific terminology. By integrating domain-specific entity filtering and spell checkers, their approach significantly reduced word error rates (WER) and character error rates (CER) on National Health Service (NHS) medical reports and publicly available datasets, achieving improved performance without domain-specific training. This study highlights the robustness of transformer-based models in OCR post-processing, demonstrating their effectiveness for real-world medical text digitization tasks. They also noted the important of using post-processing techniques to improve the quality of the OCR text. However, these post-processing methods also limited the performance as they require expensive resources.

Hsu et al. (2022) had developed a deep learning-based NLP data pipeline to extract critical values for sleep Apnea diagnosis (Apnea Hypopnea Index, AHI and oxygen saturation, SaO2) from 955 scanned sleep study reports in EHR. Their approach had integrated image preprocessing, OCR, and advanced NLP models to demonstrate that ClinicalBERT had achieved the highest document accuracy of 94.76% for AHI and 91.61% for SaO2. This study had highlighted the significance of document layout and structured input by setting a benchmark for efficient information extraction from scanned EHR documents. However, they had noticed that the text segmentation process in ClinicalBERT only captures right and left word sequences around the candidate numbers, but the table column names were not captured due to the limited research about processing natural language in tables.

Wang et al. (2022) evaluated an electronic source (eSource) record (ESR) system for real-world ophthalmology studies. This system focusing on its data transcription efficiency and quality compared to traditional manual methods. Their findings demonstrated that the ESR system reduced transcription time by 81.8% and improved accuracy to 98.17%. The high

accuracy from the result had highlighted its potential to replace manual transcription in clinical research. The study also emphasized the importance of integrating standardized data models and NLP to enhance usability and data quality. Besides, they had noticed the important of data collection method used in prospective research based on research data standards as it will directly affect the quality of the data and might lead to information lost.

Hassan et al. (2021) proposed a mobile application that uses machine learning techniques, including convolutional neural networks (CNN) and optical character recognition (OCR), to recognize and digitize handwritten medicine names and doses from doctors' prescriptions. Their approach had shown the training accuracy resulted of 73% and the testing accuracy resulted of 50% over 50 epochs and 35 batches using cross-validation. The system aims to address the challenge of handwritten recognition by increasing the accuracy in prescribed medicines identification. However, they mentioned that relatively small dataset and low-quality scanned images will obstruct the accuracy for both training and testing purpose.

Dash (2021) proposed a hybrid framework integrating Optical Character Recognition (OCR) and Natural Language Processing (NLP) for extracting meaningful information from unstructured data, such as PDFs and text documents. By employing Spark-based OCR engines for text conversion and NLP for post-processing, the system improved data accuracy and quality by over 42%, effectively addressing errors like misspellings, deletions, and insertions. The study demonstrates the potential of this approach for enterprise-level document processing with future enhancements focusing on integrating domain-specific vocabularies to further optimize the accuracy.

2.2 Review on Current Data Extraction Tools

Nandhinee et al. (2022) introduced DEXTER, an end-to-end system designed to extract tabular data from medical documents such as Electronic Health Records (EHR) and Explanation of Benefits (EOB). The system employs a two-stage transfer learning approach with CDeC-Net for table detection, a computer vision-based table type classification and cell detection, and an OCR engine for cell content extraction. Experimental results on a curated dataset (Meddata) demonstrate that DEXTER outperforms existing tools like Amazon Textract and Microsoft Azure Form Recognizer in table detection and data extraction accuracy, especially for complex tabular layouts like borderless or partially bordered tables.

Xu et al. (2021) compared three information extraction (IE) solutions which includes Amazon Textract, Microsoft Azure Form Recognizer, and DocParser by highlighting their strengths and limitations. While template-free approaches like Textract and Form Recognizer are easier to use and require less setup, they struggle with layout variations and semi-structured forms, which account for 80% of business documents. Template-based solutions, such as DocParser, offer higher accuracy for complex forms but at the cost of significant manual effort, emphasizing the trade-offs between automation and precision in IE technologies. The study highlights the trade-offs between convenience and accuracy in existing IE systems, emphasizing the need for balanced approaches to effectively manage diverse form layouts in real-world scenarios.

Li (2024) conducted a comparative analysis of four OCR engines including Tesseract, Keras, Paddle, and Azure OCR using datasets with varied challenges such as handwriting and complex document layouts. By integrating these systems into a hybrid framework, the study achieved superior recognition performance, with Azure OCR excelling in precision and adaptability across diverse text conditions. This research highlights the advantages of

combining strengths from multiple OCR engines and suggests preprocessing techniques like edge detection and thresholding for improved accuracy in real-world applications.

2.4 Review on Existing Data Extraction Applications

The following data extraction applications are selected from the Google Play Store using search query of 'image to text'. They are the top 5 applications shown in the search result that had fulfilled the criteria of 50M+ downloads and at least 4.5 score in rating.

2.4.1 Microsoft Lens

Figure 2.1: Application Icon of Microsoft Lens

Microsoft Lens is a pocket PDF scanner with integrated OCR offered by Microsoft Corporation which have more than 50 million of downloads and 4.9/5.0-star rating on Google Play Store (Microsoft Corporation, 2015). Figure 2.1 shows the icon of Microsoft Lens retrieve from Google Play Store. The users will need to sign in using their business, school or own Microsoft account to start this application. This mobile application is used to trim, enhance, and make pictures of whiteboards and documents readable. The users can convert images to PDF, Word, PowerPoint, and Excel files. For OCR of scanned documents, printed and handwritten text in English or printed text in other languages can be extracted and save to OneNote, OneDrive, or their local device in Word files. This application also

supports table extraction as shown in Figure 2.2. Besides, it also supports importing existing images using Gallery. This application is completely free to use without any advertisement.

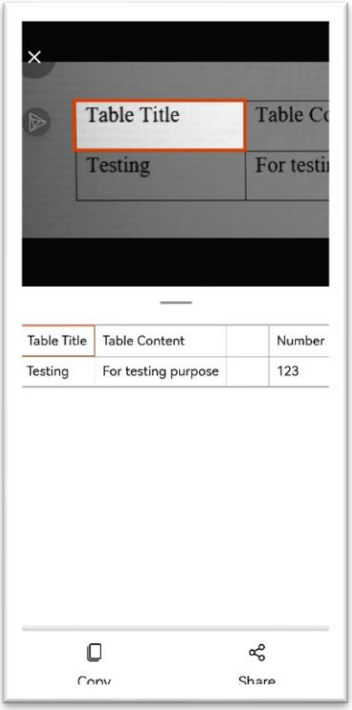


Figure 2.2: Table Extraction of Microsoft Lens

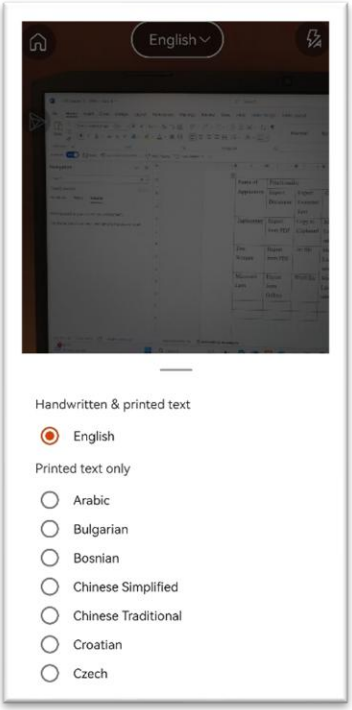


Figure 2.3: OCR Language Selection of Microsoft Lens

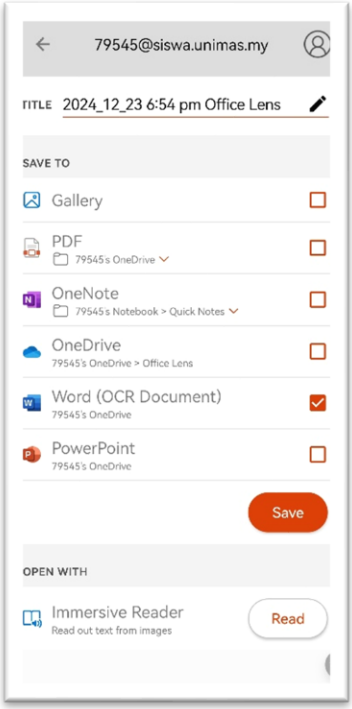


Figure 2.4: Microsoft Lens Text Extraction Export as Word File

2.4.2 Adobe Scan

Figure 2.5: Application Icon of Adobe Scan

Adobe Scan is a free mobile scanner to convert photos and documents into PDF and JPEG files offered by Adobe which have more than 50 million of downloads and 4.8/5.0-star rating on Google Play Store (Adobe, 2017). Figure 2.5 shows the icon of Adobe Scan retrieved from Google Play Store. The scanned image can be saved and export in the form of PDF file after image pre-processing and text editing using this application showing in Figure 2.4. This application will convert image into editable text and indicates it with a fine border, meanwhile the text that are not identified are not editable, showing in Figure 2.5. The users are allowed to search, select, and copy text from the PDF file. It can also save each scan to Adobe Document Cloud for instant access and sharing. This application offers subscriptions for premium features such as export PDFs to Microsoft Word or PowerPoint file formats and increase the OCR capacity from 25 to 100 pages.

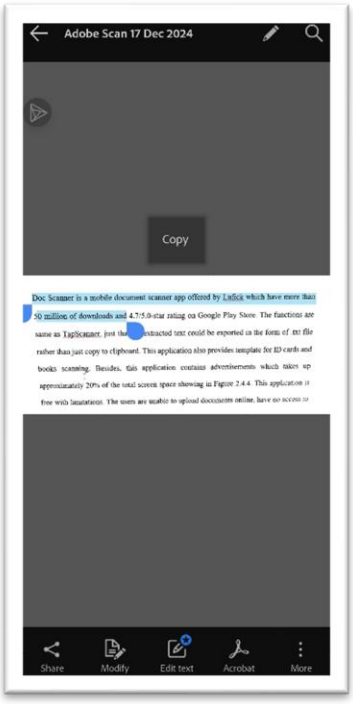


Figure 2.4: Text Extraction of Adobe Scan

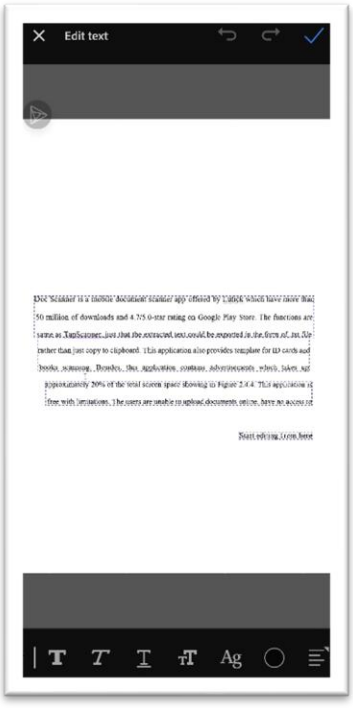


Figure 2.5: OCR Result Edit Feature of Adobe Scan

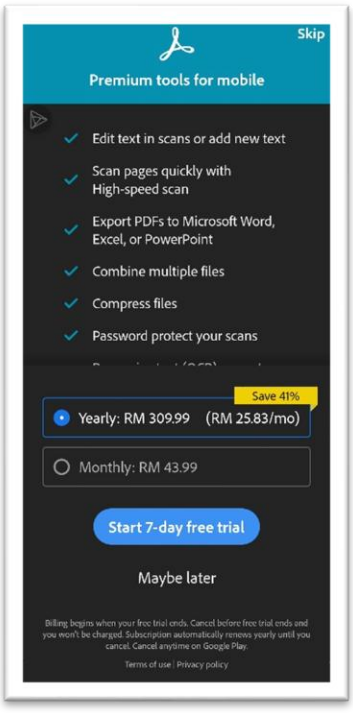


Figure 2.6: Premium Subscription of Adobe Scan

2.4.3 TapScanner

Figure 2.7: Application Icon of TapScanner

TapScanner is a mobile document scanner application offered by Tap AI which have more than 100 million of downloads and 4.8/5.0-star rating on Google Play Store (Tap AI, 2017). Figure 2.7 shows the icon of TapScanner retrieved from Google Play Store. It allows users to scan any document including receipts, documents, business cards and photos, then convert them to PDF immediately. This application also provides template for ID cards and passport scanning showing in Figure 2.8. It also allows users to import PDF for further actions. Besides, it provides several filters to enhance the scanned documents, make them look cleaner and more professional by removing the shadows and artifacts. The users will need to select a specific language for the OCR text extraction and the extraction result was shown at Figure 2.9. The extracted text could only be copied to clipboard for sharing. Figure 2.10 had shown that this application requires the users to pay for premium subscriptions after the 3-days free trial.

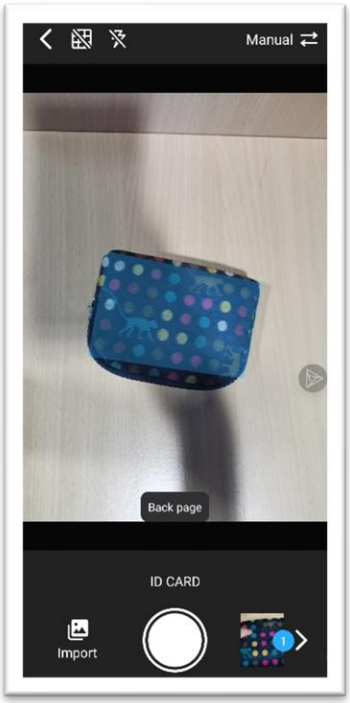


Figure 2.8: IC Card Data Extraction Template of TapScanner

Figure 2.9: OCR Result of TapScanner

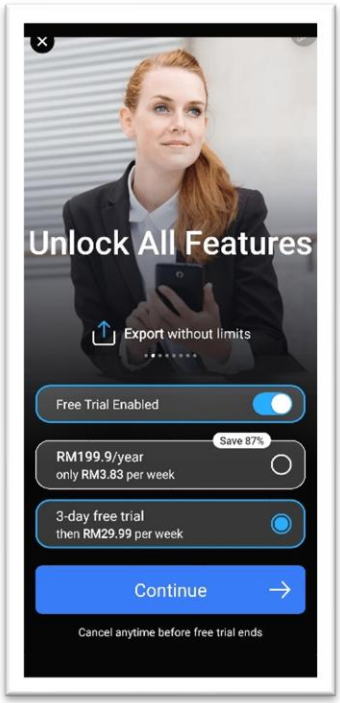


Figure 2.10: Premium Subscription of TapScanner

2.4.4 Doc Scanner

Figure 2.11: Application Icon of Doc Scanner

Doc Scanner is a mobile document scanner app offered by Lufick which have more than 50 million of downloads and 4.7/5.0-star rating on Google Play Store (Lufick Technology Private Limited, 2017). Figure 2.11 shows the icon of Doc Scanner retrieved from Google Play Store. The functions are same as TapScanner, just that the extracted text could be

exported in the form of .txt file rather than just copy to clipboard. This application also provides template for ID cards and books scanning. Besides, this application contains advertisements which takes up approximately 20% of the total screen space showing in Figure 2.12. This application is free with limitations. The users are unable to upload documents online, have no access towards OCR of scanned documents, and have limitations on document scanning without premium subscription.

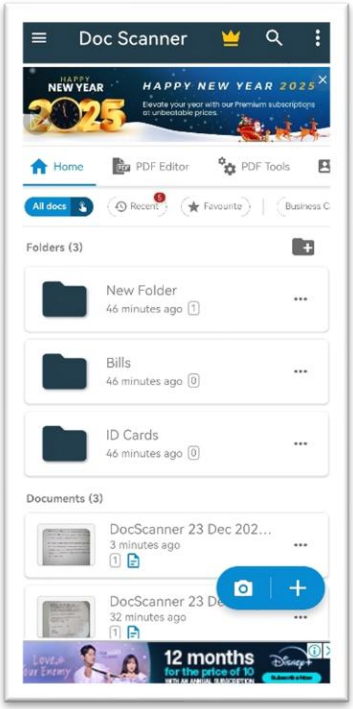


Figure 2.12: Advertisements on Home Page of Doc Scanner

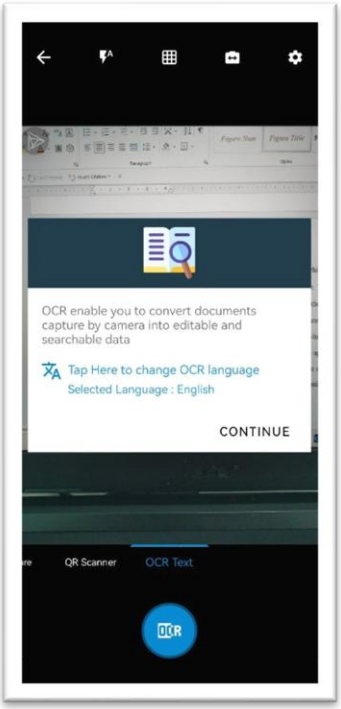


Figure 2.13: OCR Select Language Page of Doc Scanner

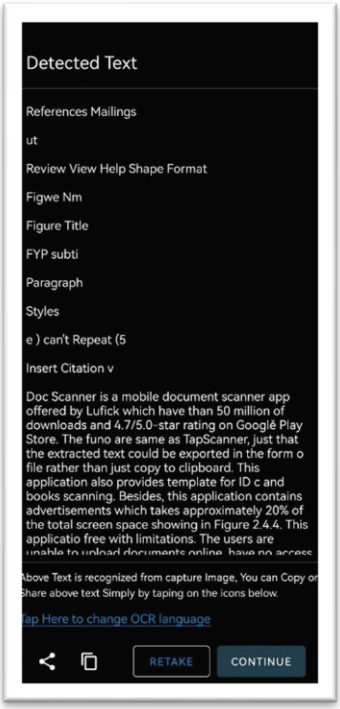


Figure 2.14: OCR Result of Doc Scanner

2.4.5 CamScanner

Figure 2.15: Application Icon of CamScanner

CamScanner is a scanner app offered by CamSoft Information which have more than 100 million of downloads and 4.6/5.0-star rating on Google Play Store (CamSoft Information, 2010). Figure 2.15 shows the icon of CamScanner retrieved from Google Play Store. This application helps users scan, edit, store, and sync contents across smartphones, iPads, tablets and computers. It also supports data extraction with OCR showing in Figure 2.17 and convert PDF to Word, Excel, etc. The users can share the extracted text as Word or Text files as shown as Figure 2.18. Besides, they can also countercheck the extraction result using the Proofread function showing in Figure 2.19.

Figure 2.16: Image Pre-processing of CamScanner

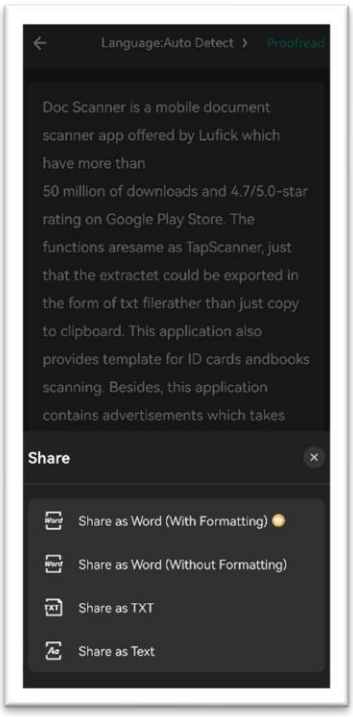


Figure 2.17: OCR Result Share Feature of CamScanner

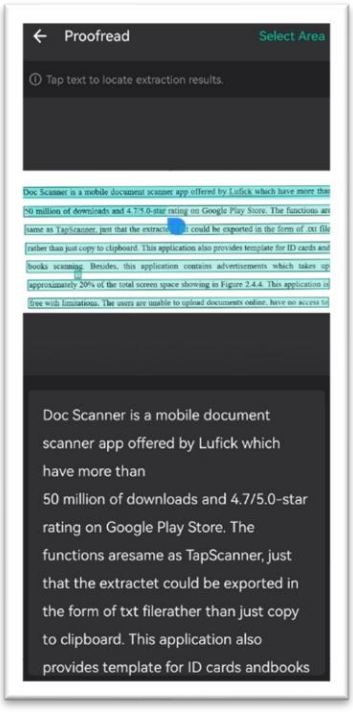


Figure 2.18: Proofhead Feature of CamScanner

2.5 Discussion

2.5.1 Discussion on Data Extraction Tools

Table 2.1: Discussion of Reviewed Data Extraction Tools

Data Extraction Tool	**Data Extraction Feature**				Fee
	Data Training Model	Internal Data Labelling Tool	Table Detection	Key-Value Pair	
Amazon Textract (Amazon Web Services, 2019)	Custom model	Included	Available	Support	Subscription needed (3 months trial)
Microsoft Azure Document Intelligence (Microsoft Azure, 2021)	Custom model	Included	Available	Support	Subscription needed (12 months trial)
DocParser (SureSwift Capital Inc., 2018)	Template-based	Included	Available	Support	Subscription needed (14 days trial)
Tesseract (Shafait & Smith, 2010)	Custom model	Not included	N/A	Does not support	Free of charge
Keras (Fausto, 2019)	Custom model	Not included	N/A	Does not support	Free of charge
PaddleOCR (PaddlePaddle, 2024)	Custom model	Not included	Available	Does not support	Free of charge

Table 2.1 had concluded the discussion result of the 6 reviewed data extraction tools including Amazon Textract, Microsoft Azure AI Document Intelligence, DocParser, PaddleOCR, Tesseract, and Keras. All the mentioned data extraction tools support custom data training except for DocParser as it is a template-based tool designed for structured document processing. Tesseract, Keras and PaddleOCR does not support Key-Value Pair, and they require external tool for data labelling to add bounding box annotations and export

them in JSON or in another compatible format. Tesseract and Keras are both weak in table detection as they do not natively support structure recognition meaning that they can recognize text within table cells but does not detect rows, columns, or cell boundaries. Amazon Textract, Microsoft Azure AI Document Intelligence, and DocParser are subscription-based application whereas Tesseract, Keras and PaddleOCR are completely free to use. Amazon Textract offers free trial up to 3 months, Microsoft Azure AI Document Intelligence offers a 12-months free trial, meanwhile DocParser only offers a free trial for 14 days.

2.5.2 Discussion on Data Extraction Applications

Table 2.2: Discussion of Reviewed Data Extraction Applications

Name of Data Extraction Application	**Application Feature**					Advertisement	Fee
	Import Document	Export Extracted Text	OCR	Upload Document Online	Image Pre-processing		
Microsoft Lens (Microsoft Corporation, 2015)	Import as Image	Word file	Manual Language Selection	Available	Available	Contain Ads	Free of Charge
Adobe Scan (Adobe, 2017)	Import as Image	PDF file	No Language Selection	Available	Available	Ads-free	Free with in-app-purchase
TapScanner (Tap AI, 2017)	Import as PDF or Image	N/A	Manual Language Selection	N/A	Available	Contain Ads	Subscription needed
Doc Scanner (Lufick Technology Private Limited, 2017)	Import as PDF or Image	Text file	Manual Language Selection	Available	Available	Contain Ads	Subscription needed

CamScanner (CamSoft Information, 2010)	Import as PDF or Image	PDF file and Word file	No Language Selection	N/A	Available	Contain Ads	Free with in-app-purchase

Table 2.2 had concluded the discussion result of reviewed application functionality for Microsoft Lens, Adobe Scan, TapScanner, Doc Scanner, and CamScanner. Microsoft Lens and Adobe Scan only supports imports as image meanwhile the other 3 applications can import as both image and PDF file. All reviewed applications are available to export extracted text in the form of Word, Text or PDF file, except for TapScanner which only allows users to manually copy and paste to share the extracted text. Microsoft Lens, TapScanner, and Doc Scanner requires the users to select a specific language for data extraction meanwhile Adobe Scan and CamScanner offers automatic language detection for data extraction. Microsoft Lens, Adobe Scan, and Doc Scanner allows users to sync their extracted documents on virtual database such as OneDrive and Cloud, which had increased the accessibility of the digitized documents compared to TapScanner and CamScanner which only allows users to save their work on local devices. All of the reviewed applications include image pre-processing for clear and optimal data screening. They provide various image filters and image refining functions such as crop, rotate and shear to enhance the image quality for optimal data extraction. Move on to the monetary aspects, Microsoft Lens is the only application that is completely free to use and without advertisement, while Adobe Scan is a freemium application without advertisement and CamScanner is a freemium application with advertisement. The other 2 applications are subscription-based application with advertisement embedded in it.

2.6 Summary

This chapter reviewed the data extraction techniques and applications that are relevant to the development of an automated blood test data extraction system. It highlighted key insights from existing work and emphasized the potential of using advanced technologies like OCR and NLP for digitizing medical data. These studies also showcased the effectiveness of hybrid approaches combining OCR and NLP, which improve accuracy and handle complex document layouts. However, limitations such as high resource requirements and challenges in processing structured data were noted.

Current data extraction applications such as Microsoft Lens, Adobe Scan, and Azure Document Intelligence were evaluated for their functionalities, efficiency, and usability. Most tools exhibited strong OCR capabilities but lacked in seamless integration of advanced NLP for context-specific data extraction. These findings revealed the gaps in existing solutions especially in handling diverse layouts and ensuring high accuracy in healthcare contexts.

To address these challenges, the proposed solution leverages Azure AI Document Intelligence. It is a robust framework that integrates cutting-edge OCR and NLP techniques. This platform offers advanced capabilities such as custom model training, table detection, and key-value pair recognition, making it well-suited for extracting structured data from complex blood test reports. By streamlining data digitization, Azure AI Document Intelligence has the potential to significantly enhance the efficiency and accuracy of blood test data record handling and reducing reliance on manual processes.

Chapter 3: Methodology

3.1 Background

This chapter will describe the methodology used for this project in detail with the explanation of each development phases. The proposed title, Automatic Data Extraction from Blood Test Report using Microsoft Azure AI Document Intelligence requires a suitable methodology to monitor and manage the project development. The selected methodology for the project implementation is Rapid Application Development model (RAD). RAD was selected due to its flexibility development cycles and the relatively short resource planning time.

The main idea of this chapter is to discuss the requirement analysis and the design of the proposed application. The Business Model phase is the first phase in RAD to identify the users and application requirements. Data Modelling phase will focus on the definition of the collected requirements to further identify the needs of the users. Move on to the Process Modelling phase, it provides a complete and clear overview of the application by illustrating flow charts and data flow diagram. The application will be developed and tested referring to the proposed design layout.

3.2 Architecture

Document processing solutions are rapidly evolving to meet the demands of diverse industries including healthcare. Intelligent Document Processing (IDP) systems such as Azure AI Document Intelligence had represented the advancements over traditional OCR (Kocbek et al, 2021). Unlike template-based systems, IDP solutions dynamically adapt to unstructured and semi-structured data formats. Azure's AI Document Intelligence offers both prebuilt and customizable models which enables the targeted extraction of key-value pairs, tables, and unstructured text from diverse document types without manual template creation.

Microsoft Azure AI Document Intelligence provides both prebuilt (template-based) and customizable models. For the template-based models, these models are prebuilt to handle common document types, such as invoices, bank statements and ID documents. They require minimal setup but may struggle with unique or non-standardized formats. Meanwhile for custom models, these models can be trained using as few as five documents to adapt specifically to unique document structures. Custom models are excelling in extracting domain-specific data where predefined templates fail to account for variability in layouts and terminologies.

Figure 3.1 had shown the workflow of automatic data extraction from blood test report using Microsoft Azure AI Document Intelligence. The collected dataset which are blood test report from various clinic or laboratories will be labelled into various classes using Azure AI Document Intelligence Studio based on the blood test metrics identified in Appendix A. The output of the training creates the model. The model will be transformed and embedded into the mobile application development and will be used for data extraction of the grey-scaled input image to produce output in text format. The output text will then be saved into the database after text validation.

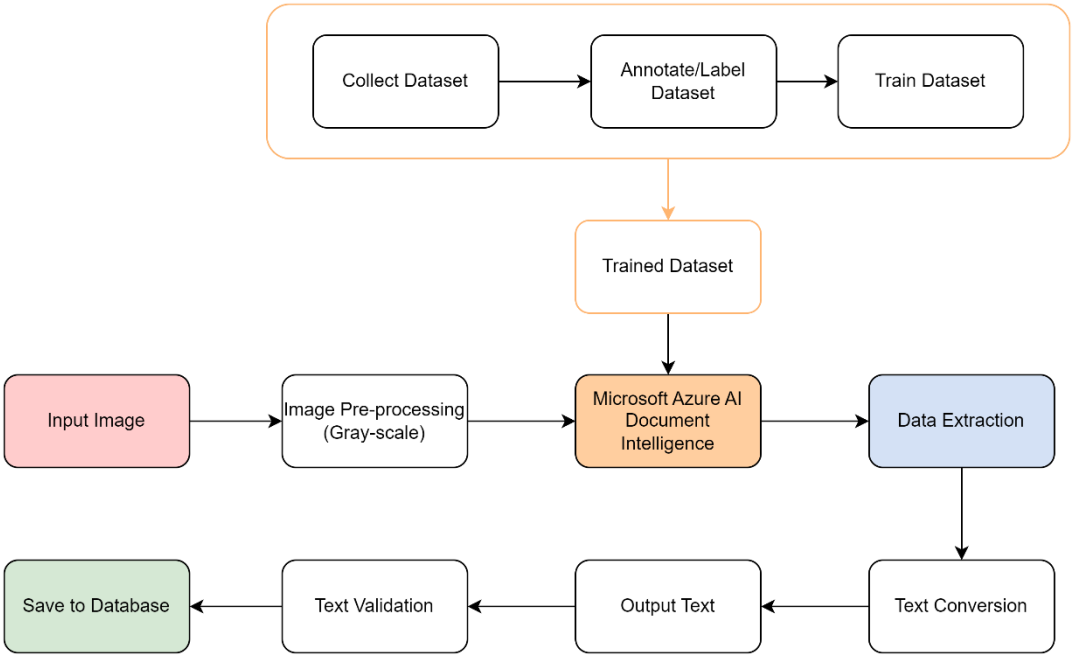


Figure 3.1: Automatic Data Extraction from Blood Test Report Pipeline

Hardware and Software

The correct set of software is needed for the mobile application development of automatic data extraction from blood test report. As mentioned in Chapter 2, Microsoft Azure AI Document Intelligence will be used for model training and data extraction. While for the mobile application development, PyCharm will be used as the integrated development environment (IDE) for Python programming. Azure SQL Database will be used as the database to store the extracted data. Besides, Figma and Canva will be used to create wireframe and the prototype of the application while Diagram.io will be used to create flowchart, DFD and ERD in User Design Phase. Lastly, Google Form will be used to create the questionnaire for user requirement collection purposes. Table 3.1 shows the list of software used.

Table 3.1: List of Software

Software	**Description**
Microsoft Azure AI Document Intelligence	A document processing service provided by Microsoft Corporation which supports data extraction.
PyCharm	An integrated development environment (IDE) for Python programming.
Android Studio	An Integrated Development Environment (IDE) for building Android applications.
Azure SQL Database	A database provided by Microsoft Corporation to store the extracted data.
Figma	An online design tool that used to create application prototype.
Diagram.io	An online diagram software for making flowcharts, DFD and ERD.
Canva	An online design platform used to create application wireframe.
Google Form	An online tool used to create questionnaire for user requirement collection.

The hardware requirement for the application will be based on the mobile device which act as the application running platform. The hardware tool needed for the application development is the Acer Nitro AN515-45 laptop. It has CPU of AMD Ryzen 5 5600H with Radeon Graphics which supports the application development. The 16.00 GB RAM in the machine will enhance quicker output during development. Table 3.2 shows the list of hardware used.

Table 3.2: List of Hardware

Hardware	Specification	Description
Laptop	Acer Nitro AN515-45 AMD Ryzen 5 5600H with Radeon Graphics	Supports the application development entirely.
	16.00 GB RAM	Enhance quicker output during development.
Mobile Phone	Huawei Mate 30 with EMUI 12	Used for testing the application in Android 10 (API 29) environment.

3.3 Rapid Application Development (RAD)

The methodology used for designing and developing this proposed blood test data extraction mobile application is Rapid Application Development (RAD). There are 4 stages in RAD which includes requirement planning, user design, development, and cutover (Martin, 1991). This methodology is used to have faster and flexible application development by dividing the development process into responsive modules and having short working cycles (Beynon-Davies et al., 1999; Putra & Lolly, 2021). The RAD model can be depicted in the Figure 3.2.

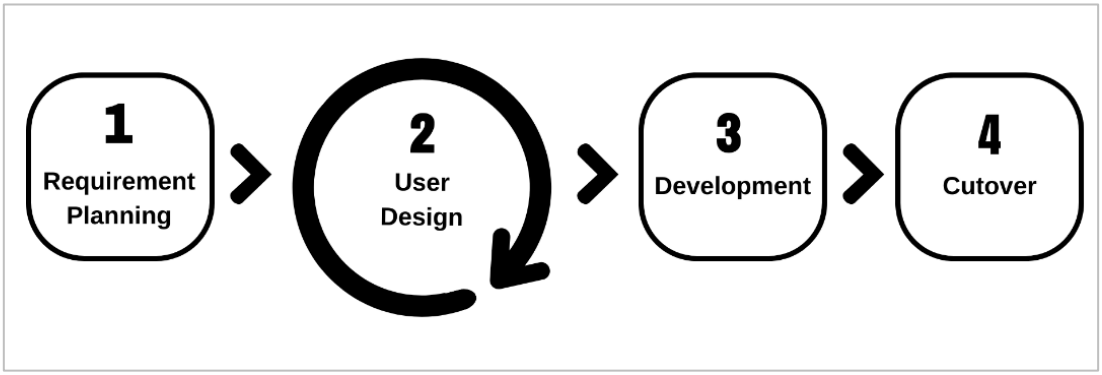


Figure 3.2: Rapid Application Development (RAD) Model

3.3.1 Requirement Planning

In Requirement Planning stage, activities that will be carried out by the application regarding the target user of the application and the user requirement for the automatic data extraction from blood test report mobile application will be identified. A questionnaire which is an efficient data collection instrumental technique was used to collect user requirements. Appendix B: User Requirement Questionnaire shows the total 20 questions categorized in 5 sections namely Section A: General Information, Section B: System Features, Usability and Compatibility, Section C: Data Accuracy and Validation, Section D: Reporting and Data Export, and Section E: Additional Requirements. This questionnaire helps to better understand the current needs in extracting blood test data from printed reports, as well as gathering the requirements from medical personnel to be used in the implementation of the automatic data extraction from blood test report mobile application.

The responses collected will then be used to define the information flow among application functions. Besides, analysis of the insights collected through the questionnaire will be conducted. The information gathered such as the current practice of blood test report digitization and the list of blood test report health metrics will be recorded and analysed for the use of the next stage. **Appendix A: Blood Test Report Health Metrics** had recorded the blood test report health metrics that will be used for data labelling.

In the first section of the questionnaire, Figure 3.3 had shown the first question which is to get the consent of the participant for their insight towards automatic data extraction from blood test report mobile application. Figure 3.4 had indicated the occupation of the participant. Where Figure 3.5 was asked to have a brief acknowledgement on the data size. Figure 3.6 had gathered the issue faced in extracting patient's blood test report data and this

had reflected toward the problem statement of the project. The urge of need towards the proposed automatic data extraction application had presented in Figure 3.7.

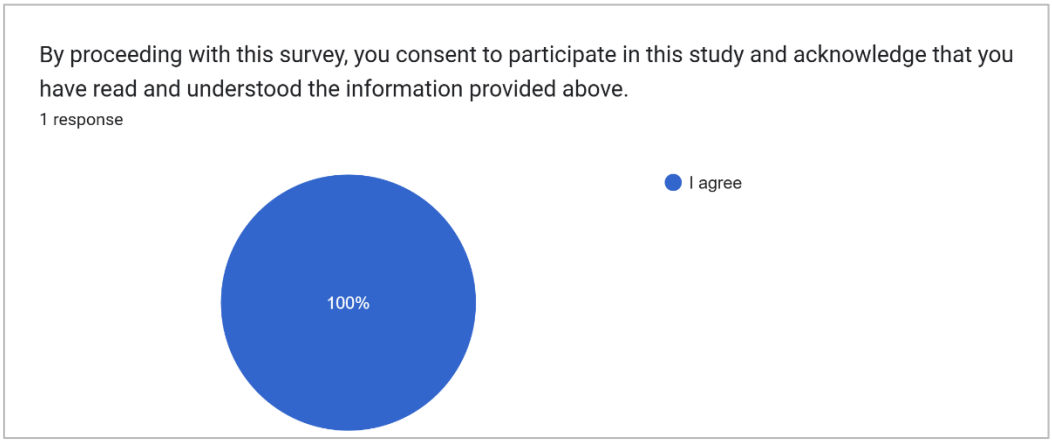


Figure 3.3: User Response for Question 1

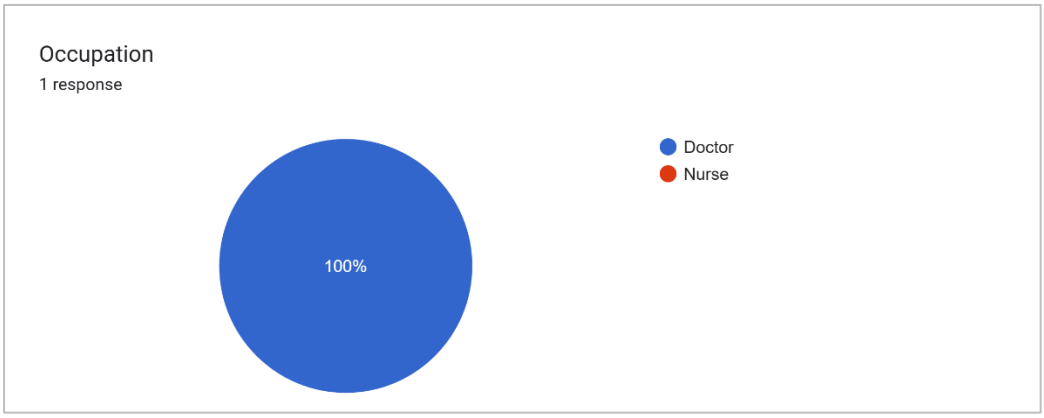


Figure 3.4: User Response for Question 2

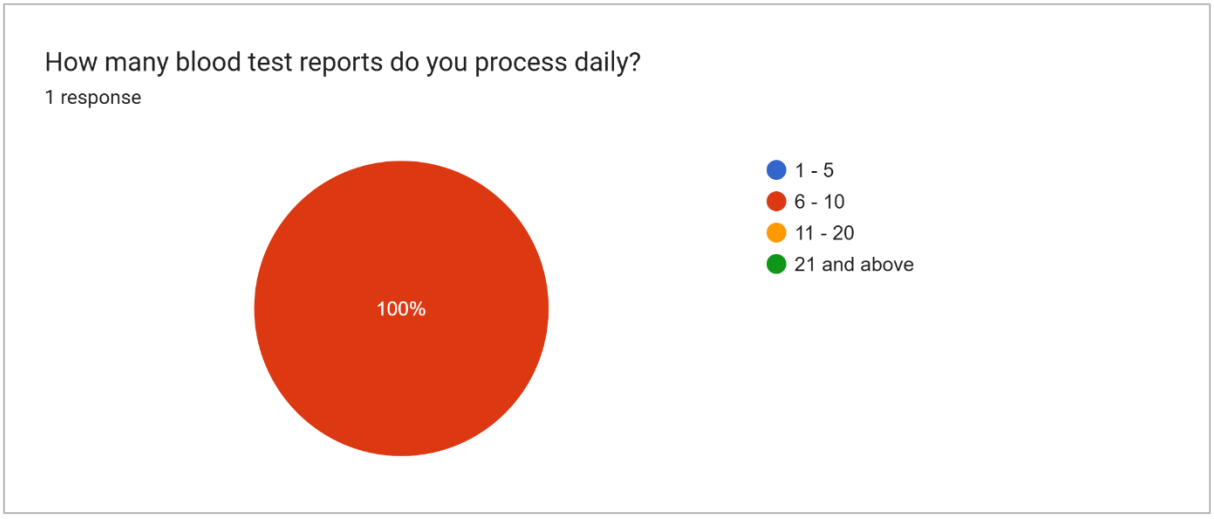


Figure 3.5: User Response for Question 3

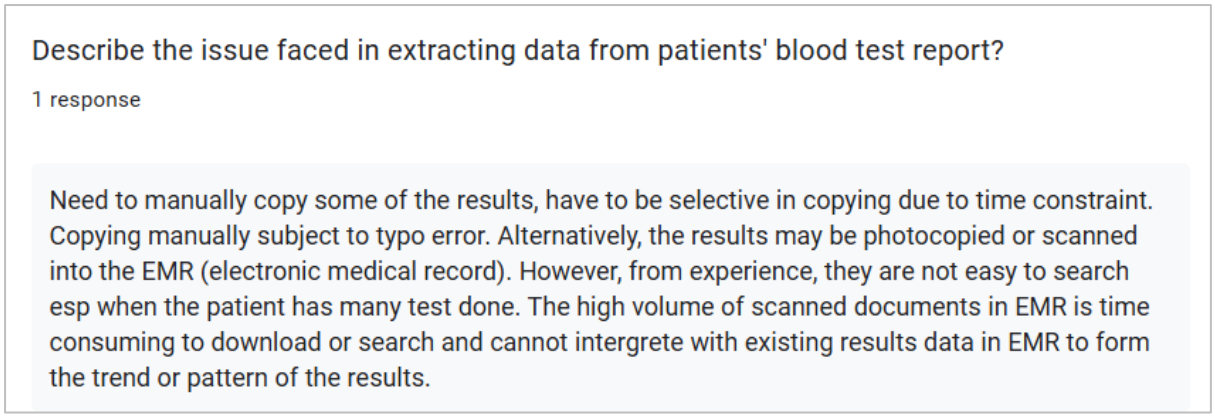


Figure 3.6: User Response for Question 4

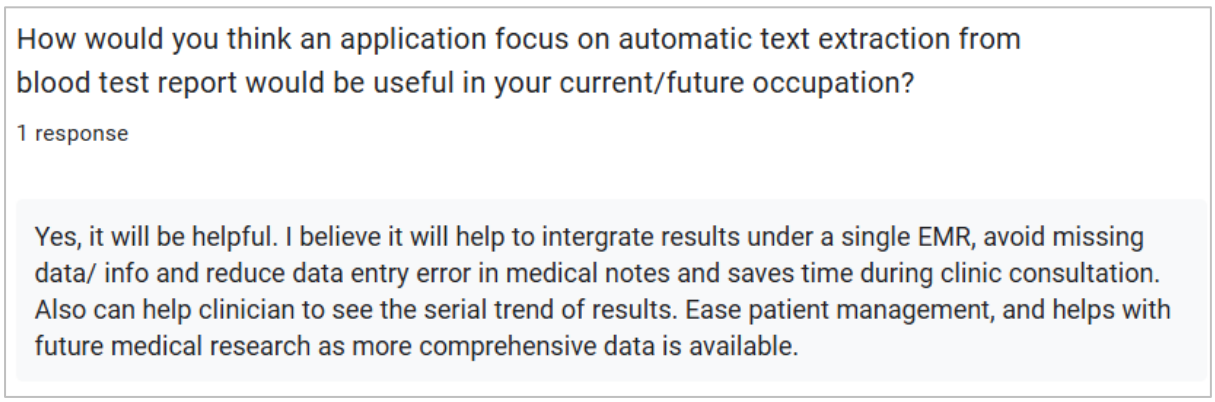


Figure 3.7: User Response for Question 5

In the second section of the questionnaire, Figure 3.8 had reflected the current practice in patient's blood test report managing and recording. It had shown that clinicians might still rely on time consuming manual transcription to record blood test report. User requirements such as highlighting abnormalities and non-reference units and automated data extraction from the blood test report had been shown in Figure 3.9. Where Figure 3.10 was asked to ensure that the proposed project is matching with the users' device. Figure 3.11 had gathered the actual patient information which will need to be extracted from the blood test report.

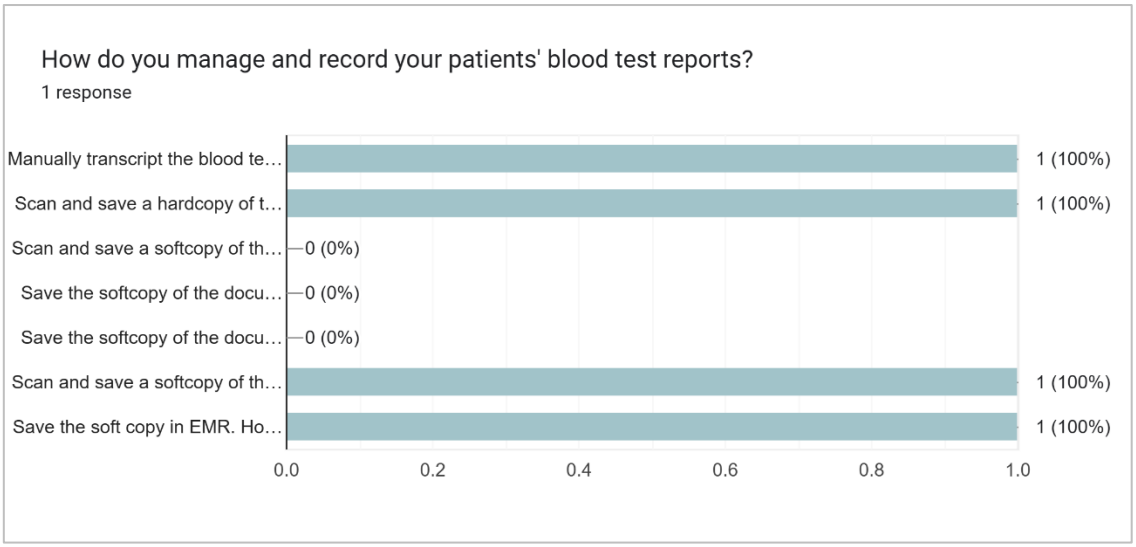


Figure 3.8: User Response for Question 6

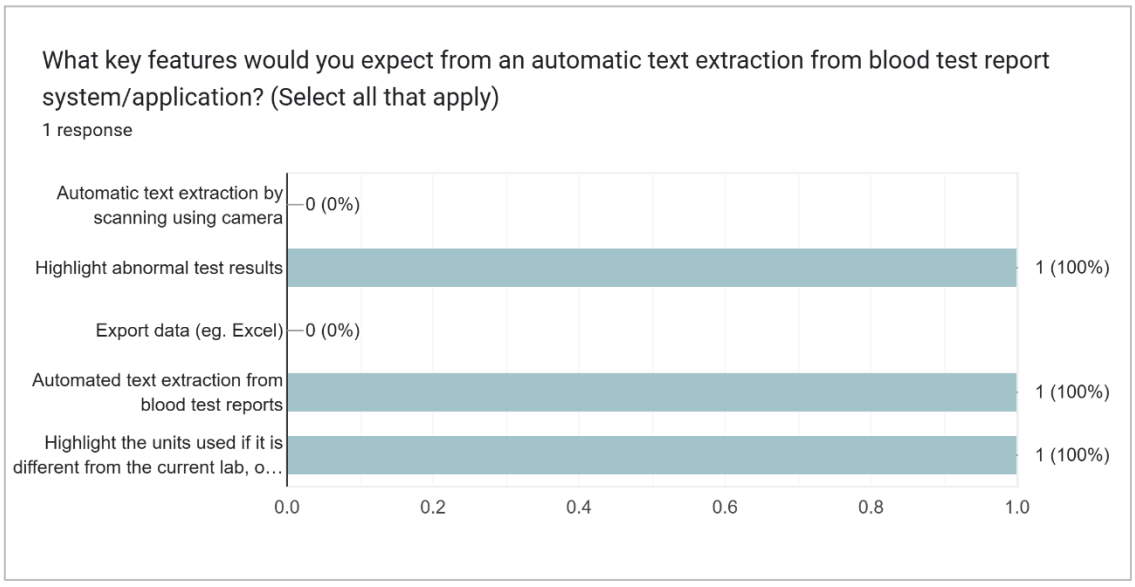


Figure 3.9: User Response for Question 7

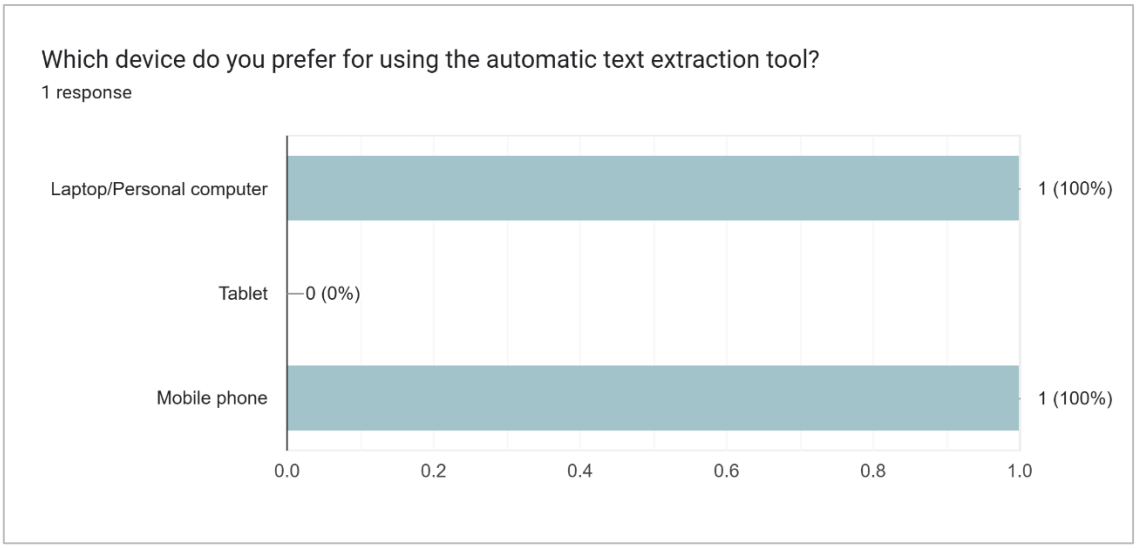


Figure 3.10: User Response for Question 8

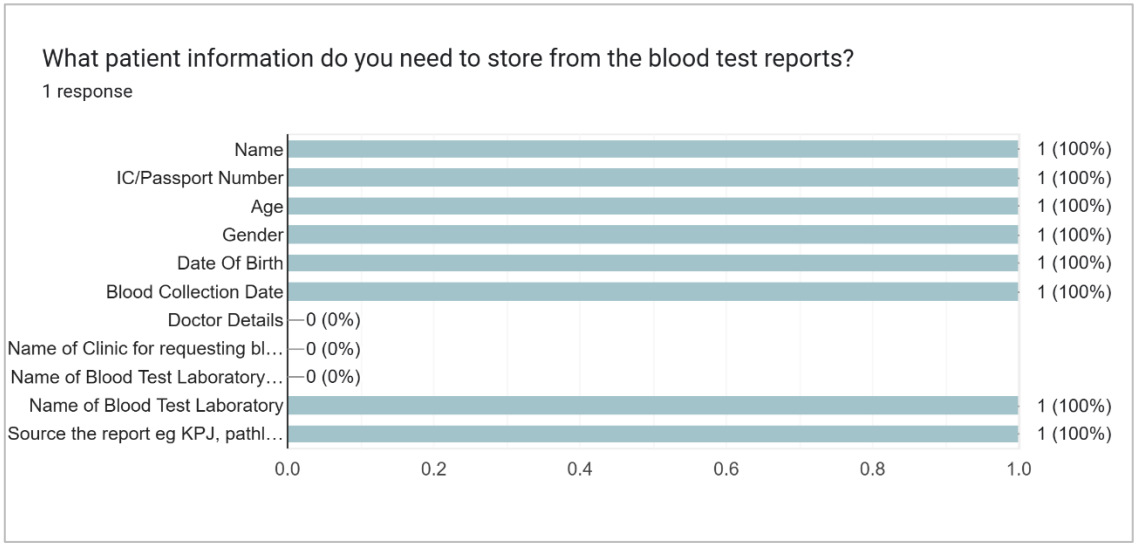


Figure 3.11: User Response for Question 9

Move on to the third section of the questionnaire, Figure 3.12 and Figure 3.13 had shown that user prefer to have manual validation and correction towards the blood test report data extraction to ensure the correct data was saved to the database. Where Figure 3.14 had represented the important of user authentication in data privacy and security.

Hence, user login and account sign up must be implement in the proposed application. Figure 3.15 had reflected the current issue faced by the participant regarding the heavily burdened EMR system due to large number of stored reports. This had indicated that the data should be stored in online database rather than local device storage.

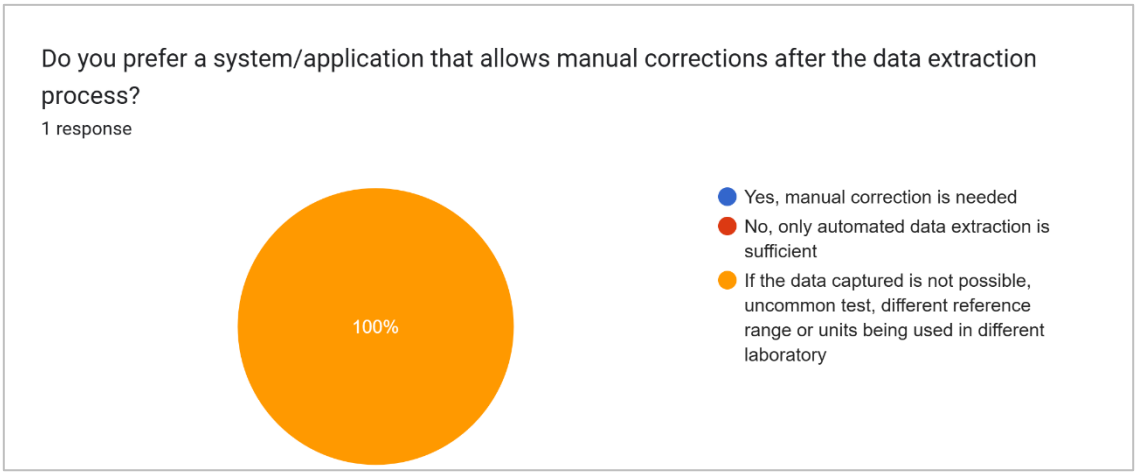


Figure 3.12: User Response for Question 10

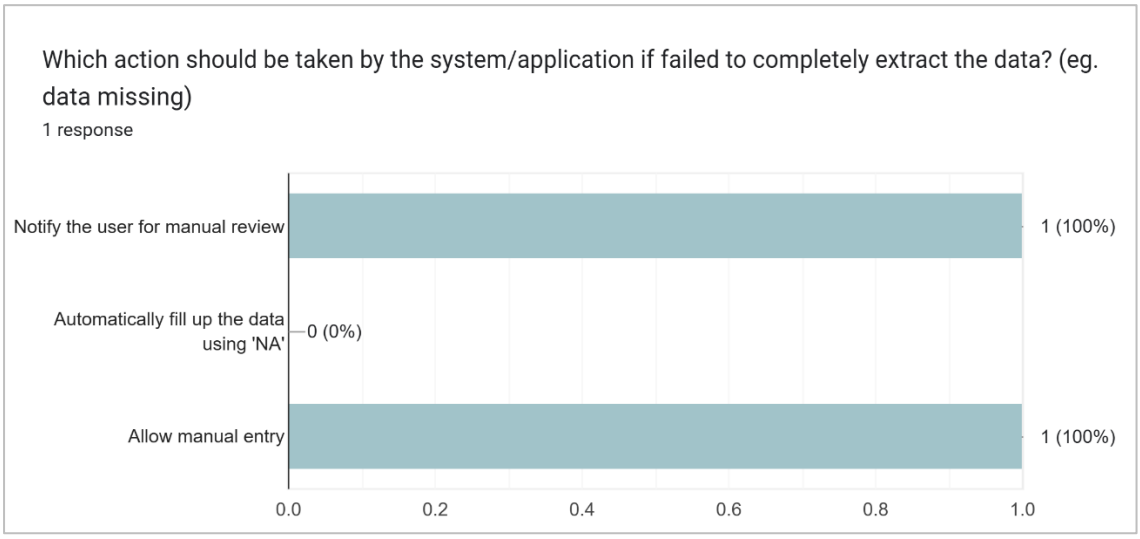


Figure 3.13: User Response for Question 11

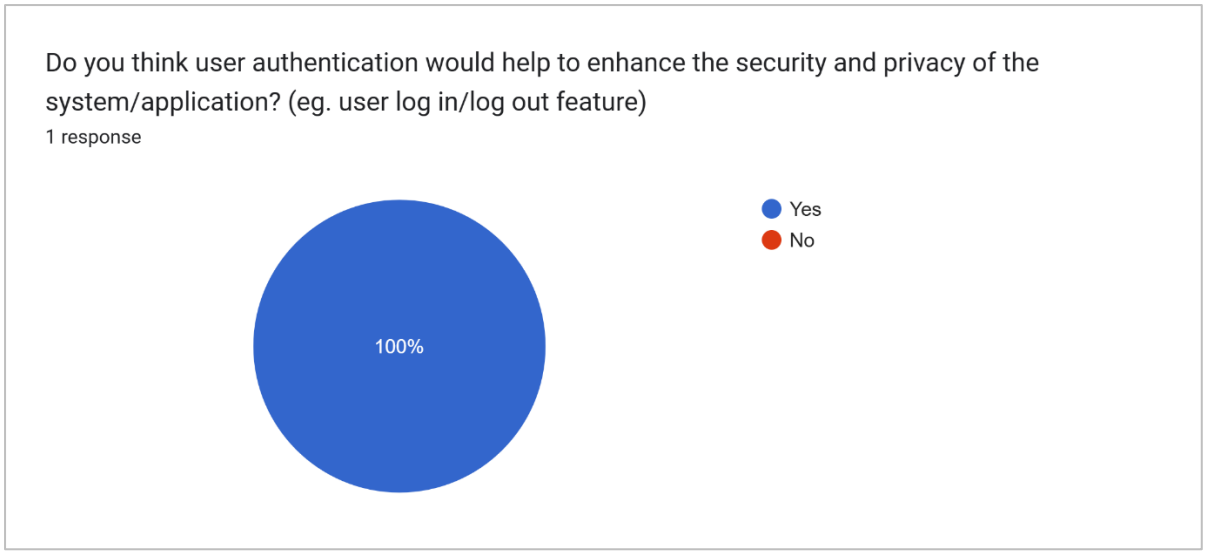


Figure 3.14: User Response for Question 12

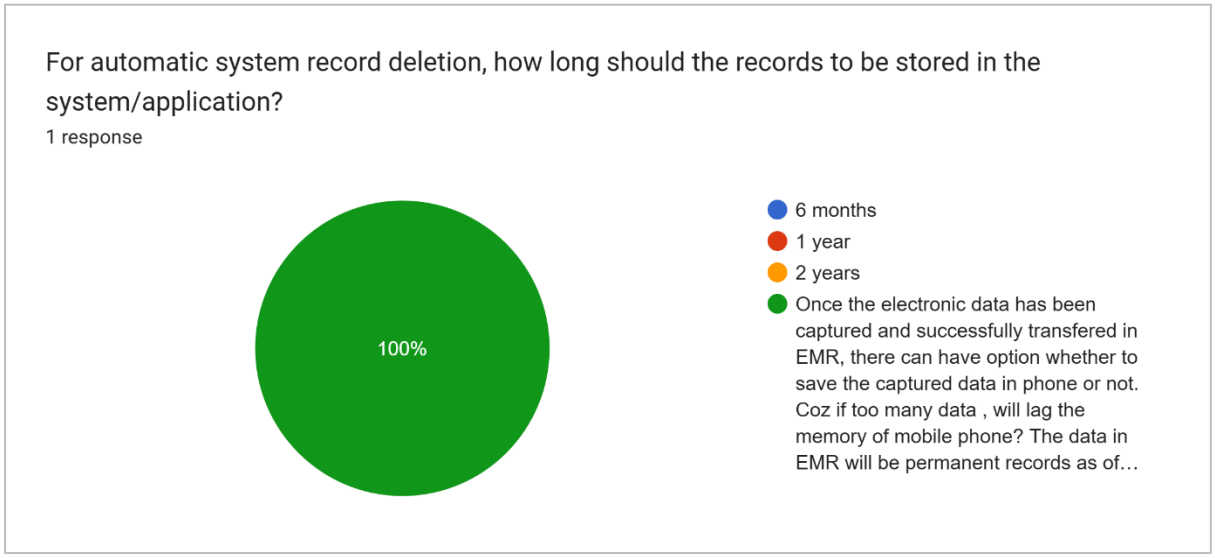


Figure 3.15: User Response for Question 13

In the fourth section of the questionnaire, Figure 3.16 had shown the preferred visualization format to view the extracted blood test reports which is Table and Line Chart. Figure 3.17 had shown that a detailed report with all blood test metrics was needed. The participant also provided a direction for future research which is expanding the scope to include all test metrics from other health test such as renal test and liver

function test into the data extraction project. Where Figure 3.18 had reflected the preferred indicator used to search for a specific blood test report from the database which is the patient registration reference number.

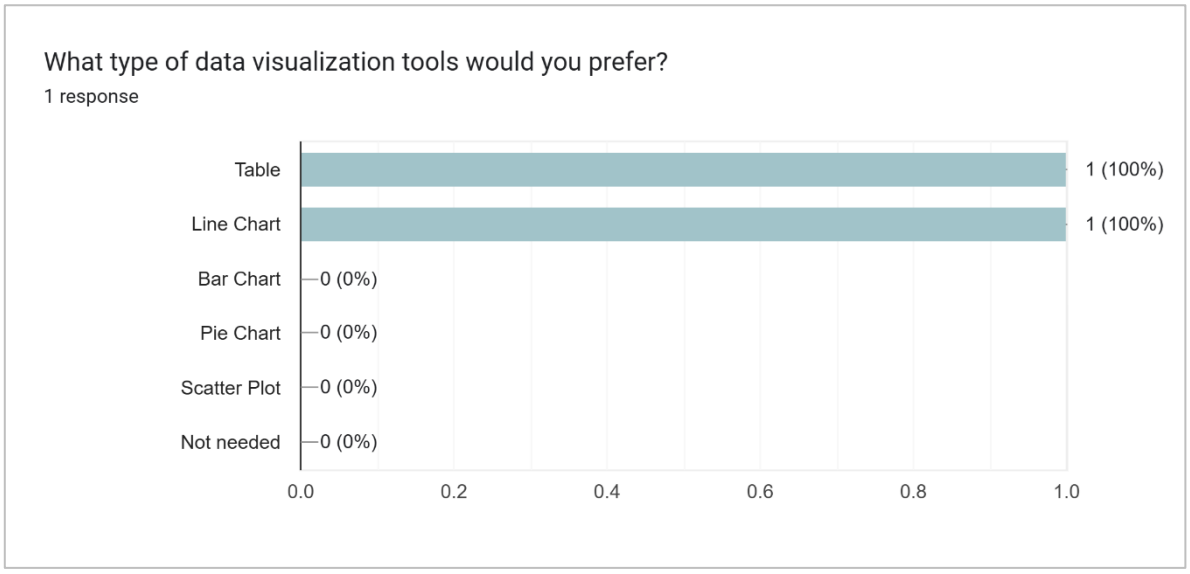


Figure 3.16: User Response for Question 14

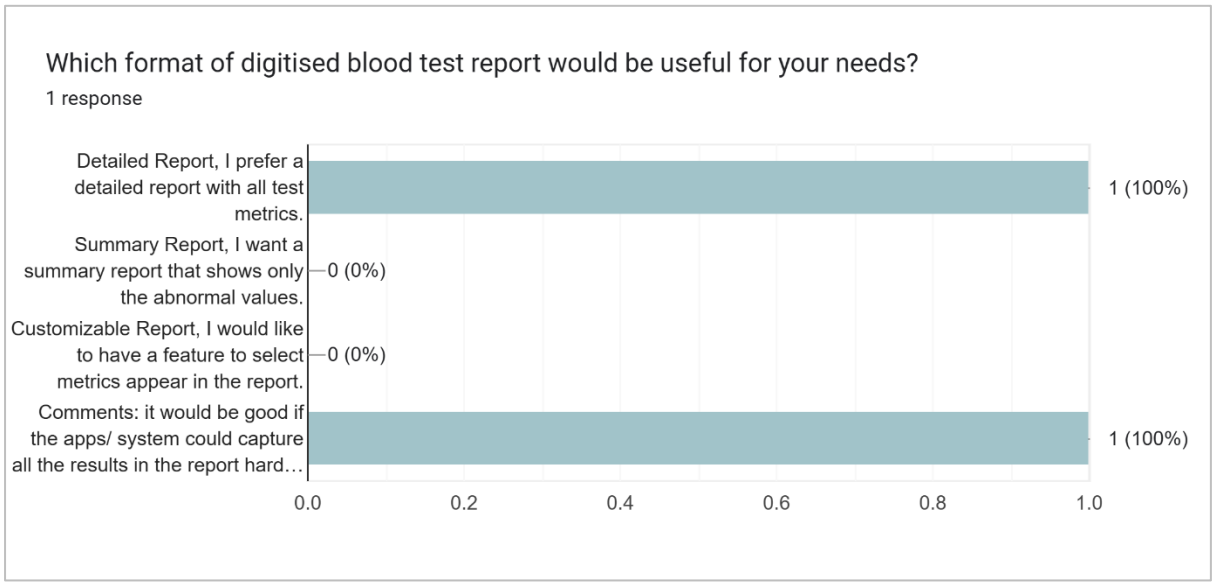


Figure 3.17: User Response for Question 15

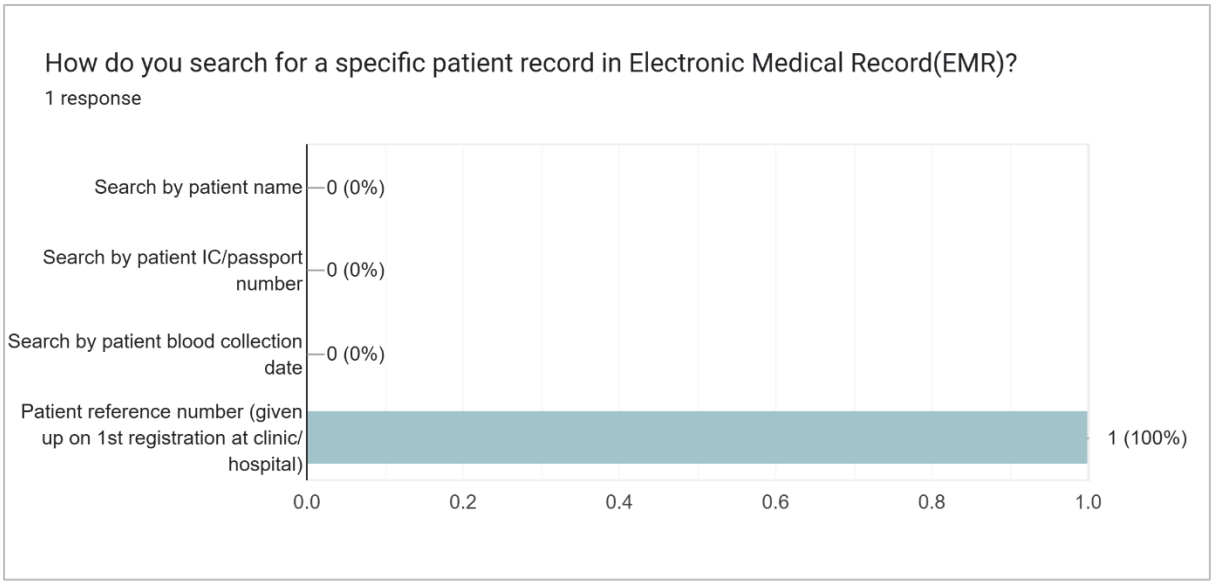


Figure 3.18: User Response for Question 16

In the last section of the questionnaire, Figure 3.19 had shown the preferred key data display in the dashboard. The participant prefers the use of text instead of icon in the application showing in Figure 3.20. Besides, the participant chosen to have user interface design with multiple colours as shown in Figure 3.21. Where Figure 3.22 had described the extra features suggested by the participants which is to highlight the units of the test metrics that differ from the lab reference value.

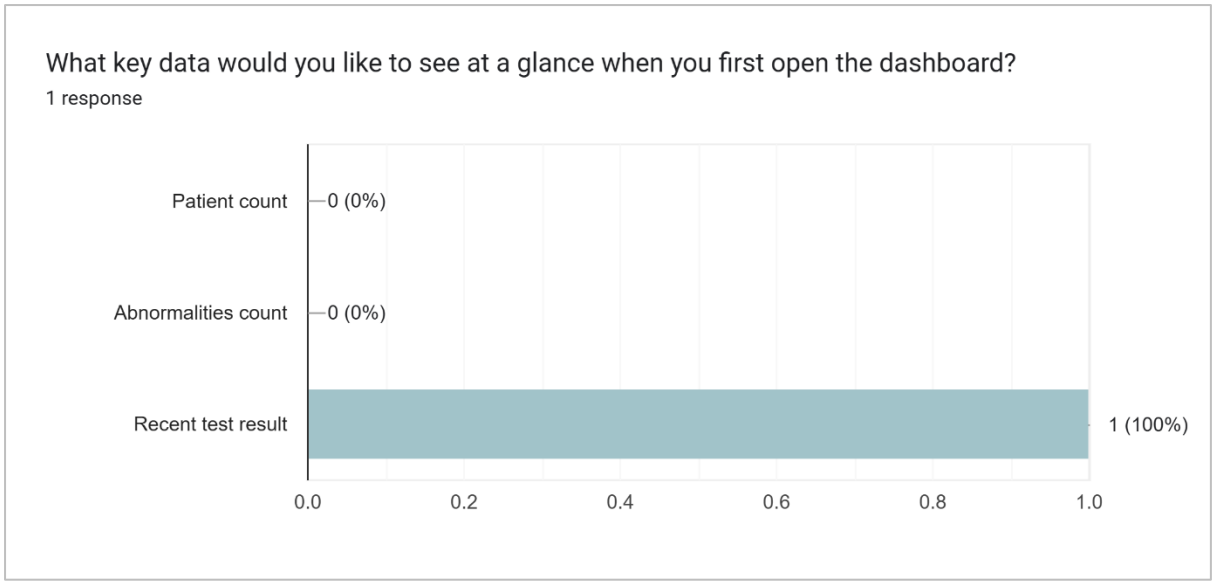


Figure 3.19: User Response for Question 17

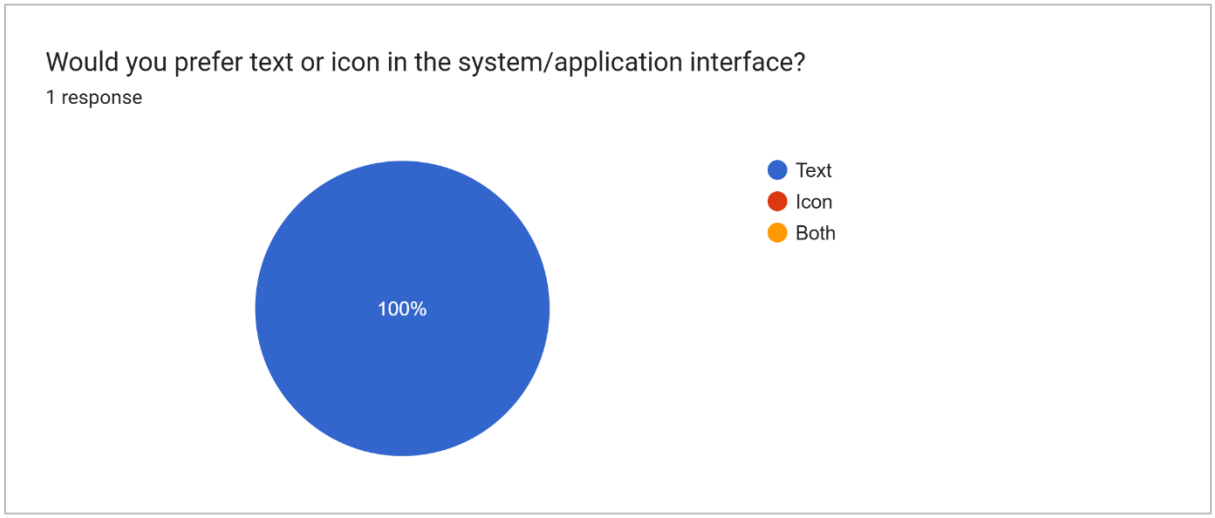


Figure 3.20: User Response for Question 18

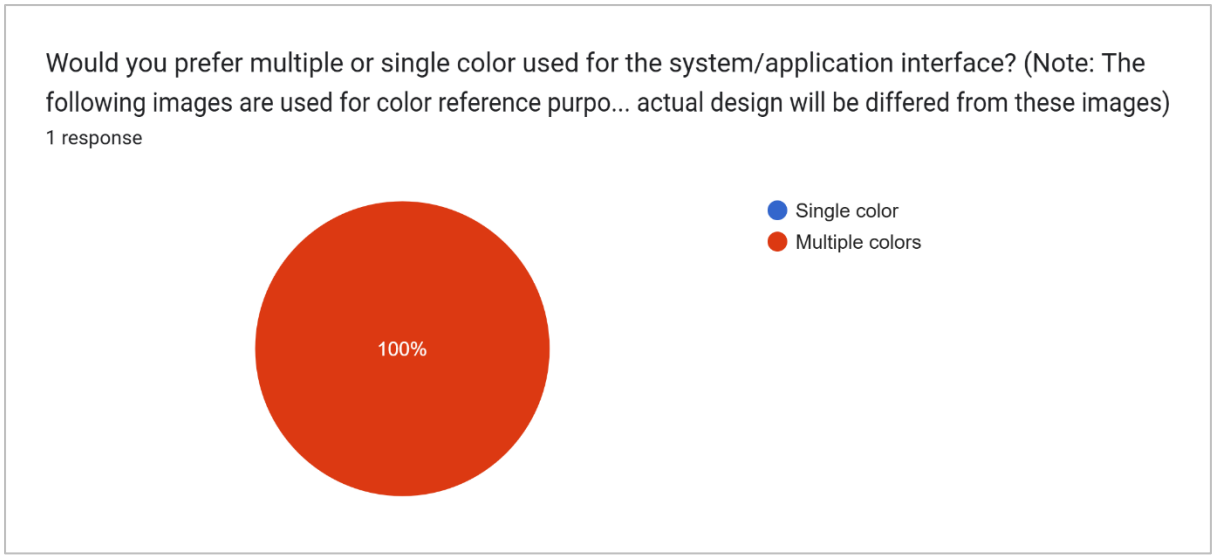


Figure 3.21: User Response for Question 19

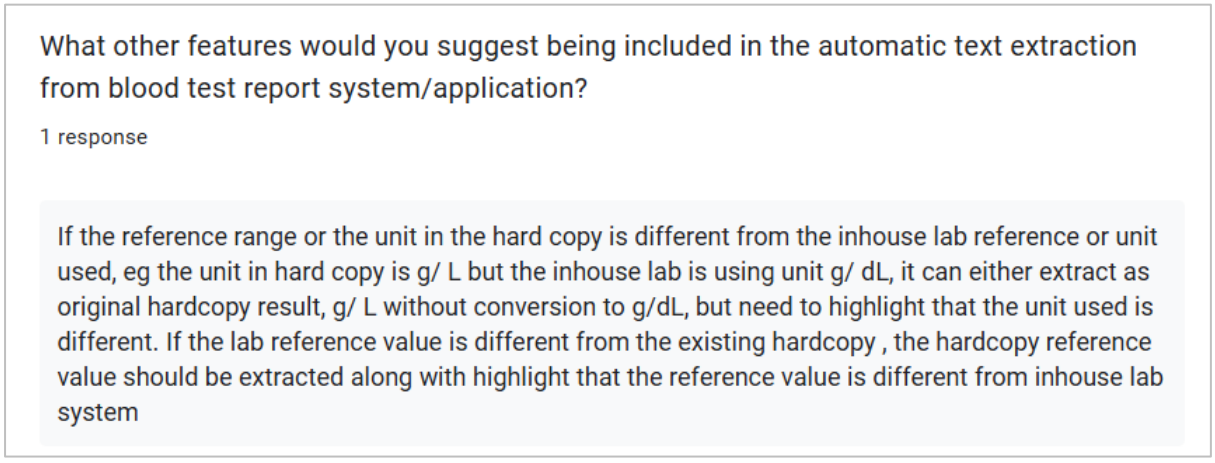


Figure 3.22: User Response for Question 20

3.3.2 User Design

The data collected from User Design stage is refined into a set of entities that support the application. The information had identified the characters of each entity and the relation between these entities. This stage is crucial to ensure determine the application functions

which complies with the project objectives before progressing deeper into the project. The proposed data extraction mobile application will offer the users with the features as stated below:

- Sign up account
- Login and log out account
- Extract data from input image
- Verify extracted data
- Save validated data to database
- View extracted blood test report in table format
- Search extracted blood test report from database by patient registration reference number
- Manage saved blood test report
- Generate line chart visualization
- Export saved blood test report

Besides, the proposed data extraction mobile application will offer the admin users with the features as stated below:

- Verify user account
- Login and log out account
- View evaluation such as CER and confidence score for the extracted fields of the report

- Approve user usage limit request
- Delete user account
- View user profile
- Search extracted blood test report from database by report ID

Entity Relationship Diagram (ERD)

The information which had been defined in the data modelling phase are then transformed into Entity Relationship Diagram (ERD) to describe the database design in a logical and graphical view about the relationship of the application. Figure 3.22 shows the ERD of the Automatic Data Extraction from Blood Test Report Mobile Application.

The user account details such as user's name, email address, account password, IC/passport number, account created date and company/organization name will be collected during account sign up and recorded in the 'Users' table. Two table parameters namely 'isVerified' and 'isAdmin' will also be stored under this table for user account verification. Besides, parameters such as 'usageLimit' and 'requestLimit' will also be stored under this table for document uploading control.

The 'Report' table will used to store the patient age, report collection date, laboratory of blood test, and saved reference report file URL. This table contained a foreign key of user ID referenced from the 'Users' table and another foreign key of patient ID referenced from the 'Patient table to identify the creator and the patient details of the report.

The patients' details including name, IC/Passport number, age, gender, date of birth and blood bank information such as ABO grouping and Rh (Anti-D) will be store in the 'Patient' table. Each patient will have an ID number given by the healthcare organization during

patient registration which is the patient registration reference number. This ID number will be used as the primary key for the ‘Patient’ table. Besides, this table contained a foreign key of user ID referenced from the ‘Users’ table to identify the creator of the patient profile.

The extracted blood test metrics stated in Appendix A will be store into the ‘Parameter’ table which contain table parameters such as the metrics’ value, unit, normal range, and health status. These data will then be referenced to the ‘ReportParameterResult’ table which used to view the extracted data saved in ‘Report’ table and ‘Parameter’ table using the report ID and Parameter ID as foreign key.

Table ‘PerformanceEvaluation’ is used to store the CER and confidence score of each blood test metric. Admin could retrieve and view the data from this table for application performance evaluation.

In this mobile application, each user can have multiple patients and blood test reports while each report will consist of only one patient with multiple parameters.

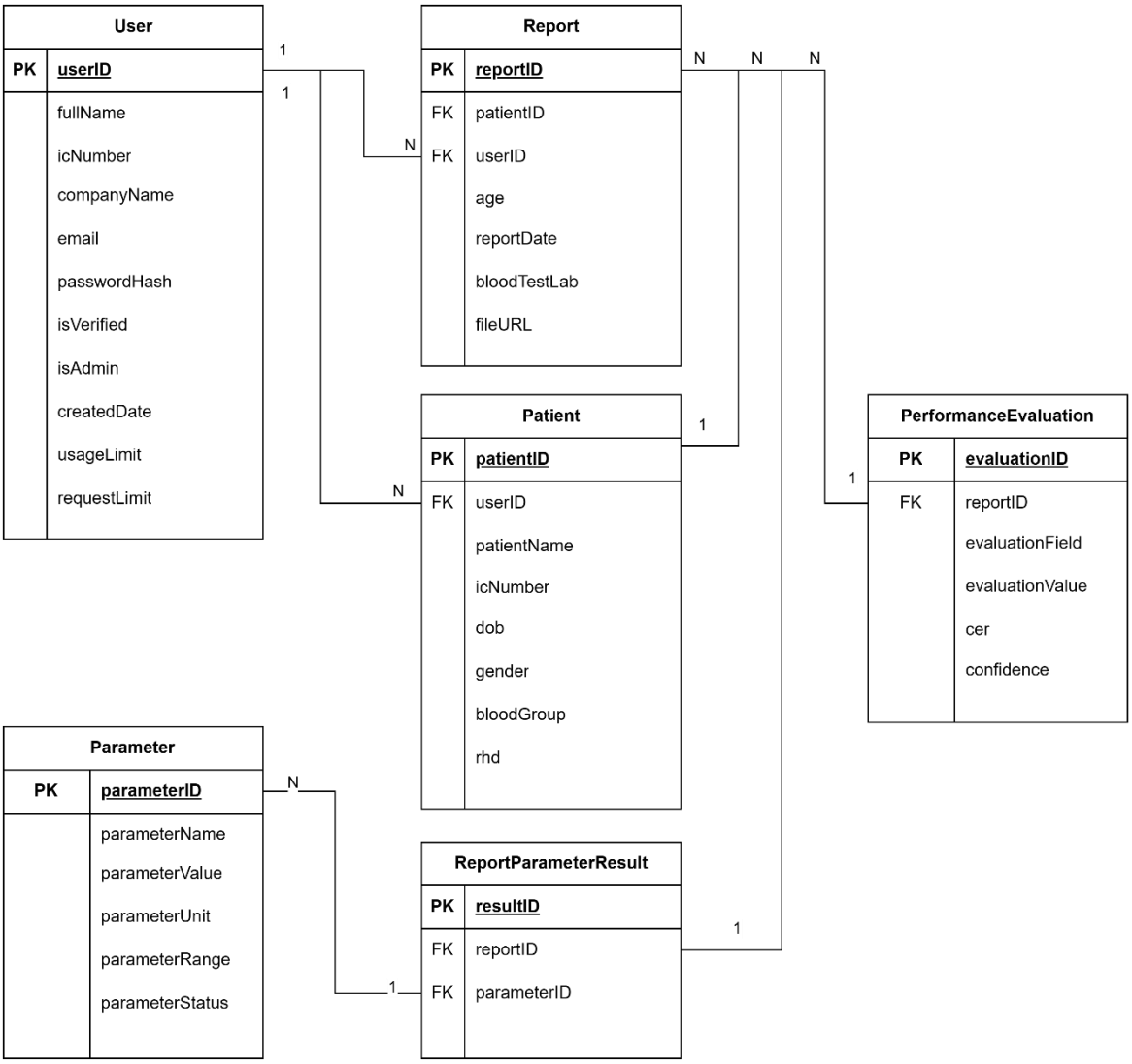


Figure 3.23: ERD of Proposed Solution

Data Dictionary

Data dictionary is a collection of the data object description which includes the table name, field name, data type, and data constraint. Table 3.3, Table 3.4, Table 3.5, Table 3.6, Table 3.7, and Table 3.8 had shown the data dictionary for the automatic data extraction from blood test report application.

Table 3.3: User Table

Field Name	Data Type	Constraint
userID	Integer	Primary Key
fullName	Nvarchar (100)	No Constraint
icNumber	Nvarchar (20)	No Constraint
companyName	Nvarchar (100)	No Constraint
email	Nvarchar (100)	No Constraint
passwordHash	Nvarchar (255)	No Constraint
isVerified	Bit	No Constraint
isAdmin	Bit	No Constraint
createdDate	Date	No Constraint
usageLimit	Integer	No Constraint
requestLimit	Integer	No Constraint

Table 3.4: Report Table

Field Name	Data Type	Constraint
reportID	Integer	Primary Key
patientID	Integer	Foreign Key
userID	Integer	Foreign Key
age	Integer	Foreign Key
reportDate	Date	No Constraint
bloodTestLab	Nvarchar (100)	No Constraint
fileURL	Nvarchar (2083)	No Constraint

Table 3.5: Patient Table

Field Name	Data Type	Constraint
patientID	Integer	Primary Key
userID	Integer	Foreign Key
patientName	Nvarchar (100)	No Constraint
icNumber	Nvarchar (20)	No Constraint
dob	Date	No Constraint
gender	Varchar (10)	No Constraint
bloodGroup	Varchar (3)	No Constraint
rhd	Nvarchar (8)	No Constraint

Table 3.6: Parameter Table

Field Name	Data Type	Constraint
parameterID	Integer	Primary Key
parameterName	Nvarchar (100)	No Constraint
parameterValue	Float	No Constraint
parameterUnit	Nvarchar (20)	No Constraint
parameterRange	Nvarchar (50)	No Constraint
parameterStatus	Varchar (50)	No Constraint

Table 3.7: ReportParameterResult Table

Field Name	Data Type	Constraint
resultID _ID	Integer	Primary Key
reportID	Integer	Foreign Key

parameterID	Integer	Foreign Key

Table 3.8: PerformanceEvaluation Table

Field Name	**Data Type**	**Constraint**
evaluationID	Integer	Primary Key
reportID	Integer	Foreign Key
evaluationField	Nvarchar (255)	No Constraint
evaluationValue	Nvarchar (50)	No Constraint
cer	Float	No Constraint
confidence	Float	No Constraint

Context Diagram

Context Diagram is the highest level of Data Flow Diagram which represent the entire process flow of the application. This diagram helps to visualize the relationship between the proposed application and the entity involved. Figure 3.23 had shown the context diagram of the proposed application.

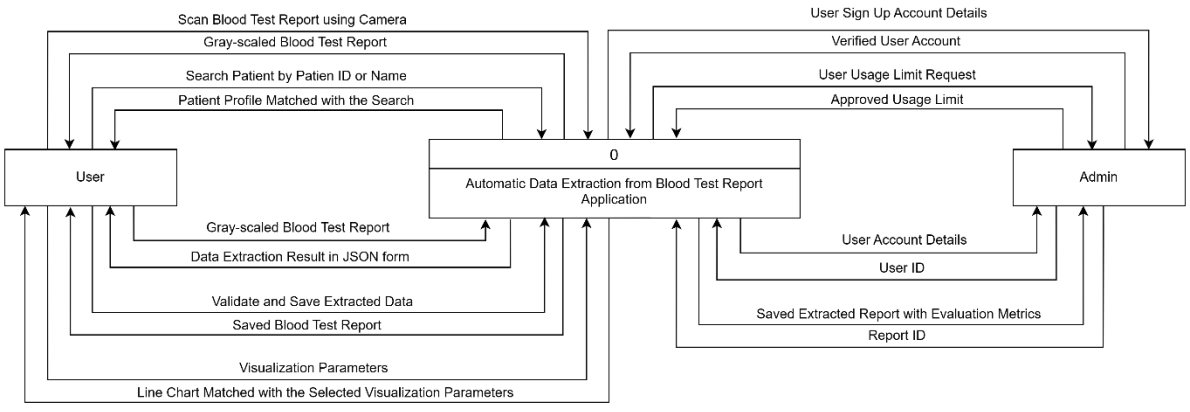


Figure 3.24: Context Diagram of Proposed Solution

Data Flow Diagram (DFD)

Figure 3.24 had shown the activity description of major application processes of the proposed application. There are 5 major processes identified in this application which is user account management, data extraction and storage, data retrieval, data visualization and reporting, and data export and management.

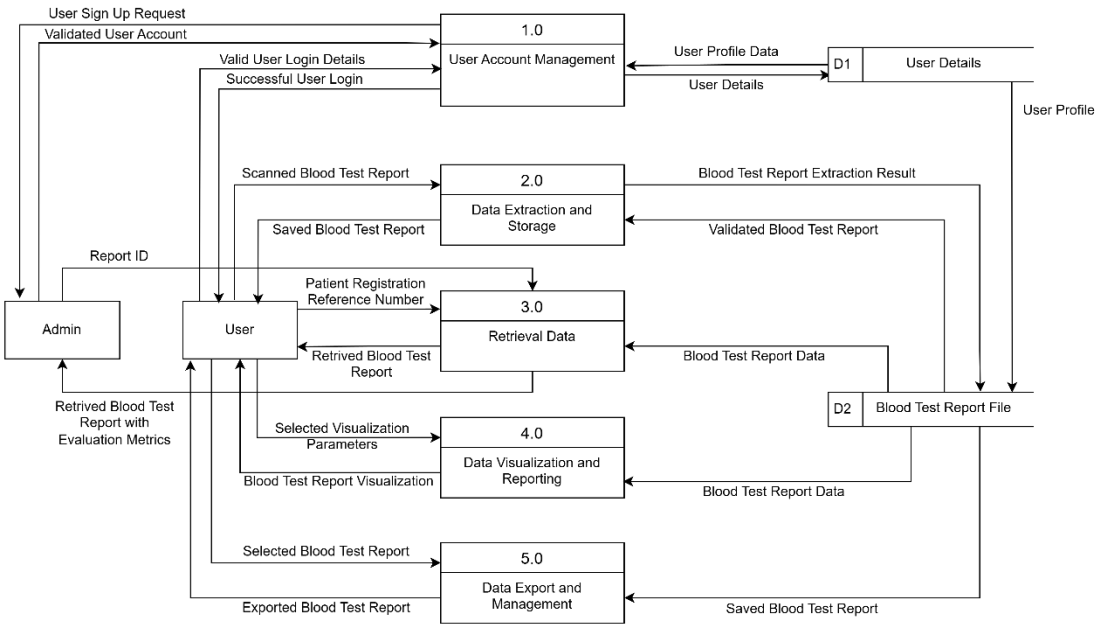


Figure 3.25: Data Flow Diagram Level 1 of Proposed Solution

Process 1.0 – User Account Management

The new user will need to sign up for an account and the sign-up request will be validated by the admin. After account validation, user may proceed to user login. This user authentication process is needed for user privacy and data security.

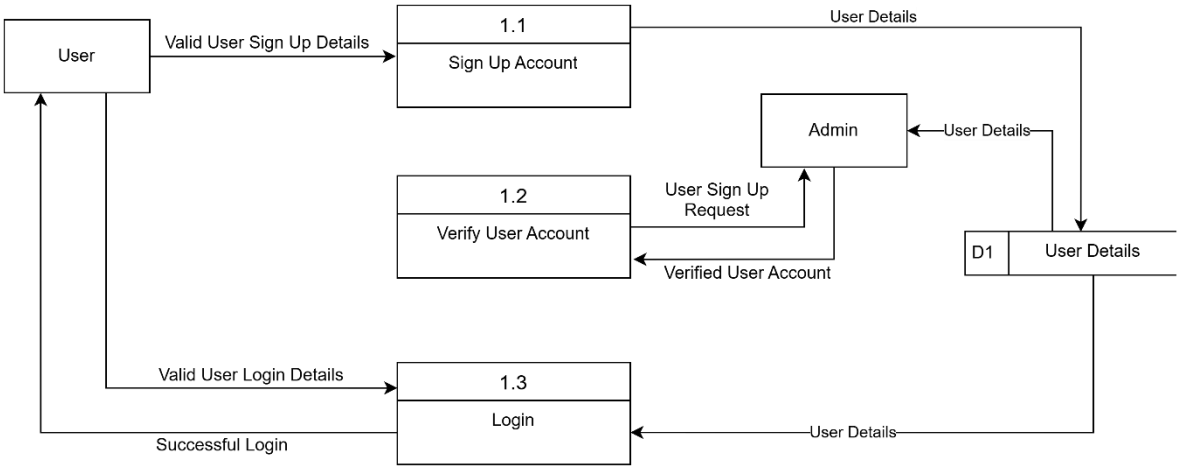


Figure 3.26: Data Flow Diagram Level 2 for Process 1.0

Process 2.0 – Data Extraction and Storage

User will need to approve for the camera usage permission to capture the image of the blood test report for further data extraction. The extracted blood test data will be manually verified by the user before storing into the database. The user will be navigate to a page that displays the saved blood test report data in table format.

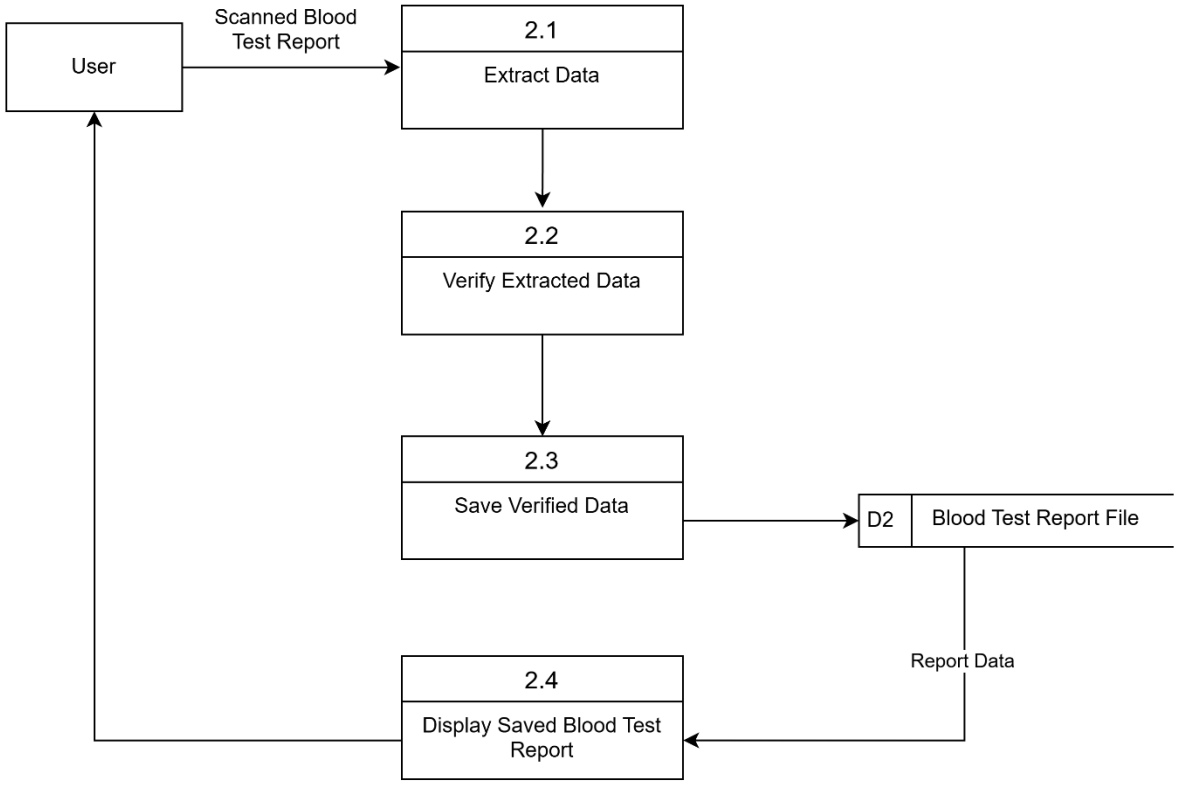


Figure 3.27: Data Flow Diagram Level 2 for Process 2.0

Process 3.0 – Data Retrieval

User will search for patient using patient registration reference number (patient ID). The search result including the matching patient details and report registered under the same patient ID will then be displayed to the user. The user can view the complete report data by selecting a report option. For admin, they can search for report using report ID or report created date. The search result will then be displayed to the admin and they can view the report with evaluation metrics including CER and confidence score by selecting an report option from the search result.

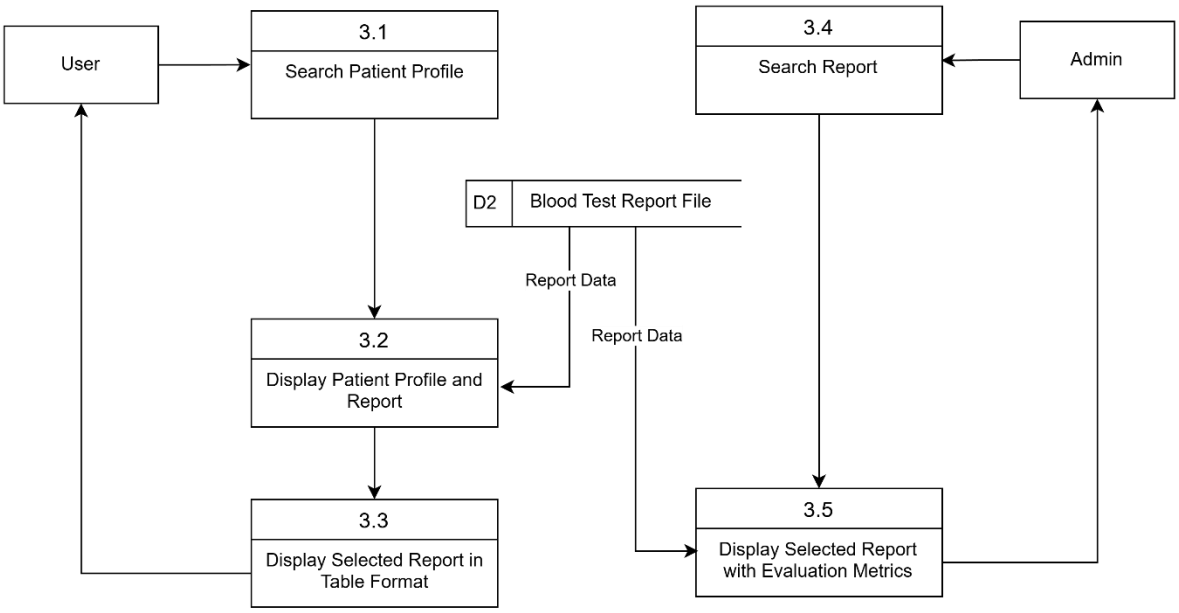


Figure 3.28: Data Flow Diagram Level 2 for Process 3.0

Process 4.0 – Data Visualization and Reporting

User will need to select the visualization parameter by selecting 1 registered patient, 1 blood test parameter and a target unit to generate the line chart. A line chart displaying the selected blood test parameter with normalized value based on the selected target unit of all saved report under the selected patient will be displayed. The user can interact with the generated line chart by tapping on the marker and the report details will be shown in tooltip.

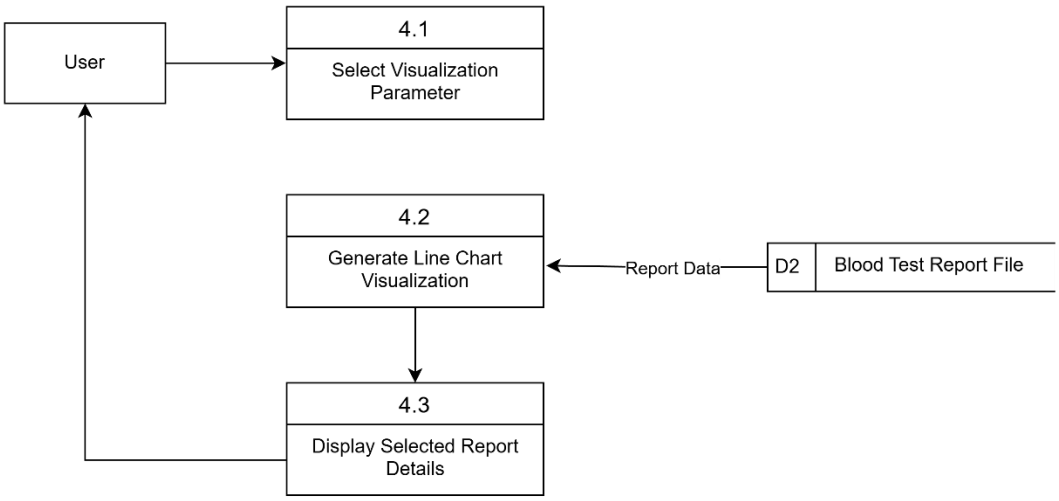


Figure 3.29: Data Flow Diagram Level 2 for Process 4.0

Process 5.0 – Data Export and Management

User will need to select a specific blood test report for edit or delete purposes. The edited report will then be saved into the database. Besides, user will have an option to export the saved blood test report in.

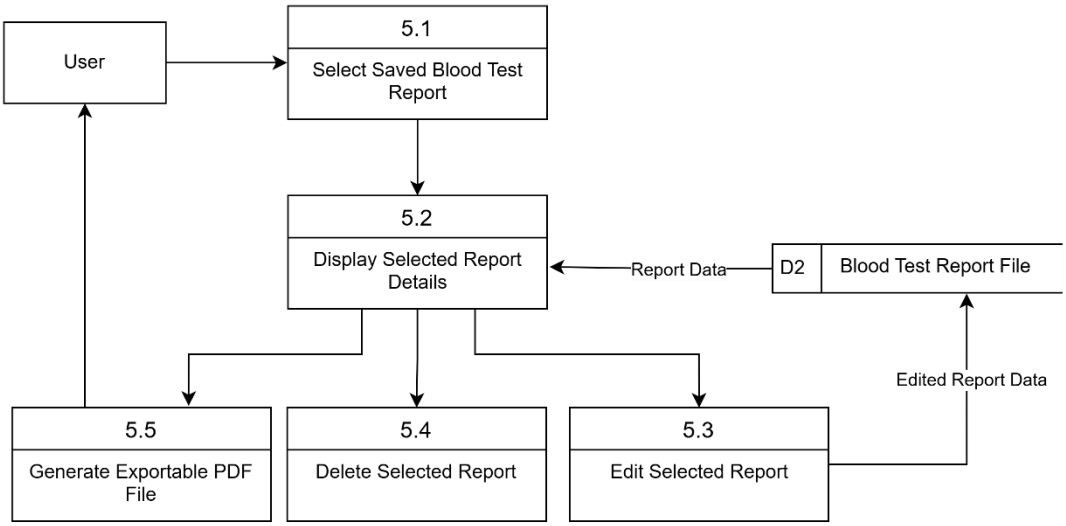


Figure 3.30: Data Flow Diagram Level 2 for Process 5.0

Wireframe

The wireframe of the mobile application which indicates the user interface will be created in this stage to show the basic structure and navigation of the application pages. Figure 3.31 shows the login page of the application where the user will need to key in their registered email address and their password. Then, the user will be led to the home page which is also the scanning page as shown as Figure 3.32. When the icon at the upper right corner was activated, the sidebar menu will appear as shown as Figure 3.33. Users can navigate to the Scanned Record page and Patient Detail page from the sidebar menu.

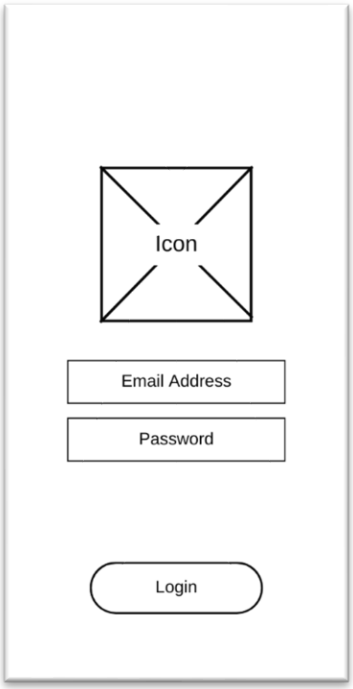


Figure 3.31: Login Page

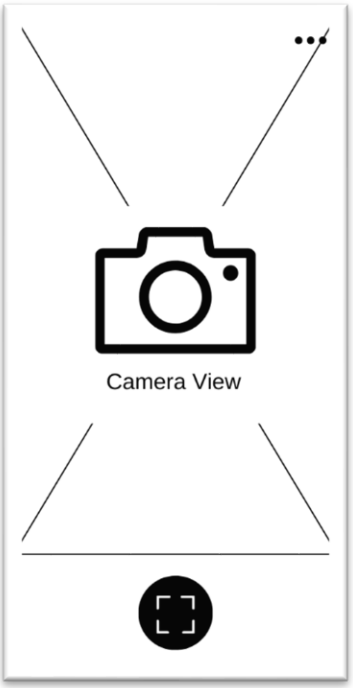


Figure 3.32: Home Page

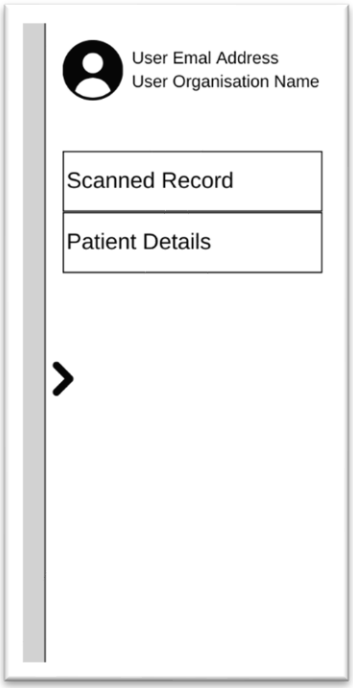


Figure 3.33: Sidebar Menu

Figure 3.34 shows the Scanned Result page of the application where the user can validate the extracted result then save it into the database. The saved records will be shown in the Scanned Report page and the Patient Details page respectively. The user can view,

edit, or delete the selected patient details as shown in Figure 3.35. For the saved blood test reports, user can visualize, edit, or delete the selected reports as shown in Figure 3.36.

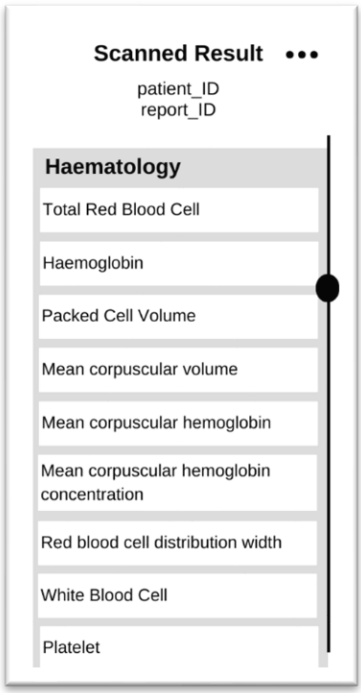


Figure 3.34: Scanned Result Page

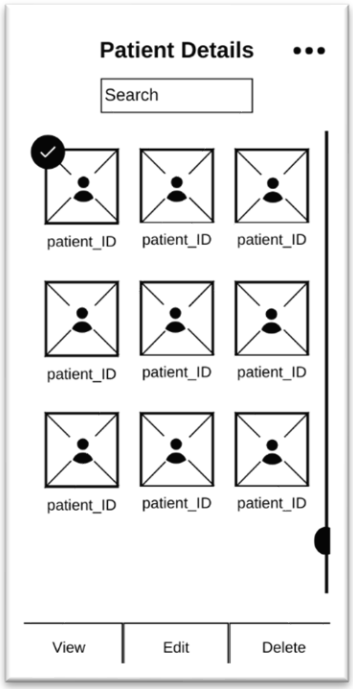


Figure 3.35: Patient Details Page

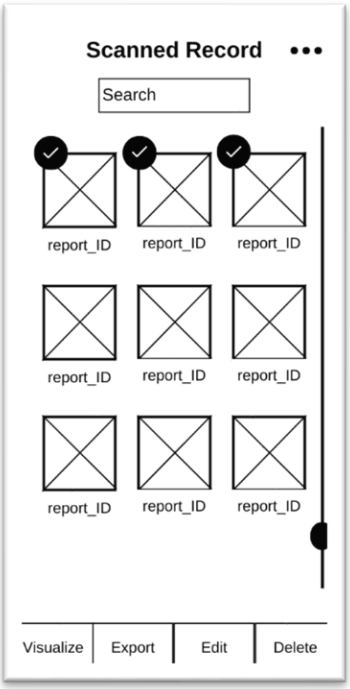


Figure 3.36: Scanned Report Page

For admin side, Figure 3.37 shows the Report Evaluation page of the application where the admin can view the extracted report with evaluation metrics consists of CER and confidence score value. Figure 3.38 shows the Verify User page for the admin to view current user account verification status. Unverified user account will be shown under the 'Unverified User' tab while 'All User Account' will shows all registered users. The admin can select an user and verify the user details showing in Figure 3.39. The admin can choose to verify, delete, or increase usage limit by tapping on respective buttons.

Figure 3.37: Report Evaluation Page

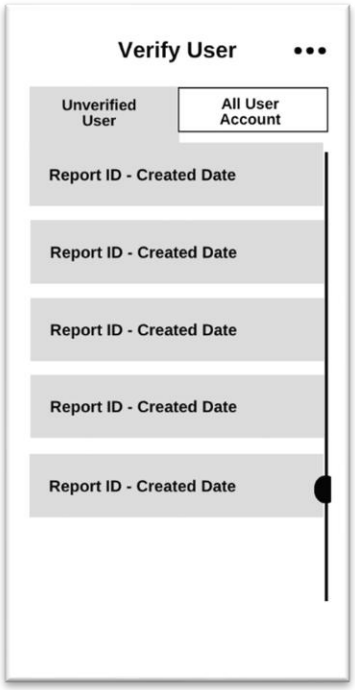


Figure 3.38: Verify User Page

Figure 3.39: Verify User Popup

Test Case

Test case is designed for the user testing which includes a detailed set of inputs to verify the application features and requirements. Table 3.9 had shown the test cases created for each of the features mentioned including sign up, login and log out account, extract data from input image, validate extracted data, save validated data to database, view extracted blood test report in table format, search extracted blood test report from database by patient registration reference number, manage saved blood test report, generate line chart visualization, export saved blood test report, verify user account, delete user, increase user usage limit, and view report evaluation.

Table 3.9: Test Case Design of Proposed Application

Test Case No.	Action	Input	Expected Output	Actual Output	Test Result (Pass/Failed)
1	Sign up account	Name: Chin Ah Mei Password: 123Testing!	Sign up successful notification		
2	Login account	Name: Chin Ah Mei Password: 123Testing!	Successful login account		
3	Extract data from input image	Scan blood test report with Haematology test	Data extraction result displayed for validation.		
4	Validate extracted data	Manual input for total red blood cell of 126	Total red blood cell value changed to 126		
5	Save validated data to database	Click the save button after validation	Successfully saved record notification displayed.		
6	View extracted blood test report in table format	Select a saved blood test record for viewing	Selected saved blood test record displayed in table format		
7	Search extracted blood test report from database by patient registration reference number	Search using patient registration reference number of P001	Blood test record with P001 displayed		
8	Edit saved blood test report	Edit the value of the white blood cell to 0 and save it	White blood cell value changed to 0		
9	Delete saved blood test report	Select a saved blood test record for deletion	Selected blood test record had been deleted		
10	Generate chart visualization	Select a blood test parameter and a month selection for chart visualization	Selected blood test parameter record displayed in line chart		
11	Export saved blood test report	Select a saved blood test record to export	Selected blood test record had been downloaded in local device		
12	Verify user account	Select a user account for verification	Selected user had been verified and disappeared from the 'Unverified User' tab session.		
13	Delete user account	Select a user account for deletion	Selected user record had been removed from database.		

14	Increase user usage limit	Select a user account to increase his usage limit	Selected user usage limit had been increased		
15	View report evaluation	Select a report and view it in table form with evaluation metrics	Selected report data had been displayed in table form with evaluation metrics for all extracted fields.		

3.3.3 Development

In this stage, the designed model will be executed in the form of coding after it has been accepted by the users and the stakeholders. Python will be used as the programming language and Microsoft Azure Document Intelligence API will be integrated in the construction of the mobile application. Besides, a structured database for the proposed application will be created in this stage using Azure SQL Database based on the specifications gathered from the previous stages. The completed application will be tested by the users to ensure the application functions correctly and meets the requirements according to the designed test cases. The application will undergo iterative amendment and refinement process until it reached the satisfaction level of the users and stakeholders.

3.3.4 Cutover

This stage is the final phase of the development of this project. It had indicated that the project is ready for deployment in real world. Hence, it is important to ensure that application features, usability and interface are finalized with the users and stakeholders. User training should be clearly delivered, and the maintenance and evaluation should be carried out in this stage to ensure the stability and maintainability of the application.

3.4 Summary

This chapter outlines the methodology used in developing the Automated Blood Test Data Extraction Mobile Application using Microsoft Azure AI Document Intelligence. The chapter begins by discussing the chosen Rapid Application Development (RAD) model, which facilitates a flexible application development process. The RAD model is divided into four key stages which is requirement planning, user design, development, and cutover.

Besides, the chapter had highlighted the architecture of the proposed application. The integration of Microsoft Azure AI Document Intelligence will be the focus of this application as it supports customizable models for extracting structured and unstructured data from diverse document types. The workflow involves training the model on labelled datasets of blood test reports, embedding the trained model into a mobile application with Flask API, and enabling users to extract, validate, and store data efficiently. Azure SQL Database is chosen to manage extracted data, while the development is supported by tools such as PyCharm, Android Studio and hardware with robust specifications.

The methodology details each RAD phase. In the requirement planning stage, user requirements were gathered through questionnaires. The user design phase refined these requirements and further developed the logical framework with an Entity Relationship Diagram (ERD) and data flow diagrams (DFDs) that mapped out the major processes, including user account management, data extraction and storage, retrieval, visualization, and export. Besides, the key pages such as login, home, scanned results, and patient details, were designed for seamless user interaction in this stage. During the development phase, coding will be performed in Python by integrating the Azure AI Document Intelligence API based on the created wireframe from the previous stage for the application's user interface. Finally, the cutover phase is outlined, including alpha testing to ensure functionality alignment with requirements and beta testing by healthcare professionals to create a finalized application.

This chapter will act as a foundation for the mobile application development of automatic data extraction from blood test report. More detailed account of the actual development process will be described in the next chapter.

Chapter 4: Implementation

4.1 Introduction

This chapter discussed about the technical details of how the app was built by providing a detailed description of the steps took to build the app. The implementation includes the following subsections: design, development, programming language, app architecture, and features and functionality.

4.2 Environment Installation and Configuration

Tools used in developing Automatic Data Extraction from Blood Test Report are PyCharm as the IDE, Microsoft Azure AI Document Intelligence Studio as the model labelling and training platform, Azure Blob Storage for cloud file storage, and Azure SQL Database as the database.

4.2.1 PyCharm

PyCharm is a Python-IDE developed by JetBrains for the use of performing scripting in Python language. Hence, installation of Python in system is a pre-requisite for PyCharm installation. First, download the pycharm-community-2025.1.exe file from the Community section of the Jetbrains official website showing in Figure 4.1. Then, run and install the file following the instruction given.

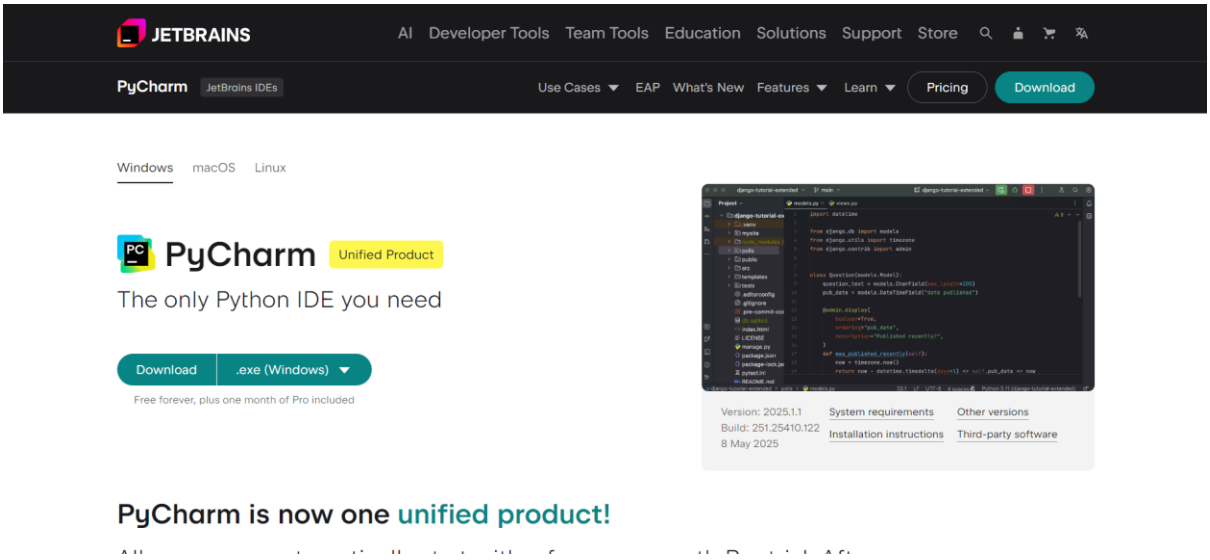


Figure 4.1: PyCharm Official Download Page

Run the pycharm-community-2025.1 application and open the Pycharm Main Menu ribbon bar and click on file button to show create project options. Figure 4.2 shows the screen to create a new project with the name of 'BloodTestReportExtractionApp' in desired file location.

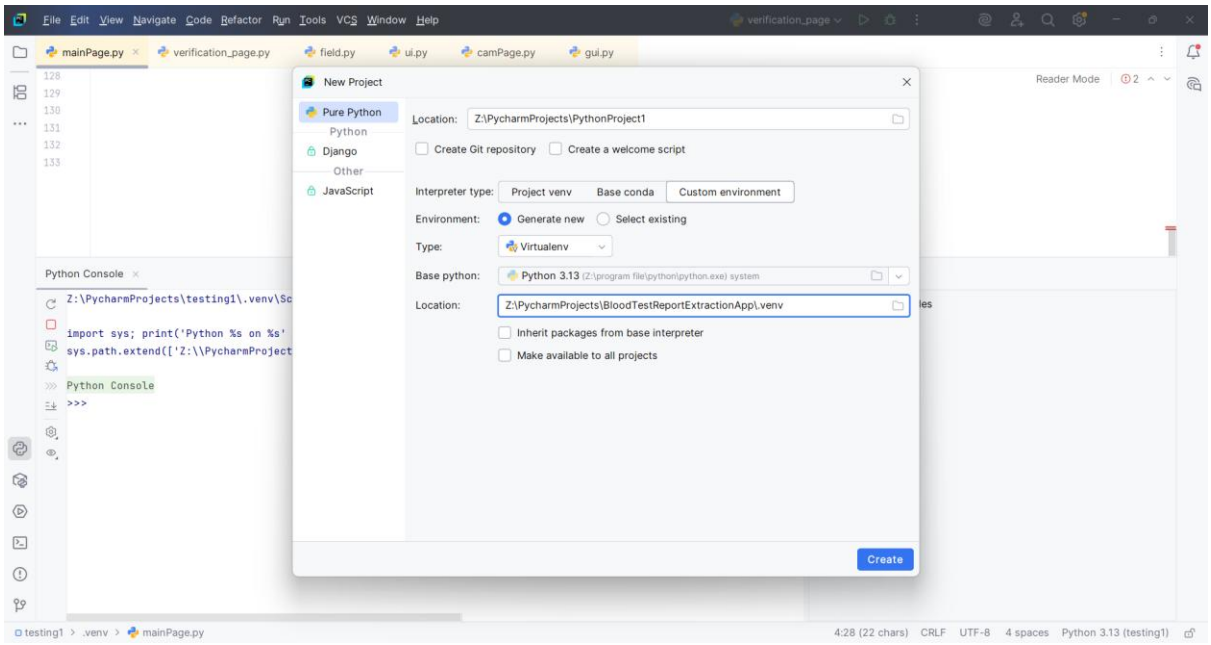


Figure 4.2: Python Create New Project Configuration

Figure 4.3 shows the screen to choose and install the library needed for the application implementation such as azure-core, azure-ai-documentintelligence, kivy, pillow, fitz, and future.

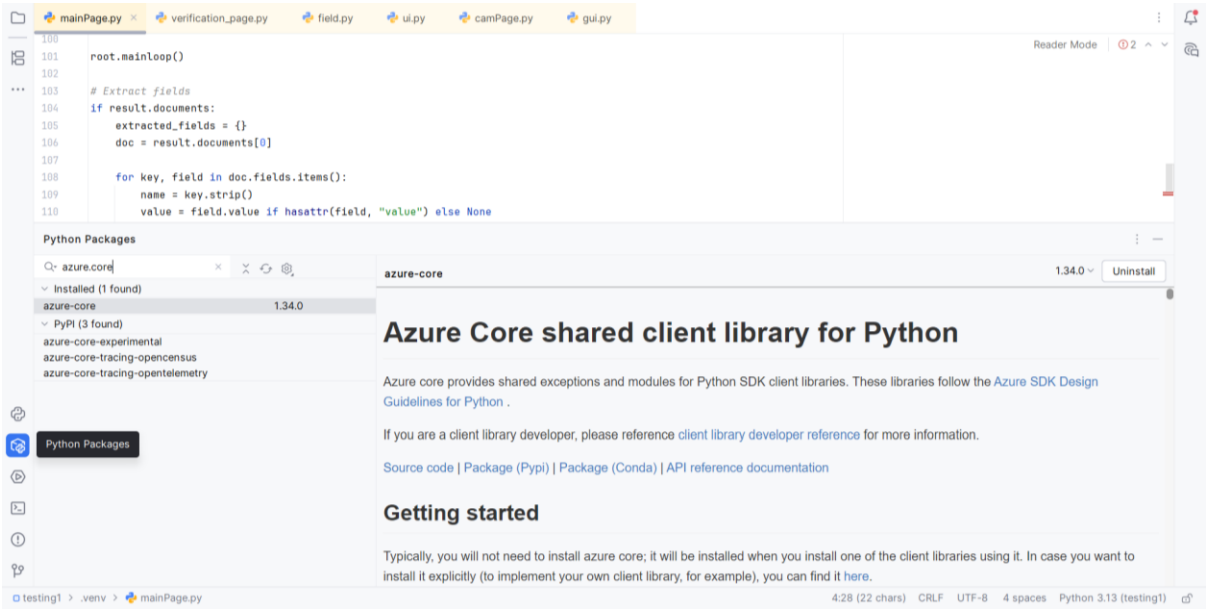


Figure 4.3: Python Package Installation Page

4.2.2 Azure AI Document Intelligence Studio

The studio is an online platform to visually train and integrate features from the Document Intelligence service into the applications (https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/studio-overview?view=doc-intel-4.0.0&tabs=di-studio). An active Azure account and an Azure Storage account with a container to store the training data are needed to access the services. Figure 4.4 shows the screen of Azure AI Document Intelligence Studio main page to select Custom extraction model under Custom model section to initiate the project.

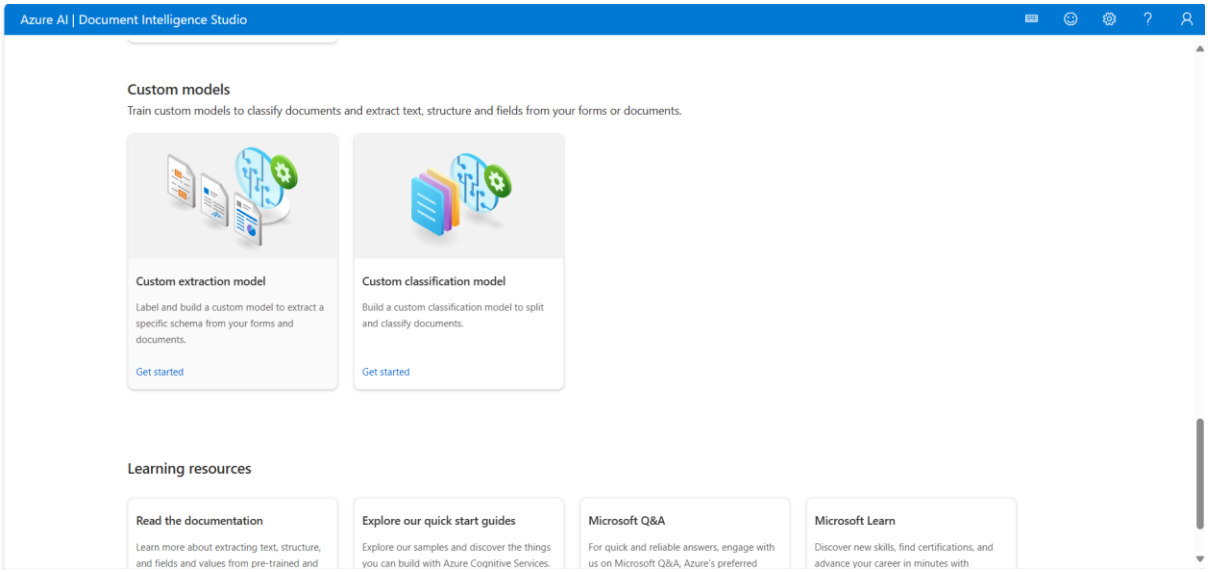


Figure 4.4: Azure AI Document Intelligence Studio Main Page

A new project will be created after confirming the settings such as choose a storage account, specify a container, and provide the path to your documents within the container. Figure 4.5 shows the project which successfully created.

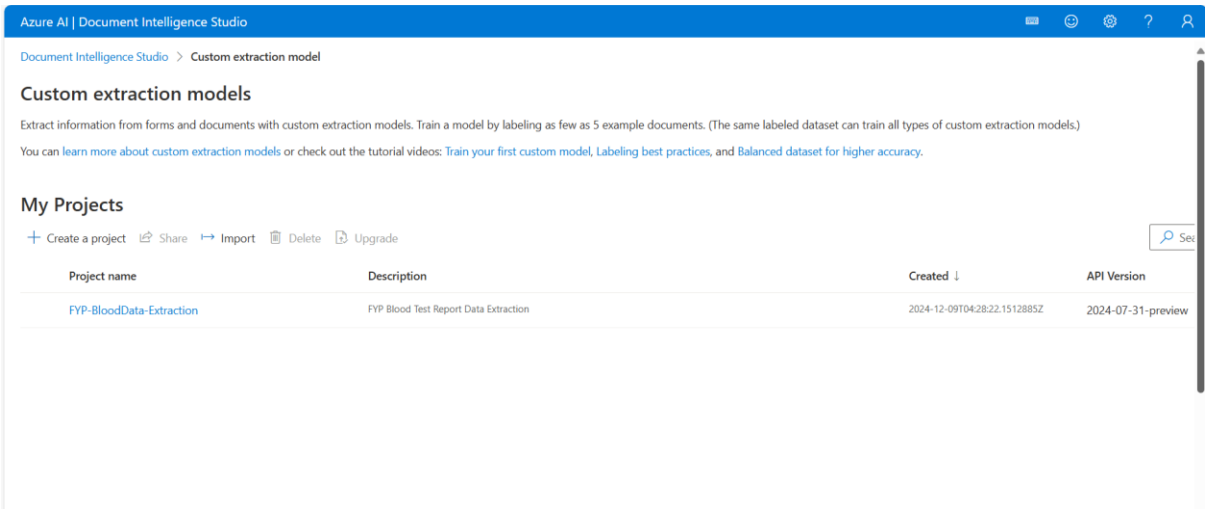


Figure 4.5: My Project Page

Add fields according to the metrics stated in Appendix A and create bounding box to label each of the fields. Figure 4.6 shows the Label data page for data labelling.

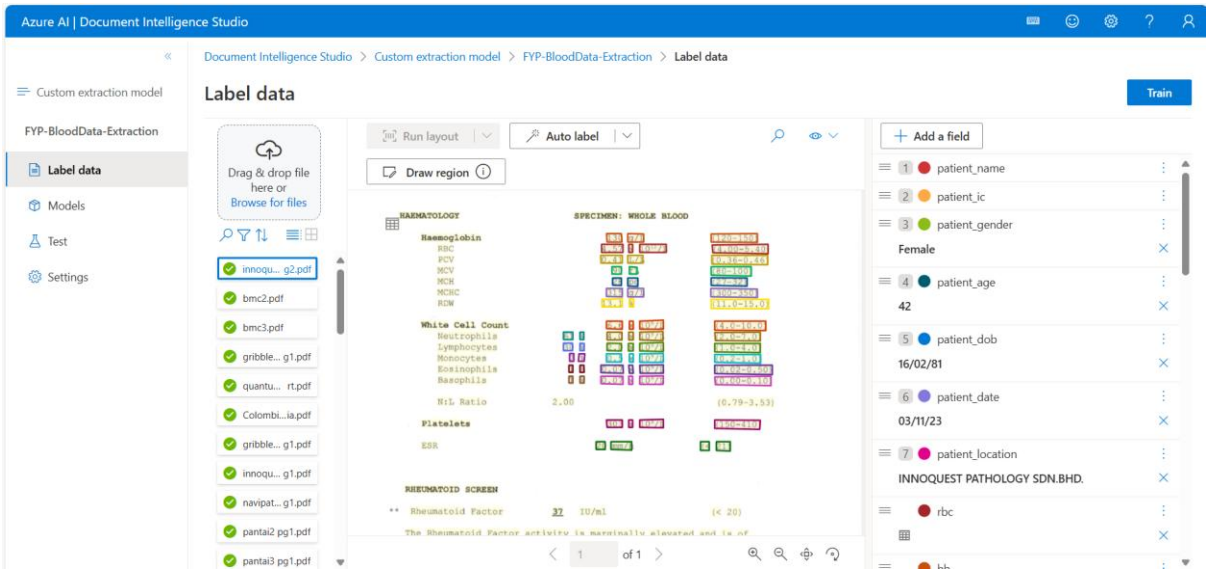


Figure 4.6: Label Data Page

This model consists of 45 blood test report samples with 12 report layouts listed in **Appendix C: List of Blood Test Report Layout Used for Model Training**. After labelling all blood test report samples, click on the 'Train' button to start model training. The model will take at least 0.5 hour for training. Figure 4.7 shows the screen of the models which are successfully trained.

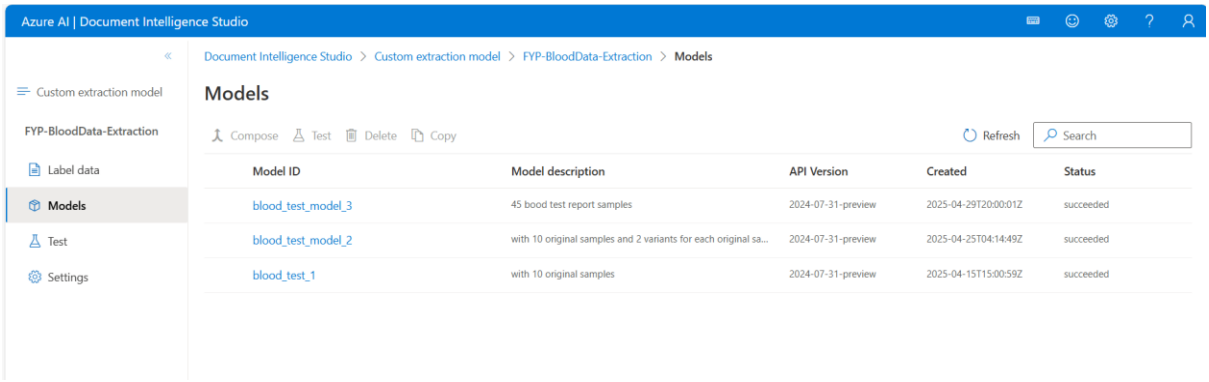


Figure 4.7: Trained Model Page

Figure 4.8 shows the Test page for model testing purposes. The analysis result in JSON format will be shown after selected a blood test report with pdf or image filetype and clicked on ‘Run analysis’ button.

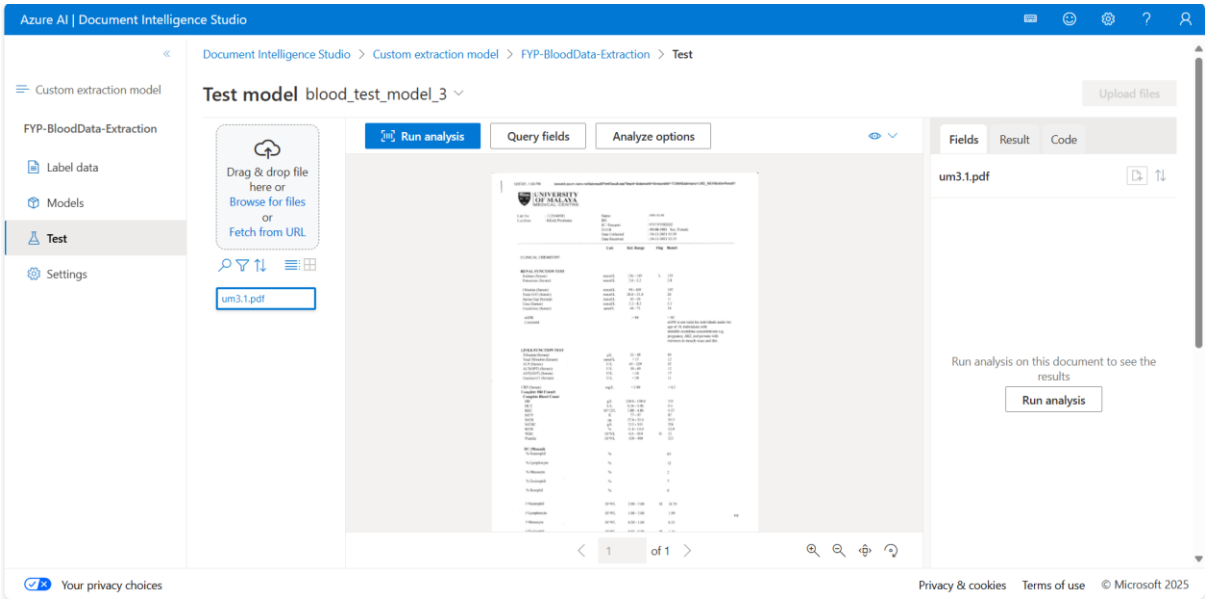


Figure 4.8: Model Test Page

4.2.3 Azure Blob Storage

Azure Blob Storage provides storage to create data lakes for analytics needs (Microsoft, n.d.). It was used to store reference blood test reports to cloud as image and pdf files. A unique link will be generated for the user to retrieve the saved files from the cloud-native blob storage without downloading it to the local storage. Figure 4.9 shows the configuration for creating a blob storage account.

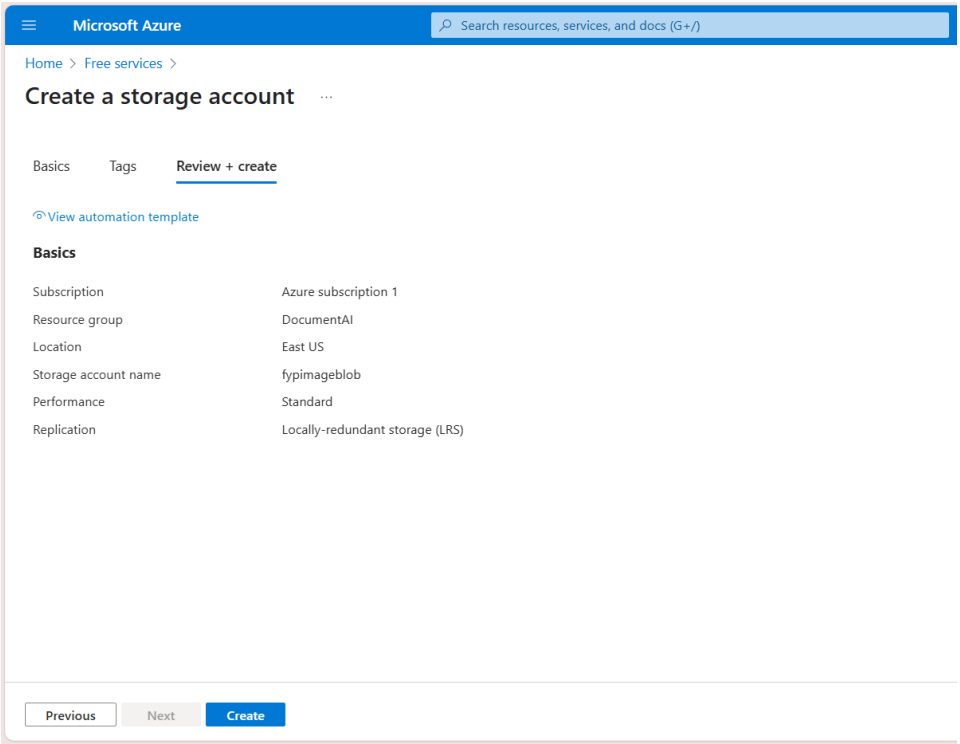
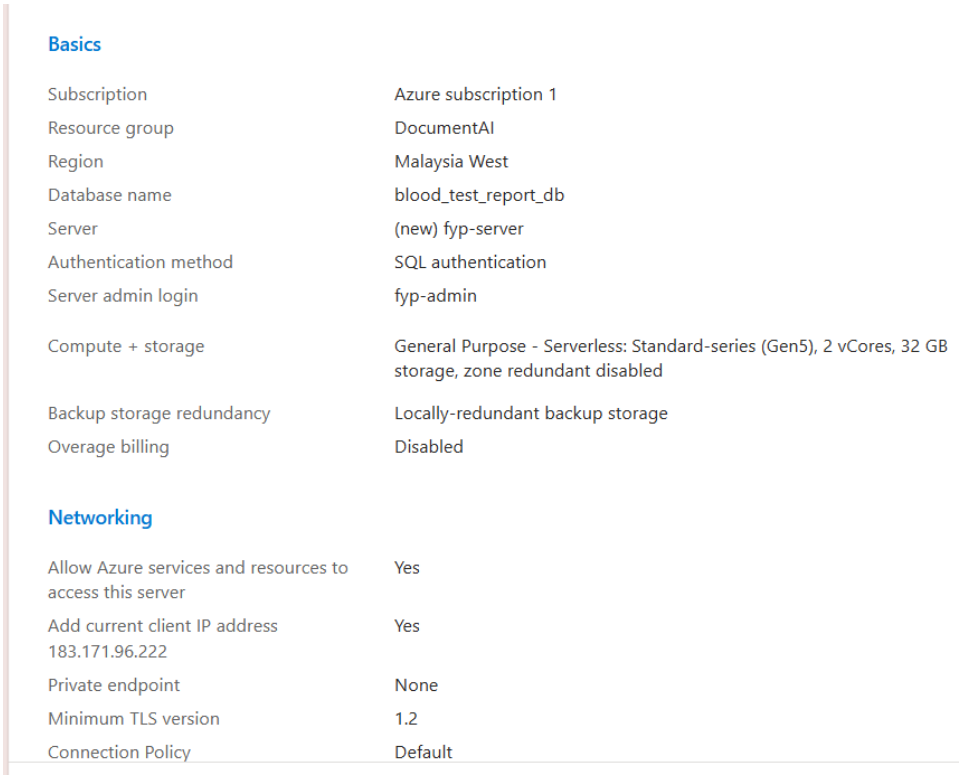


Figure 4.9: Create Blob Storage Account Page

4.2.4 Azure SQL Database

Azure SQL Database is a fully managed relational database-as-a-service (DBaaS) hosted in Azure, built on the Microsoft SQL Server database engine. It offers a highly available and high-performance data storage layer for cloud-based applications (WilliamDAssafMSFT, n.d.). Figure 4.10 and Figure 4.11 show the Basics, Networking, Security, and Additional settings for creating a database using Azure SQL Database.



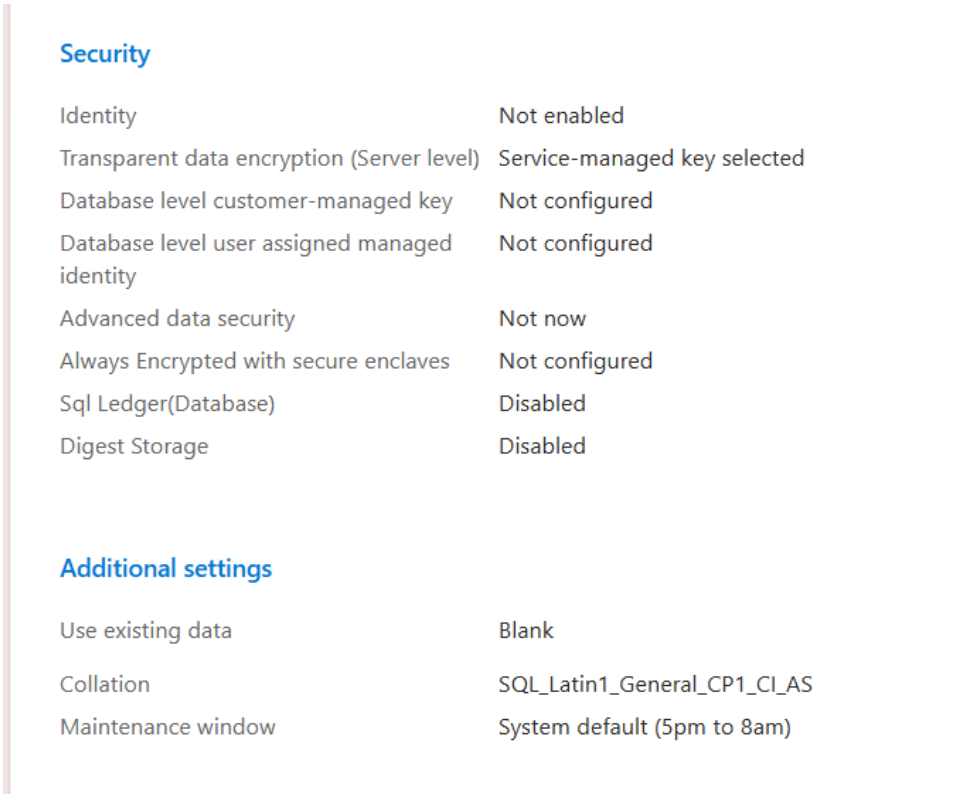
Basics

Subscription	Azure subscription 1
Resource group	DocumentAI
Region	Malaysia West
Database name	blood_test_report_db
Server	(new) fyp-server
Authentication method	SQL authentication
Server admin login	fyp-admin
Compute + storage	General Purpose - Serverless: Standard-series (Gen5), 2 vCores, 32 GB storage, zone redundant disabled
Backup storage redundancy	Locally-redundant backup storage
Overage billing	Disabled

Networking

Allow Azure services and resources to access this server	Yes
Add current client IP address 183.171.96.222	Yes
Private endpoint	None
Minimum TLS version	1.2
Connection Policy	Default

Figure 4.10: Database Basics and Networking Settings



Security

Identity	Not enabled
Transparent data encryption (Server level)	Service-managed key selected
Database level customer-managed key	Not configured
Database level user assigned managed identity	Not configured
Advanced data security	Not now
Always Encrypted with secure enclaves	Not configured
Sql Ledger(Database)	Disabled
Digest Storage	Disabled

Additional settings

Use existing data	Blank
Collation	SQL_Latin1_General_CP1_CI_AS
Maintenance window	System default (5pm to 8am)

Figure 4.11: Database Security and Additional Settings

After reviewing the settings, click on create button to start creating the database. Figure 4.12 shows the total cost summary for creating this database.

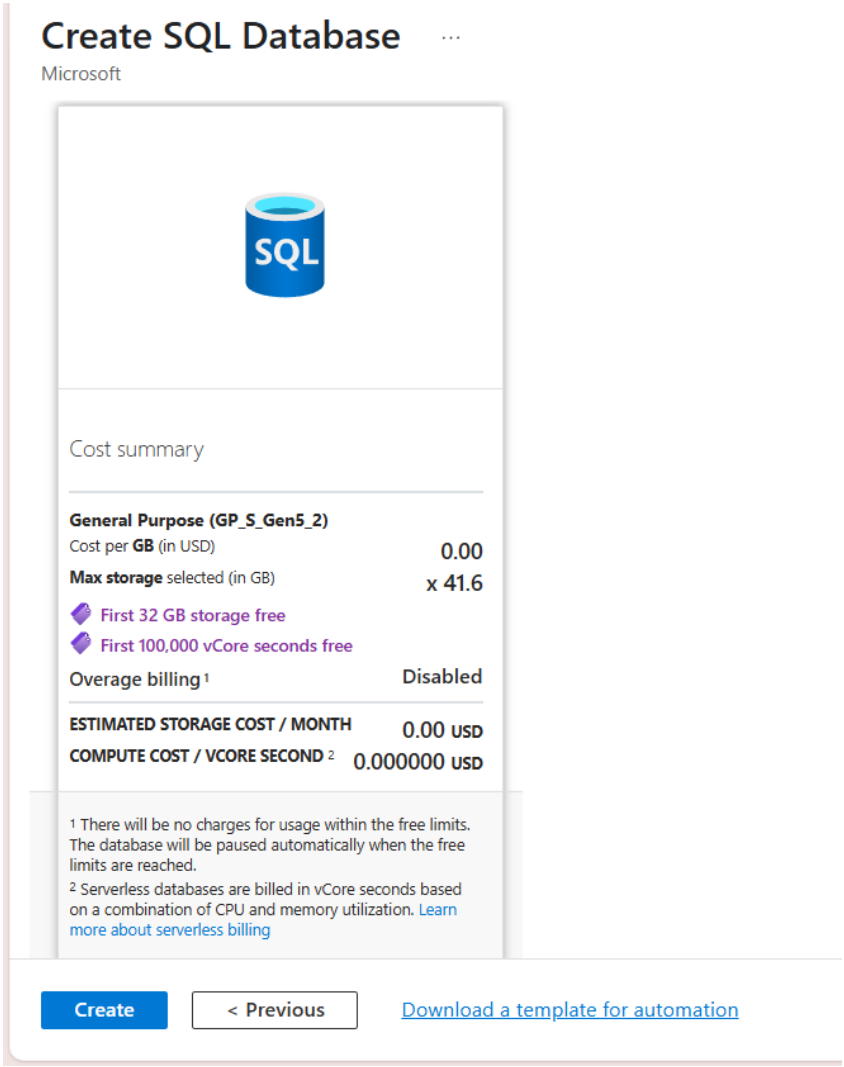


Figure 4.12: Database Cost Summary

4.2.5 Android Studio

Android Studio is an IDE for android mobile application development. It was used to implement the graphical user interface of the proposed application. First, download the android studio exe file from the Android Studio official website showing in Figure 4.13. Then, run and install the file following the instruction given.

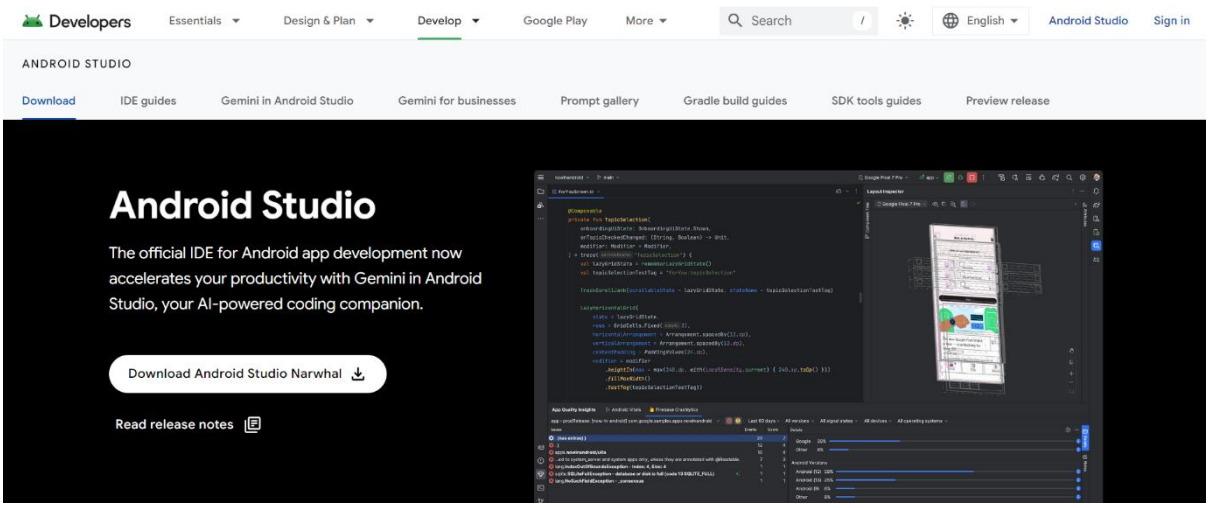


Figure 4.13: Android Studio Official Download Page

Run the android studio application and click on file button to show create project options. Figure 4.14 shows the screen to create a new project start with selecting an activity to create the first java file. Click on Navigation Drawer View Activity, name it as MainActivity and save in in a folder named 'MediExtract'.

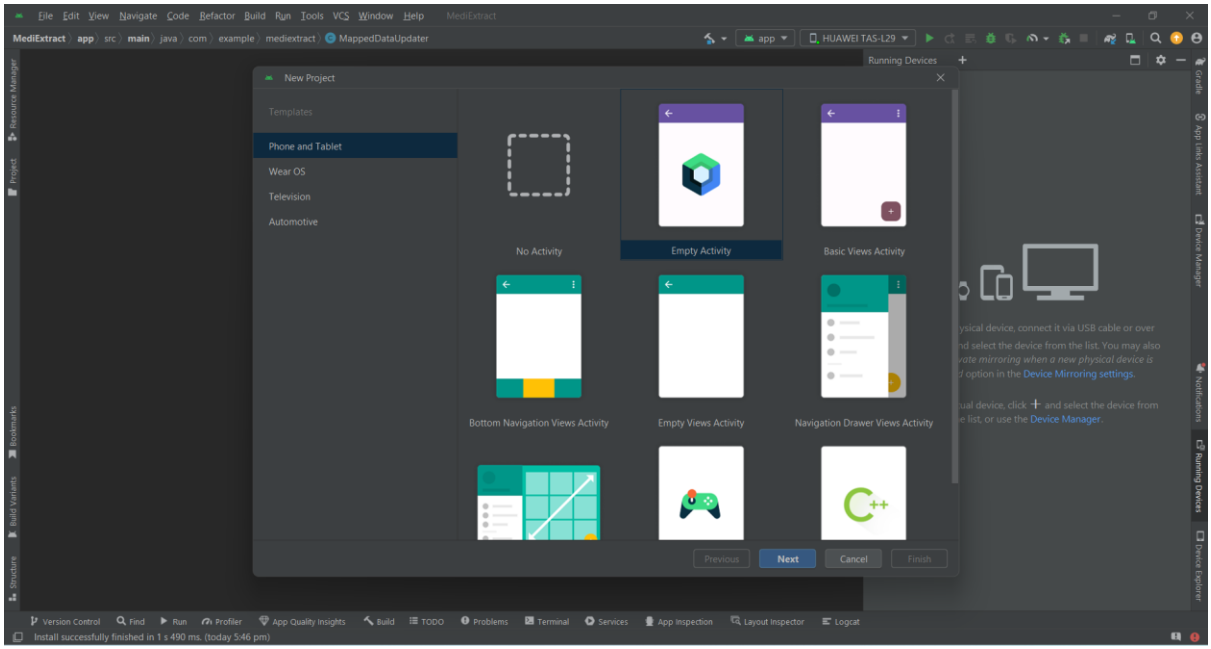


Figure 4.14: Android Studio Create New Project Configuration

4.3 User Roles

There are two different roles or perspectives for the Blood Test Report Extraction Application: Admin and User. These application starts with the User to scan or upload blood test report into the application for data extraction, then the admin could browse through the extracted reports. Below are detailed explanations of each role:

4.3.1 Admin

The main function for admin was to verify user account through the admin main page. Besides, the admin's activities also include view or delete user account details, view the extracted reports, and view own account details.

4.3.2 User

After successful login, the user can start to extract blood test report data through scanning or uploading image or PDF file into the application. The user's activities including Create, Read, Update, and Delete (CRUD) for the created patient details and extracted reports. Users also can visualize the extracted reports based on selected test metrics and month(s). They can also edit or delete their own account where the related reports will be deleted as well.

4.4 Prototype User Interface

In the development phase of Automatic Data Extraction from Blood Test Report application, the application's interface plays a crucial role for user understanding and encourages seamless application utilization. The application interface encompasses various components which enable users to interact with the application and efficiently perform their tasks such as buttons, navigation drawer, images, text views and more. In the following section,

a comprehensive analysis of the interface design for the application will be conducted by exploring various aspects and intricacies of its implementation.

4.4.1 Login Page

Figure 4.13: Login Page of the Blood Test Report Extraction Application

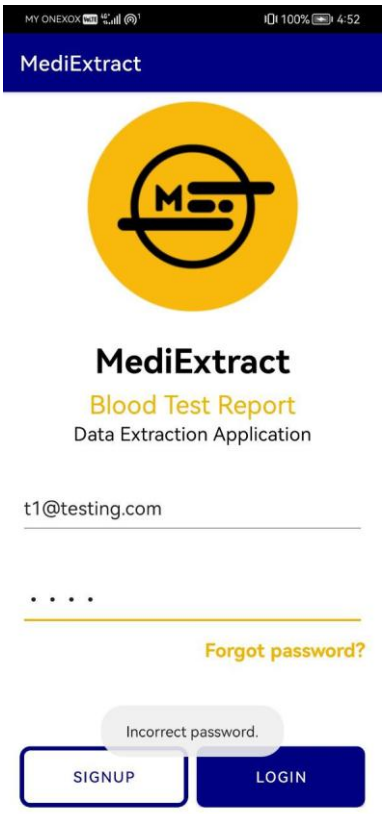


Figure 4.14: Error Popup for Incorrect Email or Password

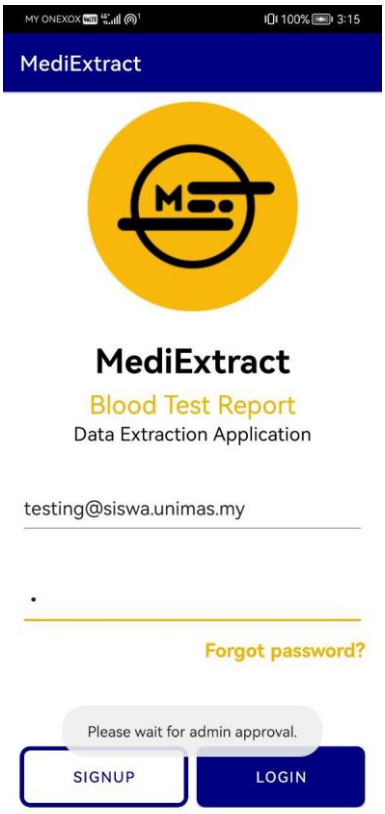


Figure 4.15: Error Popup for Unverified User Account

The application will display the login page as the landing page as shown as Figure 4.13. The user for both roles will need to login to their account through this page. Error popup will display when the input for email or password registered is/are not correct as shown as Figure 4.14. Figure 4.15 shows the error popup if the user account has not been verified by the administrator. The user can tap on the 'Sign Up' button to navigate to the Sign-Up Account Page. Besides, the user can tap on the 'Forgot Password' phrase to navigate to the Reset Password Page.

4.4.2 Sign Up Account Page

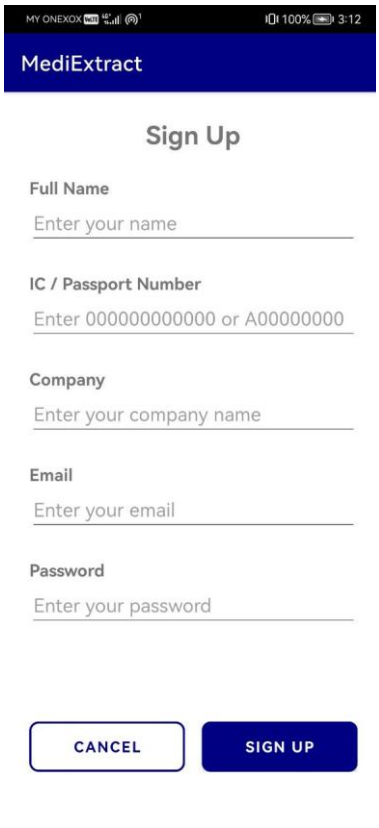


Figure 4.16: Sign Up Account Page of the Blood Test Report Extraction Application

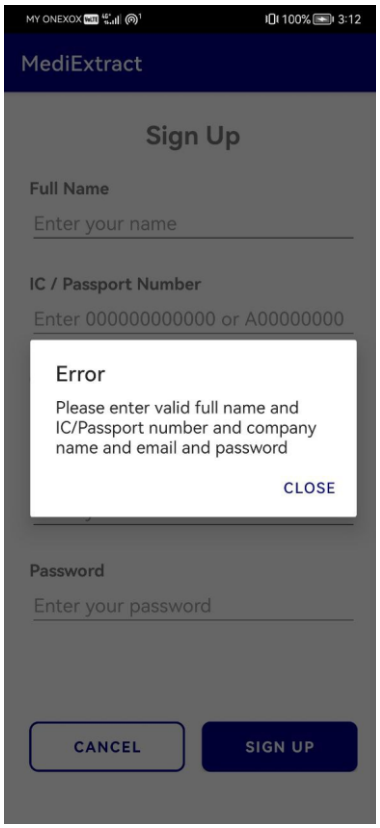


Figure 4.17: Error Popup for Incorrect Input

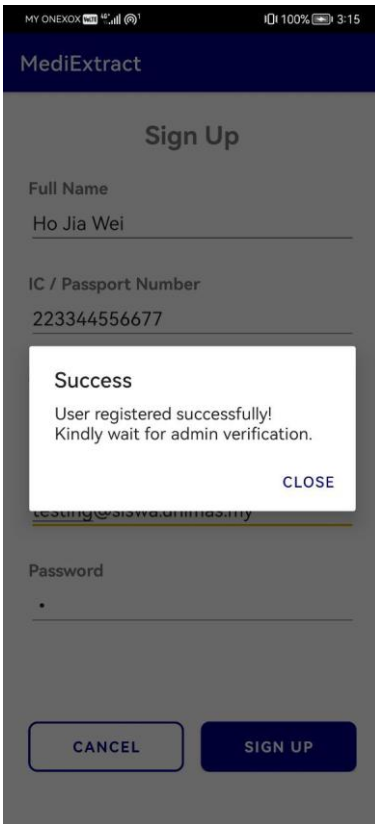


Figure 4.18: Popup for Successful User Account Registration

The user will need to fill up the information needed including full name, IC or Passport number, company or organization name, email and password to register a new user account as shown as Figure 4.16. Error popup will display when the signup inputs is/are blank as shown as Figure 4.17. Figure 4.18 shows the popup displayed for successful user account registration. The user can only start login into their account once the account had been verified by the administrator.

4.4.3 Forgot Password Page

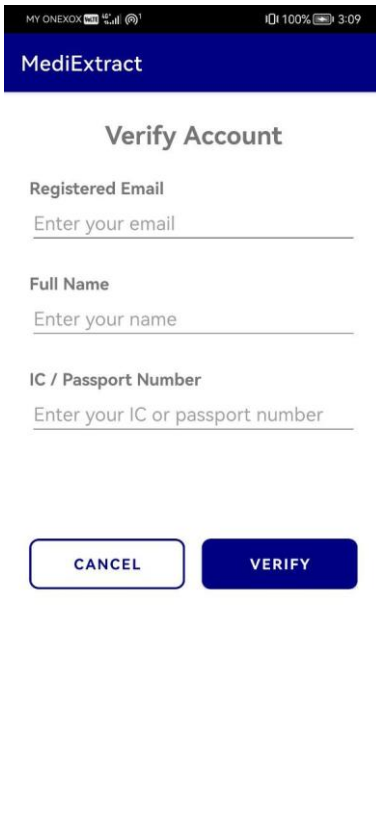


Figure 4.20: Reset Password Page of the Blood Test Report Extraction Application

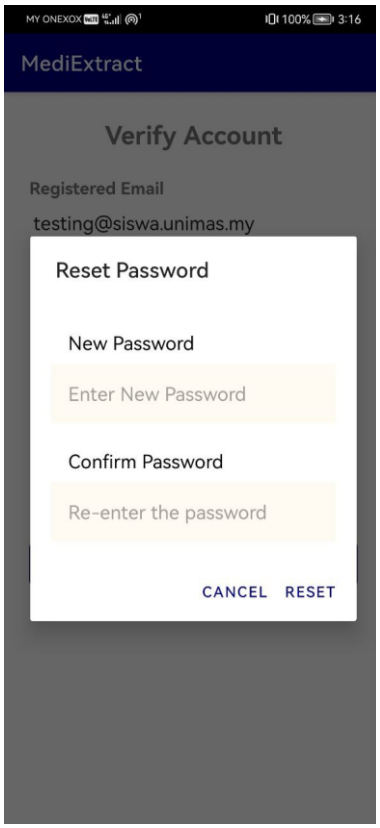


Figure 4.21: Reset Password Popup

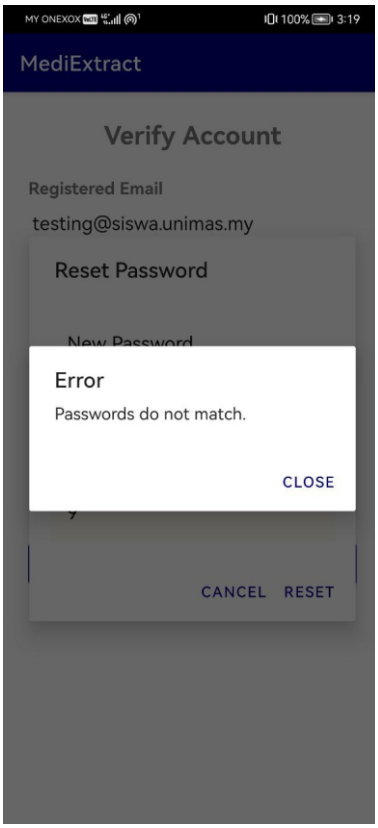


Figure 4.22: Error Popup for Incorrect or Unmatched Information

The user will need to fill up the information needed including full name, IC or Passport number, company or organization name, email that had been used during user sign-up to verify their account as shown as Figure 4.20. Reset password popup will display when the inputs are completely matched with the sign-up information as shown as Figure 4.21. Figure 4.22 shows the error popup displayed when there exists input which not matched with the sign-up information.

4.4.4 Admin Main Page

Figure 4.23: Admin Main Page of the Blood Test Report Extraction Application

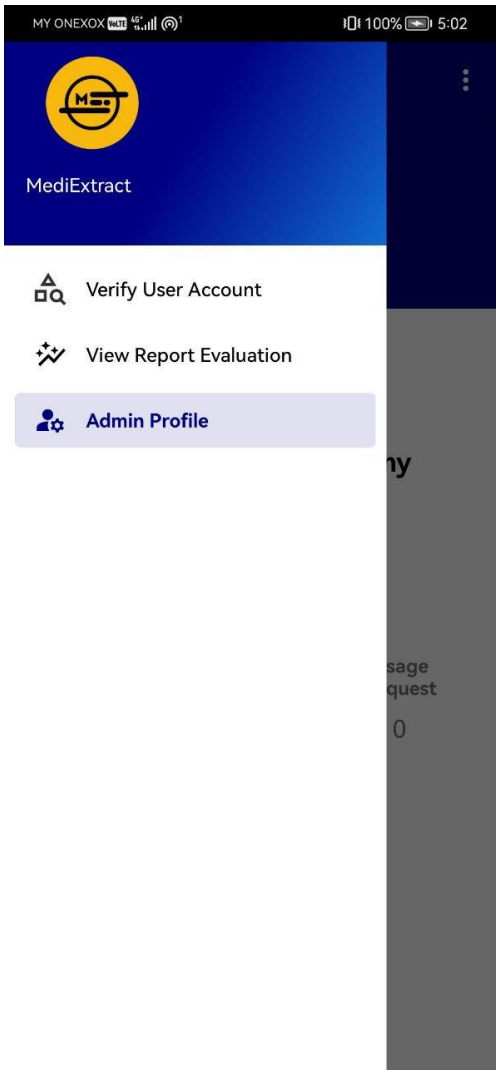


Figure 4.24: Admin Side Bar Menu of the Blood Test Report Extraction Application

After successful login, the user with 'admin' role, which is the admin, will be navigate to the admin main page as shown as Figure 4.23. Figure 4.24 shows the side bar menu for admin page. The admin can navigate to other page by tapping on the options showing in this side bar menu. For example, tapping on 'Verify User Account' button to verify new user accounts. They can also tap on 'View Report Evaluation' button to search and view

the report created by other users. The ‘Admin Profile’ button will be the main page showing after successful login.

4.4.5 Verify User Account Page

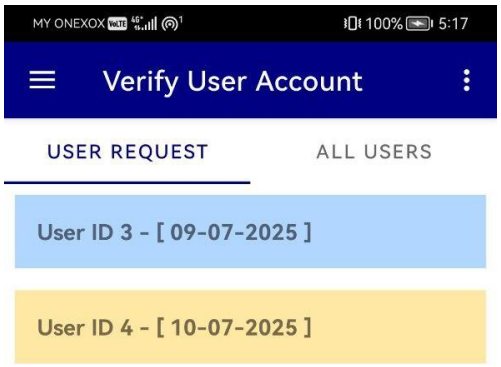


Figure 4.25: Verify User Account Page of the Blood Test Report Extraction Application (Unverified Accounts)

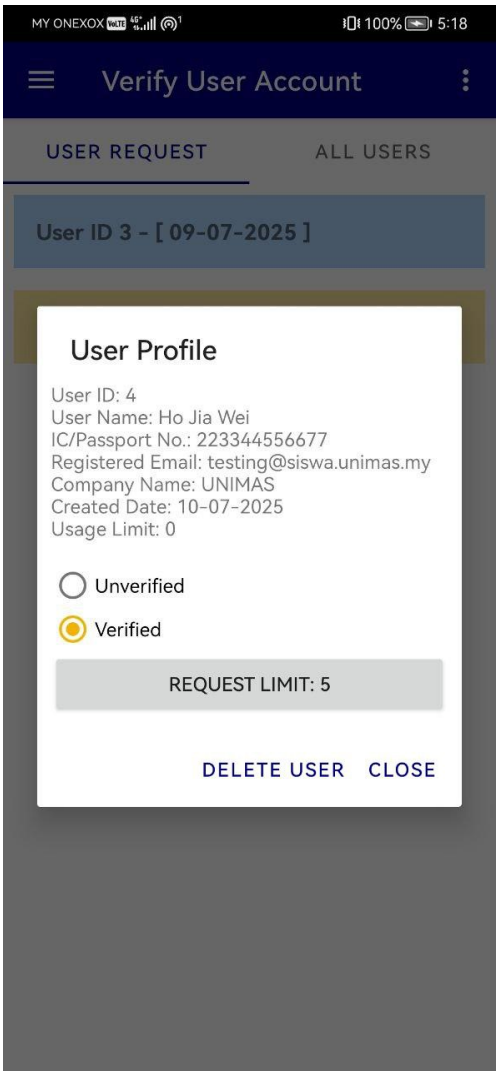


Figure 4.26: User Account Details Popup of Selected User

The admin will be navigated to the Verify User Account Page after tapping on the ‘User Verification’ button from Admin Main Page. The admin can tap on ‘User Request’ tab to show the unverified user in light blue button and user requested for extra usage limit in

light orange button as shown as Figure 4.25. Figure 4.26 shows the popup displayed when the admin taps on the unverified user light blue button. This popup consists of the user account details of the selected user. Admin can delete or verify the selected user account by tapping on respective buttons showing on the popup. The ‘Request Limit’ button will show the number of extra usages requested by the user. A popup will appear for the admin to approve the request after tapping on this button.

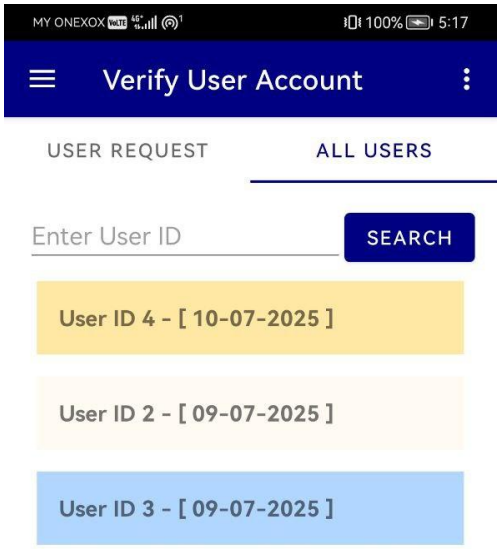


Figure 4.27: Verify User Account Page of the Blood Test Report Extraction Application (All Users)

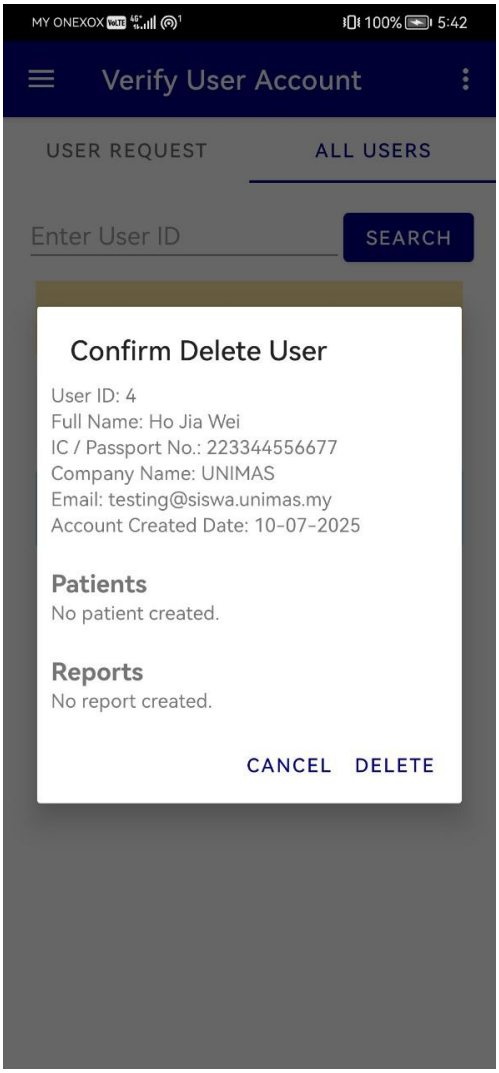


Figure 4.28: Confirm Delete User Account Popup

When the admin taps on 'All Users' tab, all the user accounts will be displayed. The beige colour button indicates the verified user shown as Figure 4.27. Figure 4.28 shows the popup displayed when the admin taps on the delete user button. This popup consists of the user account details including the patient and report created of the selected user.

4.4.6 Search Report Page

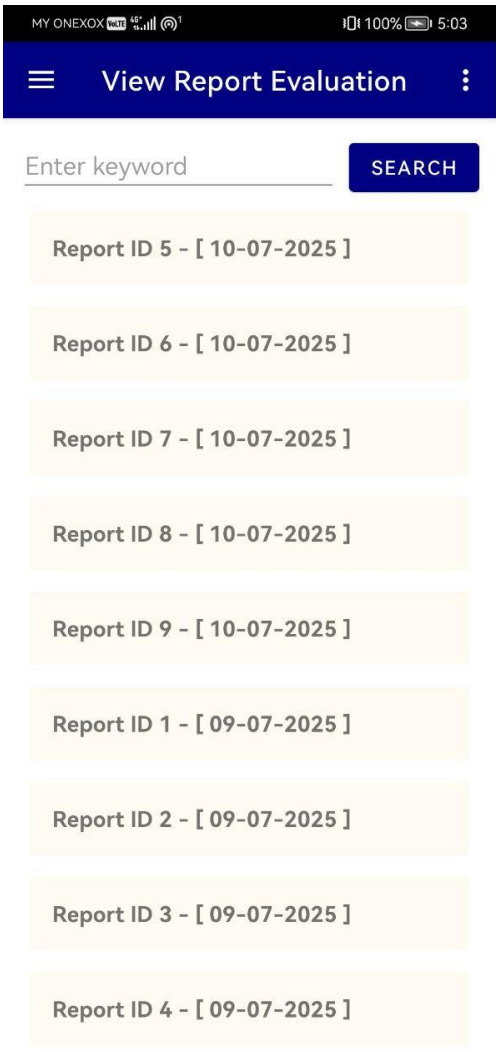


Figure 4.29: Search Report Page of the Blood Test Report Extraction Application

The admin will be navigated to the Search Report Page after tapping on the 'View Report Evaluation' button from Admin Main Page. The admin can enter Report ID and tap on the search button to search for report as shown as Figure 4.29. The result will be displayed in the form of light beige button with information including Report ID and report created date. Admin can tap on the desired light beige button to view the report.

4.4.7 Report Evaluation Page

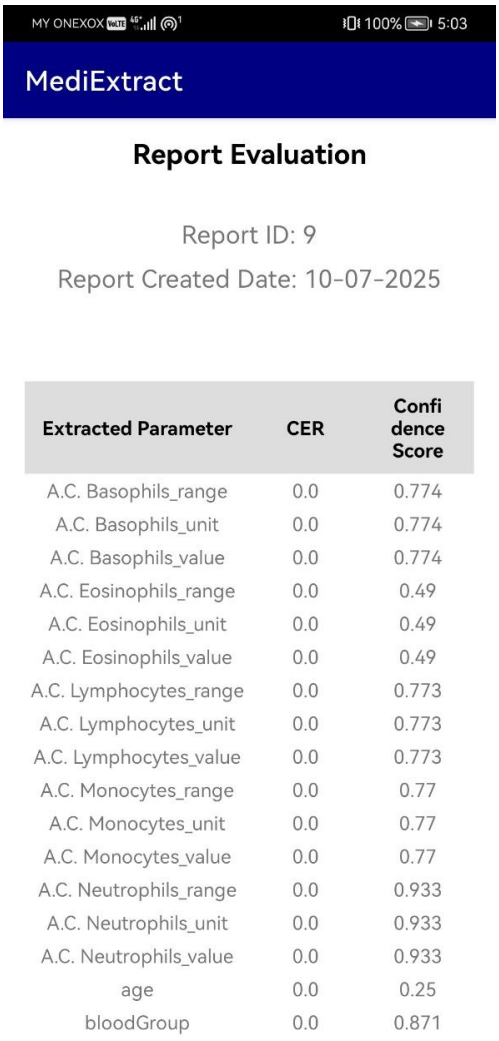


Figure 4.30: Report Evaluation Page of the Blood Test Report Extraction Application

Figure 4.30 shows the Report Evaluation Page when the admin had tapped on the light beige report button from Search Report Page. From the selected report, the admin can view the CER and confidence score of each of the extracted label.

4.4.8 Logout Interface

Figure 4.31: Logout Button of the Blood Test Report Extraction Application

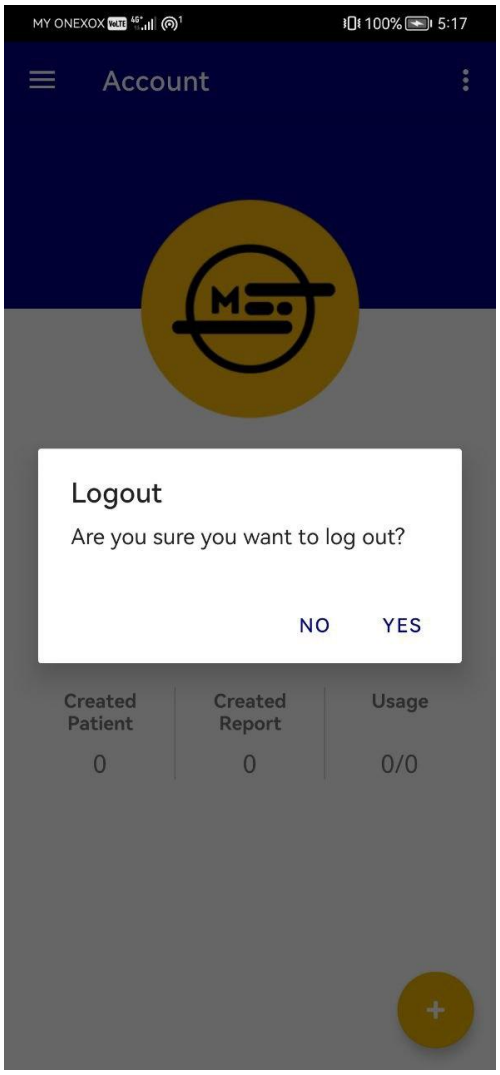


Figure 4.32: Confirm Logout Popup of the Blood Test Report Extraction Application

There will be an icon with 3 dots align in vertical placed at the top right corner of each page. This icon is for the user to logout from their account. By tapping on the 'Logout' option showing in Figure 4.31, the confirm logout popup will appear as shown in Figure 4.32. The user will be navigated to the Login Page after successful logout.

4.4.9 User Main Page

Figure 4.33: User Main Page of the Blood Test Report Extraction Application

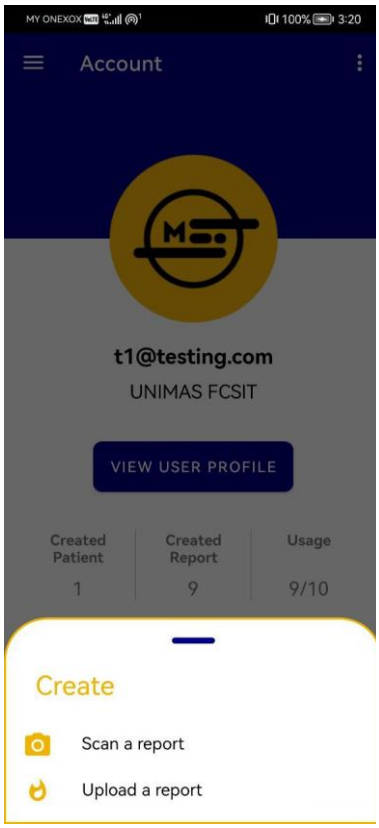


Figure 4.34: Bottom Dialog of Create Button

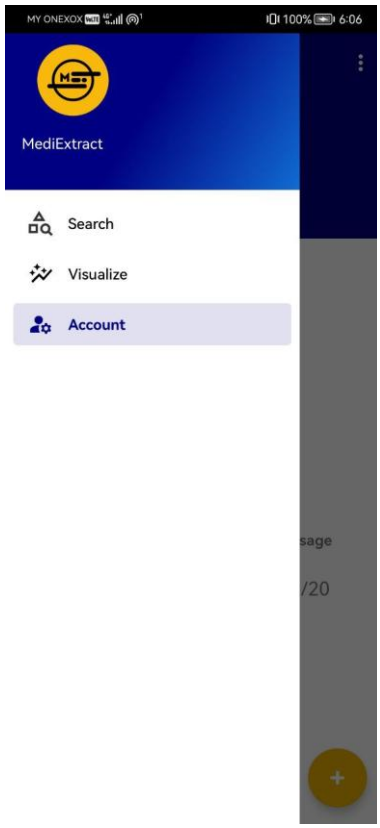


Figure 4.35: User Side Bar Menu of the Blood Test Report Extraction Application

After successful login, the user with 'user' role will be navigate to the User Main Page as shown as Figure 4.33. The user can tap on the create button, the button in round orange colour with a '+' icon, to scan or upload blood test report. Figure 4.34 shows the bottom dialog for the user to choose to scan or upload the blood test report. Camera permission will be granted as the user will use their own device camera function to scan the blood test report. Besides, document storage permission also will be granted as the user will need to upload blood test report from their device storage.

Besides, they can also tap on 'Search' button from the side bar menu showing in Figure 3.35 to search and view the patient details created by themselves. The 'Account' button

allows the user to reset account password, view, edit or delete their user profile. Besides, they can also tap on ‘Visualize’ button to view visualization based on the extracted report metrics.

```
private void uploadToAzure(File file) {
        blockingDialog.show("analyzing");

        String ip = getString(R.string.server_ip);

        if (!file.exists()) {
            Log.e("AzureUpload", "File does not exist: " + file.getAbsolutePath());
            Toast.makeText(this, "File not found: " + file.getAbsolutePath(), Toast.LENGTH_LONG).show();
            return;
        }

        OkHttpClient client = new OkHttpClient.Builder()
                .connectTimeout(60, TimeUnit.SECONDS)
                .writeTimeout(60, TimeUnit.SECONDS)
                .readTimeout(180, TimeUnit.SECONDS)
                .build();

        String fileName = file.getName().toLowerCase();
        String mimeType;

        RequestBody fileBody = RequestBody.create(file, MediaType.parse(mimeType));

        MultipartBody requestBody = new MultipartBody.Builder()
                .setType(MultipartBody.FORM)
                .addFormDataPart("file", file.getName(), fileBody)
                .build();

        Request request = new Request.Builder()
                .url("http://"+ip+"/analyze")
                .post(requestBody)
                .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                blockingDialog.hide();
                Log.e("AzureUpload", "Upload failed: " + e.getMessage());
                e.printStackTrace();
            }
```

```
        @Override
        public void onResponse(Call call, Response response) throws IOException {
            if (response.isSuccessful()) {
                blockingDialog.hide();

                String responseData = response.body().string();

                try {
                    JSONObject json = new JSONObject(responseData);

                    boolean success = json.getBoolean("success");
                    if (success) {
                        JSONObject mappedData = json.getJSONObject("data");

                        String mappedDataString = mappedData.toString();
                        String cleanedDataString = cleanICField(mappedDataString);

                        Uri fileUri;
                        fileUri = FileProvider.getUriForFile(
                                    MainActivity.this,
                                    getPackageName() + ".provider",
                                    file
                                );

                        Intent intent = new Intent(MainActivity.this, ActivityExtraction.class);
                        intent.putExtra("imgPath", fileUri .toString());
                        intent.putExtra("filePath", file.getAbsolutePath());
                        intent.putExtra("mappedData", cleanedDataString);

                        runOnUiThread(() -> startActivity(intent));

                    } else {
                        Log.e("AzureUpload", "Error from server: " + json.getString("message"));
                        runOnUiThread(() -> Toast.makeText(getApplicationContext(), "Server error", Toast.LENGTH_SHORT).show());
                    }

                } catch (JSONException e) {
                    Log.e("AzureUpload", "JSON parsing failed: " + e.getMessage());
                    e.printStackTrace();
                }
            } else {
                Log.e("AzureUpload", "Server error: " + response.code() + " - " + response.message());
                runOnUiThread(() -> Toast.makeText(getApplicationContext(), "Server error", Toast.LENGTH_SHORT).show());
            }
        }
    });
}
```

Figure 4.36: Code Snippet of Analyze Document Function

Figure 4.36 shows the upload to azure function that parse the scanned blood test report to the Flask API and return analysis result in JSON format. The JSON result will be stored in string and parse to the Verification Page.

```python
@app.route('/analyze', methods=['POST'])
def analyze_document_api():
    try:
        if 'file' not in request.files:
            return jsonify({"success": False, "message": "No file uploaded."}), 400

        file = request.files['file']
        filename = secure_filename(file.filename)
        file_path = os.path.join("uploads", filename)
        os.makedirs("uploads", exist_ok=True)
        file.save(file_path)

        # Log file saved
        print(f"File saved at: {file_path}")

        # Azure Document Intelligence API
        with open(file_path, "rb") as f:
            poller = client.begin_analyze_document(
                model_id=AZURE_MODEL_ID,
                body=f
            )
            result = poller.result()

        raw_response = poller.polling_method()._pipeline_response
        response_text = raw_response.http_response.text()
        raw_json = json.loads(response_text)

        documents = raw_json.get("analyzeResult", {}).get("documents", [])
        if not documents:
            return jsonify({"success": False, "message": "No documents found."}), 500

        fields_raw = documents[0].get("fields", {})
        azure_data = {k: v if isinstance(v, dict) else {} for k, v in fields_raw.items()}
        mapped_data = transform_extracted_fields(azure_data)

        return jsonify({"success": True, "data": mapped_data})

    except Exception as e:
        print("Server Exception:", e)
        return jsonify({"success": False, "message": str(e)}), 500
```

Figure 4.37: Code Snippet of Analyze Document Function

With correct azure credentials, the Flask API can obtain secured connection with Microsoft Azure AI Document Intelligence custom extraction model API. Figure 3.47 shows the code snippet of the flask API to analyze and clean the JSON result return back from Microsoft Azure AI Document Intelligence to obtain needed key value pairs. The cleaned data will be parse back to Verification Page for populating the values into the edit fields.

The transform extracted field function processes the analysis result from the JSON file and converts it into a standardized dictionary format. It begins by defining the normalize

unit function to remove spaces from unit strings to ensure consistency. The main part of the function iterates through the azure_data dictionary, which contains fields extracted from the blood test report. If a key corresponds to one of the predefined patient detail fields (like name, gender, or blood type), its value is extracted and mapped to a label defined in self.label_map, along with its confidence score. For fields in array type, the function navigates nested structures to extract values such as result, unit, and reference range, normalizes them, and stores them in the mapped_data dictionary using readable labels.

4.4.10 Verification Page

Figure 4.38: Verification Page of the Blood Test Report Extraction Application

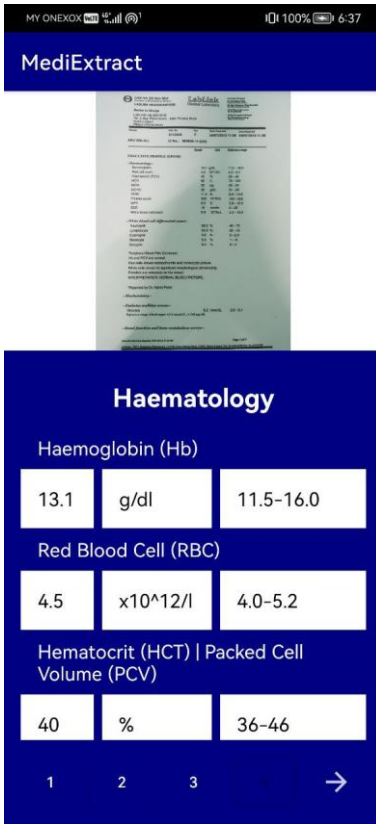


Figure 4.39: Verification Page Showing Page 2 Haematology Test Data

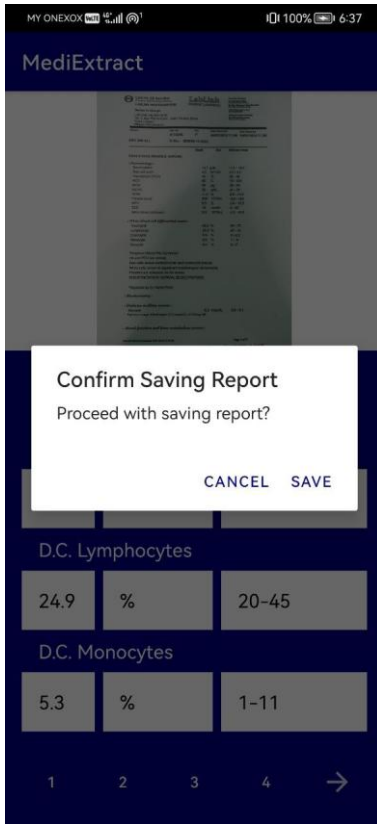


Figure 4.40: Confirm Saving Popup of Verification Page

The user will be navigated to the Verification Page after scanning or uploading a blood test report. The captured image will be displayed at the upper part of the screen while the bottom part will be showing the extracted data as shown as Figure 4.38. The users could zoom and pan to view the image displayed in detail by dragging the image. For the bottom part of the screen, it will display the extracted data in 4 pages with title respectively, 'Patient Details' for the first page, 'Haematology' for the second page, 'Absolute Count' for the third page, and 'Differential Count' for the last page. The content for each page will be shown in Appendix D. Figure 4.39 had shown the layout of edit fields for each blood test parameters. The users are required to verify all pages before saving the data to the cloud database. Besides, page flow control had been applied for the Verification Page to force the user to visit all 4 pages as the next page button will only be enabled when previous page had been visited. The next button showing in a arow icon will only be enabled after the user visited all 4 pages. Figure 4.40 shows the confirm saving report popup when the user taps on the next button.

```
public class MappedDataUpdater {

    public static JSONObject overrideMappedDataWithUpdates(JSONObject mappedData, Map<String, String> updatedData) {
        JSONObject updatedMappedData = new JSONObject();

        try {
            Iterator<String> keys = mappedData.keys();
            while (keys.hasNext()) {
                String key = keys.next();
                updatedMappedData.put(key, mappedData.get(key));
            }

            for (Map.Entry<String, String> entry : updatedData.entrySet()) {
                String key = entry.getKey();
                String updatedValue = entry.getValue().trim();

                String originalValue = mappedData.optString(key, "").trim();

                updatedMappedData.put(key, updatedValue);

                if (!updatedValue.isEmpty()) {
                    double cer = CerCalculator.calculateCharacterErrorRate(updatedValue, originalValue);
                    updatedMappedData.put(key + "_cer", cer);
                } else {
                    updatedMappedData.put(key + "_cer", JSONObject.NULL);
                }
            }

        } catch (JSONException e) {
            e.printStackTrace();
        }

        return updatedMappedData;
    }

}
```

Figure 4.41: Code Snippet of Mapped Data Adapter Class

Figure 4.41 shows the mapped data adapter Class that will update the verified key field data from the edit field of Verification Page. It will return the updated data in JSONObject form with the character error rate (CER) of each edited field. The CER value was calculated by comparing the original data return from the Microsoft Azure AI Intelligence API with the data verified by the user.

4.4.11 View Report Page

Figure 4.42: View Report Page of the Blood Test Report Extraction Application

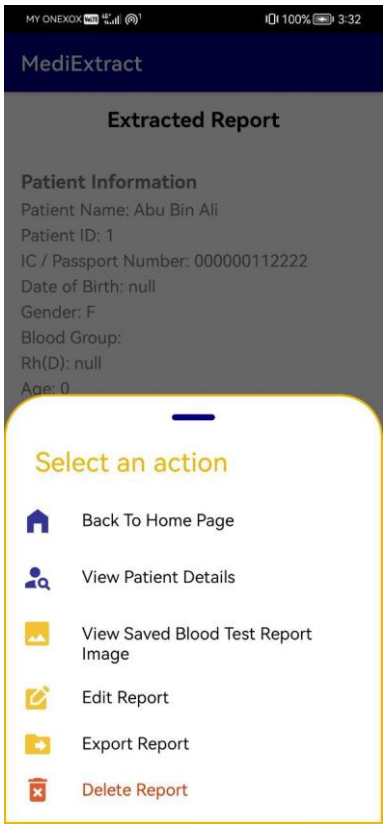


Figure 4.43: Bottom Dialog of View Report Page

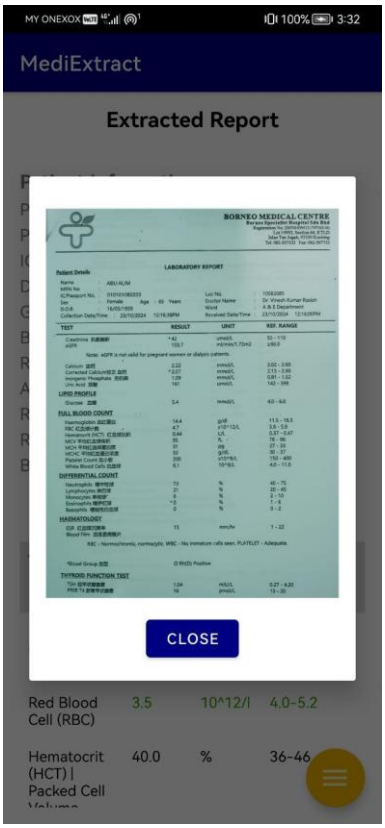


Figure 4.44: Original Report Preview Popup

The user will be navigated to the View Report Page after successfully saved the verified extracted data from Verification Page. The user can review, edit, export or delete the extracted report as shown as Figure 4.42. Figure 4.43 shows the action that can be done on the extracted report including view original scanned report, edit, export and delete report. If the user tap on 'View Saved Blood Test Report Image', the image will be retrieved from Azure Blob Storage and display in a popup as shown as Figure 4.43. Users can zoom or pan the image for clearer viewing.

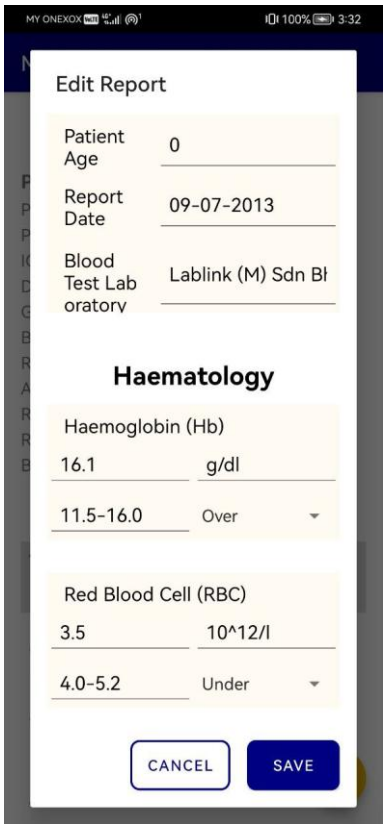


Figure 4.45: Edit Report Popup

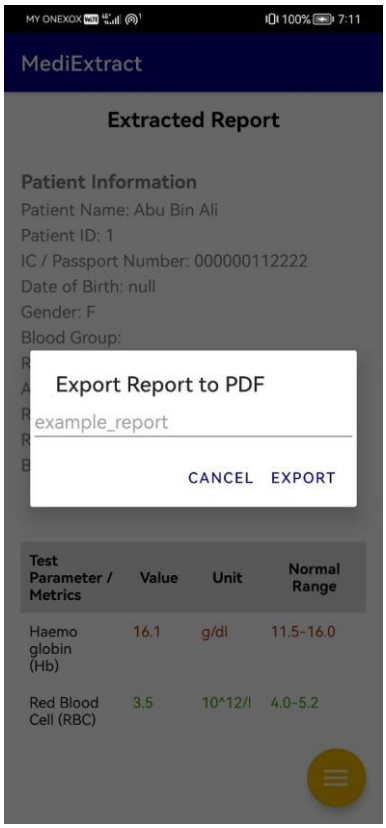


Figure 4.46: Export Report Popup

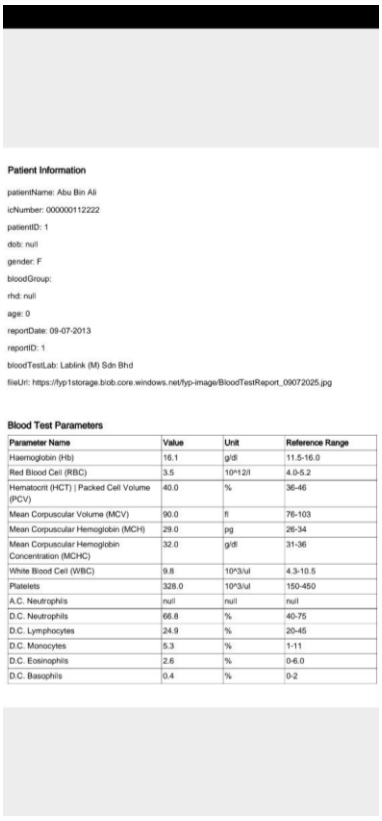


Figure 4.47: PDF Layout of Exported Report

Figure 4.45 shows the Edit Report Popup when the user chooses to edit the report by tapping on the 'Edit Report' button from the bottom dialog. Figure 4.46 shows the popup prompting the user to insert a filename for exporting extracted report as PDF. After inserting a filename for report exporting, the report will be exported in the form of PDF file. Figure 4.47 had shown the layout of the exported report. The patient information will be showing at the top section of the report and followed by the blood test data. The blood test data section consists of all blood test parameters defined in **Appendix A: Blood Test Report Health Metrics**. Each blood test parameter will have their own extracted value, unit, and reference range.

4.4.12 Search Patient Page

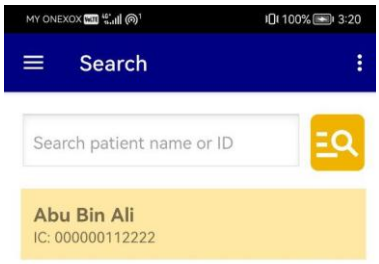


Figure 4.48: Search Patient Page of the Blood Test Report Extraction Application

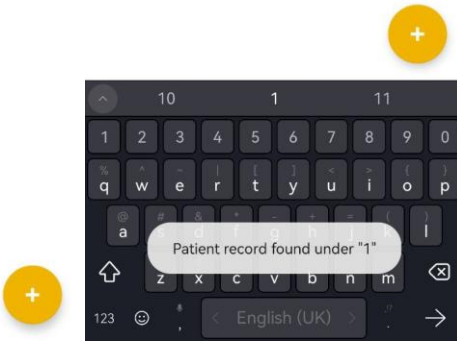


Figure 4.49: Search Patient Page with Matched Result Found

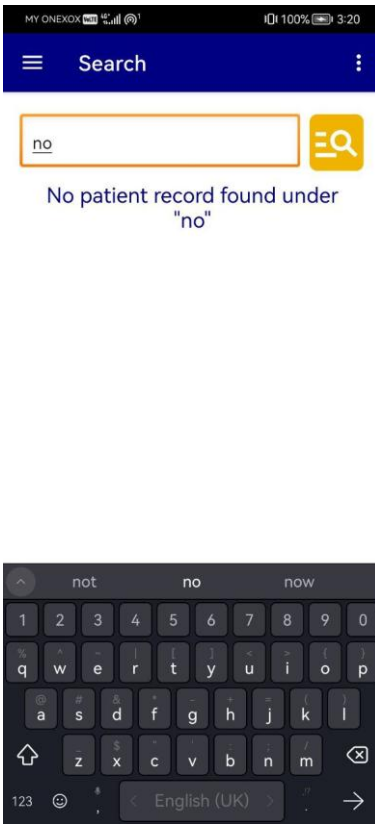


Figure 4.50: Search Patient Page with No Matched Result Found

The user will be navigated to the Search Patient Page after tapping on the 'Patient Profile' button from User Main Page. The user can search for specific patient which created by themselves by patient name or patient ID as shown as Figure 4.48. After entering the valid input, the user will need to tap on the button with magnifying glass icon positioned at the tip right corner of the screen. Figure 4.49 shows the page view of search patient by Patient ID with matched result found where Figure 4.50 shows the page view of search patient by patient name with no matched result found. The matched result will be displayed in the form of light orange colour button for each patient. Users can tap on the result button to view the selected patient profile.

4.4.13 Patient Profile Page

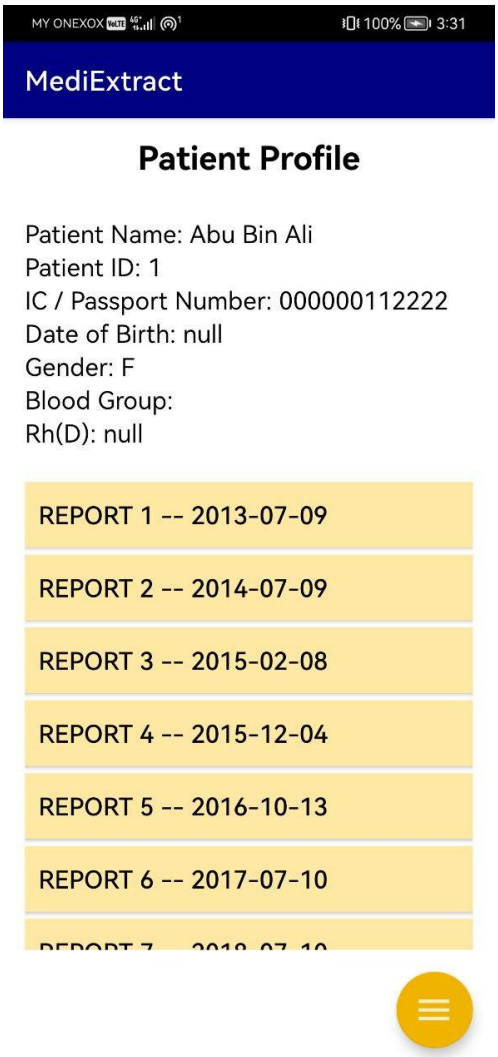


Figure 4.51: Patient Profile Page of the Blood Test Report Extraction Application

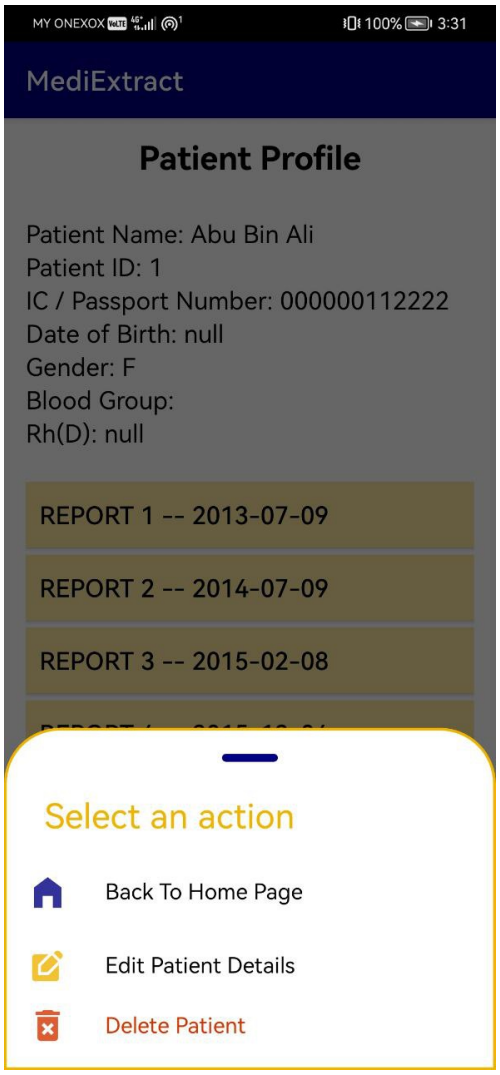


Figure 4.52: Bottom Dialog of Patient Profile Page

The user will be navigated to the Patient Profile Page after tapping on the patient button from Search Patient Page. The patient details will be displayed at the top section and followed by created report button showing in the form of light orange colour button as shown as Figure 4.51. The user will be navigated to the Report Page to view selected report after tapping on the created report button. Besides, the user can choose to edit or delete the patient profile by tapping on the respective buttons from the bottom dialog showing in

Figure 4.52. Once confirmed delete the patient profile, the related reports created corresponding to this patient will be deleted together.

4.4.14 User Profile Page

Figure 4.53: User Profile Page of the Blood Test Report Extraction Application

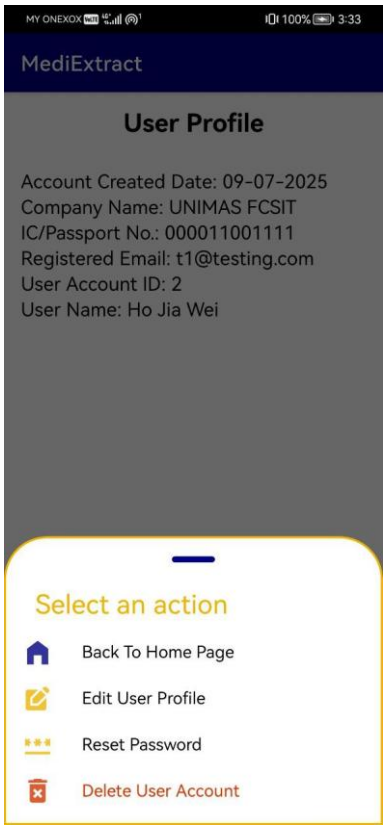


Figure 4.54: Bottom Dialog of User Profile Page

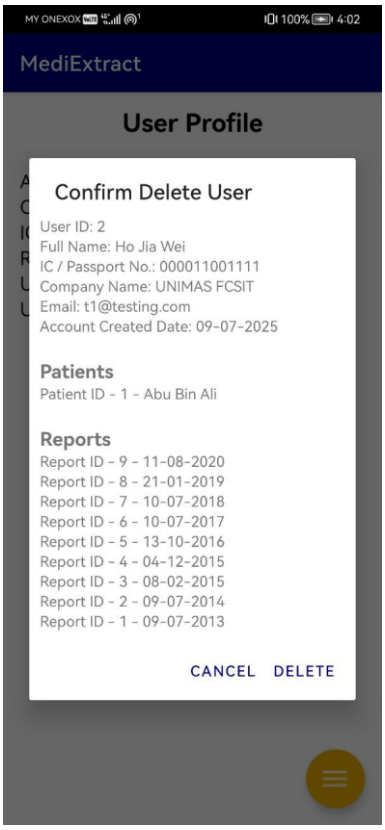


Figure 4.55: Confirm Delete User Popup

The users will be navigated to the User Profile Page after tapping on the 'Account' button from User Main Page side bar menu. The users can view their registered information in this page as shown as Figure 4.53. They can also choose to delete their current account by tapping on the red 'Delete User Account' button from the bottom dialog showing in

Figure 4.54. The confirm delete user popup will be displayed with the user details, the linked reports and patients created under this user ID as shown as Figure 4.55.

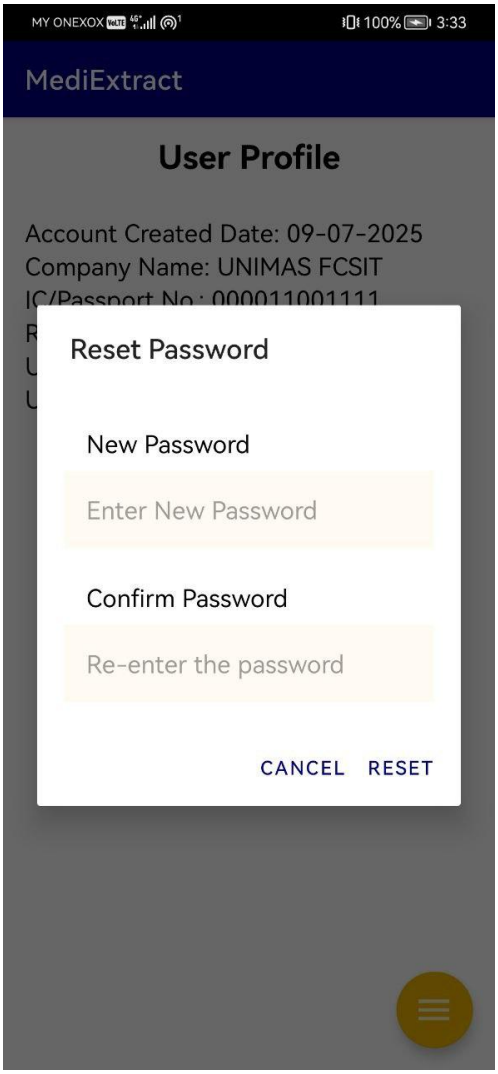


Figure 4.56: Reset Password Popup of Patient Profile Page

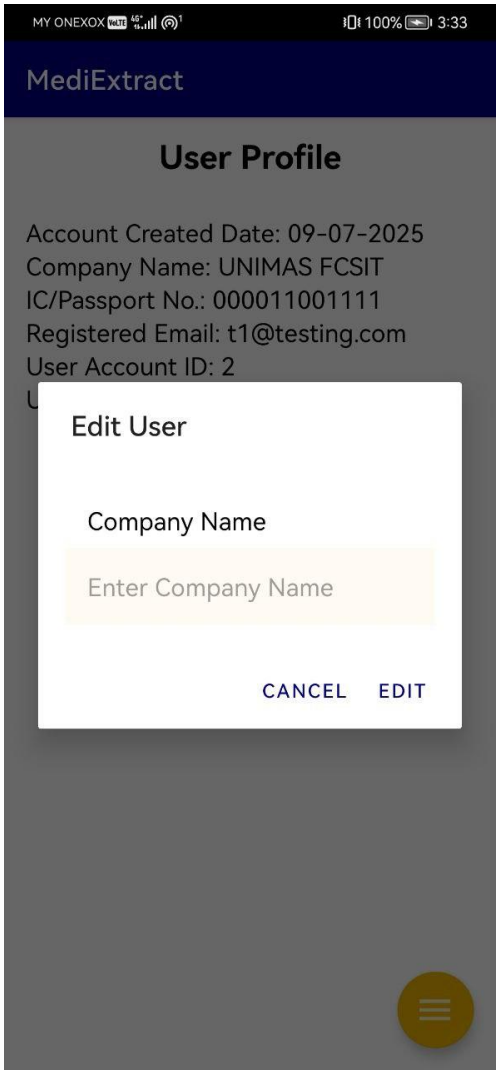


Figure 4.57: Edit User Popup of Patient Profile Page

The users can also choose to reset their account password by tapping on the 'Reset Password' button from the bottom dialog. The reset password popup will be displayed with for the users to type in new password and re-enter the new password for confirmation as shown as Figure 4.56. Figure 4.57 shows the edit user popup which only allows the user to

edit their company name as other personal information such as full name and IC/Passport number should be remain unchanged for account verification purposes.

4.4.15 Visualization Page

Figure 4.58: Visualization Page of the Blood Test Report Extraction Application

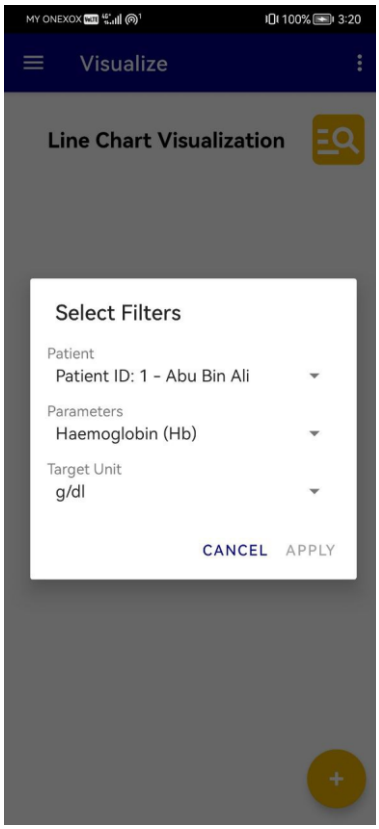


Figure 4.59: Parameter and Month Selection Popup

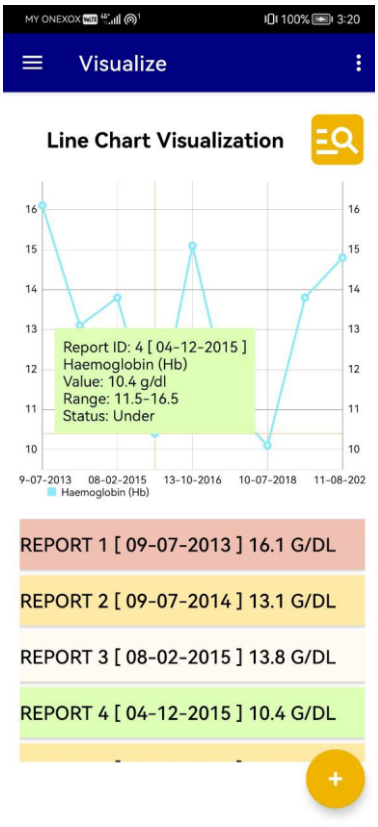


Figure 4.60: Tooltip of the Line Chart Visualization

The users will be navigated to the Visualization Page after tapping on the 'Visualize' button from User Main Page. The users are required to select a patient. a blood test parameter and the target unit to visualize the extracted reports as shown as Figure 4.59. After tapping on the 'Apply' button, the line chart of parameter value of all reports under the selected patient will be displayed at the top half section of the page as shown as Figure

4.58. The report button which displays the value and unit of each report will be shown in 4 colours base on its parameter status: red colour indicates ‘Over’, green colour indicates ‘Under’, light orange colour indicates ‘NA’, and beige colour indicates ‘Normal’. The users can tap on the report button to view the full extracted report. When the user taps on the line chart marker, the information of the selected report such as report ID, report date, selected blood test parameter value, unit, normal range, and status will be displayed in the form of tooltip on the line chart. Figure 4.60 shows the coloured tooltip which will be displayed when the user taps on the line chart marker.

4.4.16 Request Usage Limit Interface

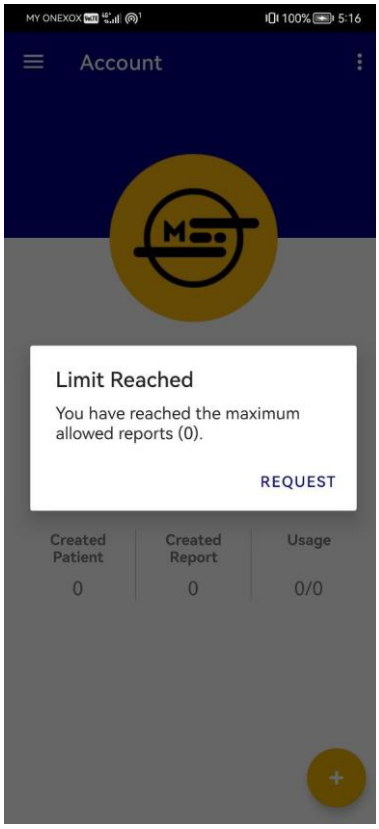


Figure 4.61: Usage Limit Reached Popup

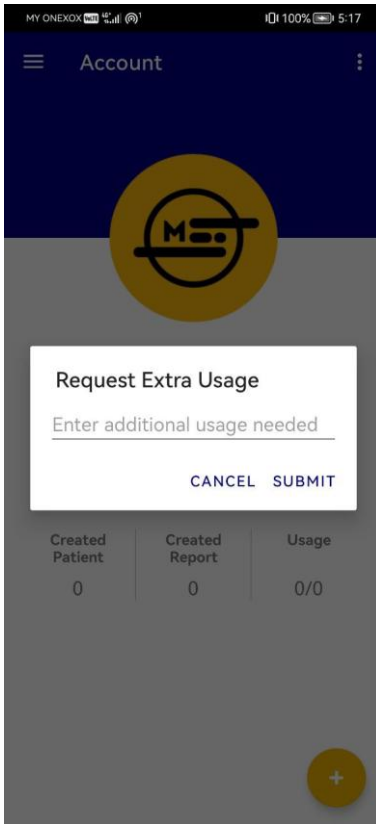


Figure 4.62: Request Extra Limit Popup

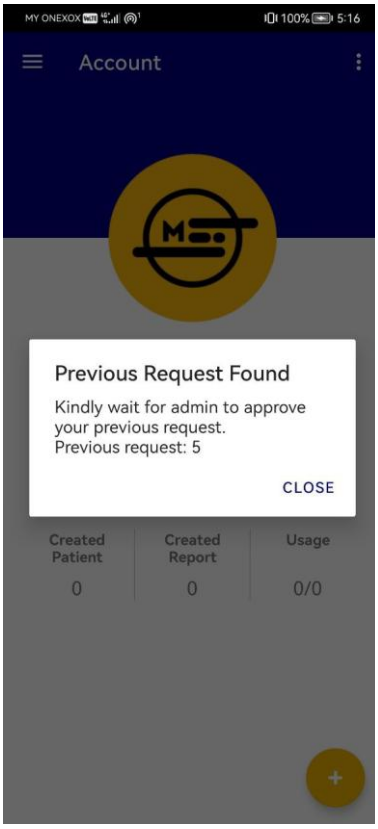


Figure 4.63: Callback Message Popup

Figure 4.61 shows the usage limit popup when the user had reached the maximum amount of extracted report. The users can choose to tap on the ‘Request’ button to request for additional usage limit as shown in Figure 4.62. The submitted request will be shown at the admin user verification page for further approval. If the user had submitted a request but have not been approved by the admin, a callback message popup will be displayed to notify the user as shown in Figure 4.63.

4.4 Summary

This chapter delves into the implementation phase of Automatic Data Extraction from Blood Test Report application, ensuring that all system components align with project requirements is paramount. Python as the programming language to implement the code for Flask API. The user interface was designed and implemented in Android Studio based on Material Design guidelines including consistent layout and expressive colour to increase user accessibility (Material Design - Version 2, n.d.). Microsoft Azure AI Document Intelligence Studio used for data model training and Azure SQL Database used for data storage.

Chapter 5: Testing

5.1 Introduction

This chapter focused on the testing of the Blood Test Report Data Extraction Application. The following discussion will be sectioned into two categories: functionality testing consists of User Acceptance Testing and Unit Testing and performance evaluation. In order for the application to be fully utilised, testing is important to ensure that the requirement has been met. Other than that, the testing result should fulfil all the objectives discussed in Chapter 1.

5.2 User Acceptance Testing

User acceptance testing (UAT) was conducted for the user to test the proposed application and evaluate their overall satisfaction based on their user experience. A Google Form with 10 questions design based on System Usability Score had been distributed to the 30 respondents. Each question will have 5 scores from 1 for Strongly Disagree to 5 for Strongly Agree. The respondent will be asked to explore the application for both user and admin roles and fill up this Google Form via link: https://forms.gle/TjgMSTfsxcRPFZjUA

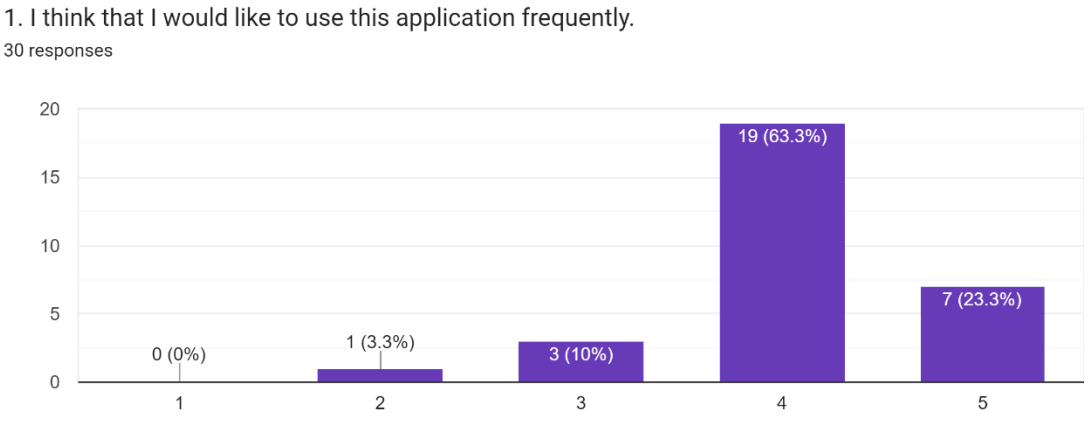


Figure 5.1: User Acceptance Testing Question 1

Figure 5.1 shows the users' willingness of using this application frequently if they are doctor or nurse who will need to deal with blood test report daily. Majority of the respondents (26 out of 30 respondents) are willing to use this application daily.

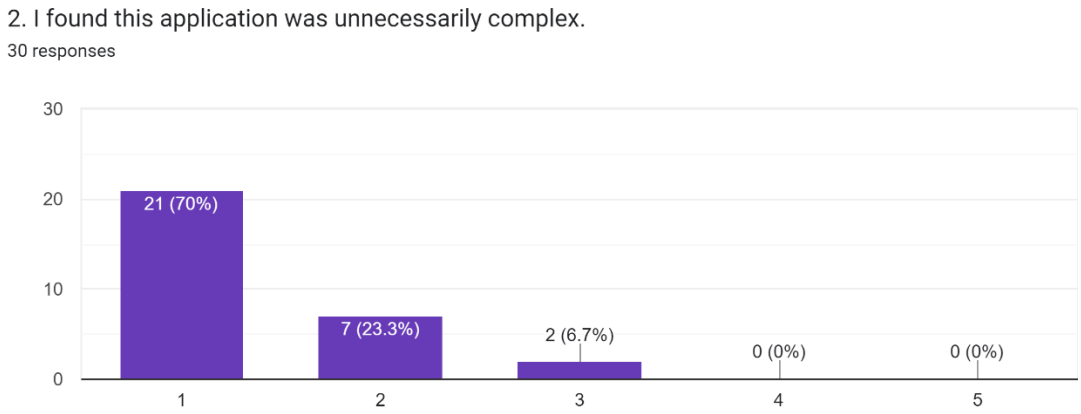


Figure 5.2: User Acceptance Testing Question 2

Figure 5.2 shows the user responses regarding the unnecessarily complexity of the application design. Majority of the respondents (28 out of 30 respondents) feel that the application had agreed that the application was not unnecessarily complex. The remaining 2 respondents stay neutral toward this question.

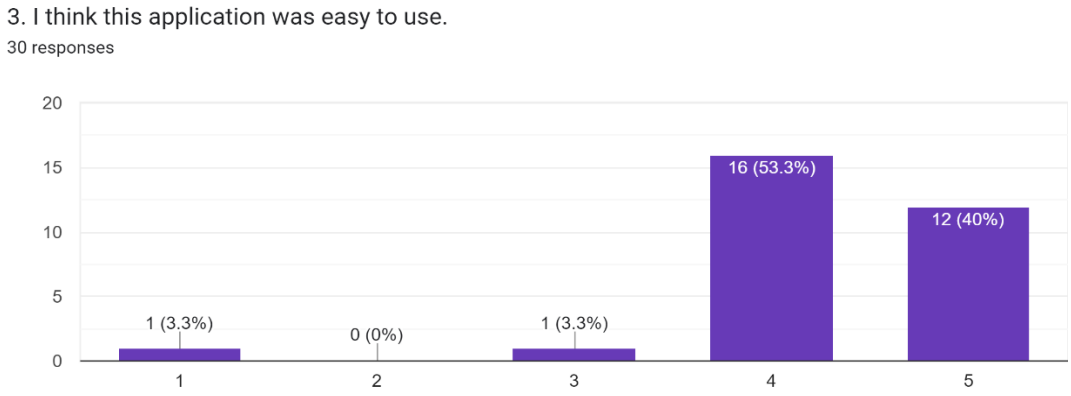


Figure 5.3: User Acceptance Testing Question 3

Figure 5.3 shows the user responses towards the ease of use of the application. 12 out of 30 respondents had strongly agreed that the application was easy to use. Another 16 respondents feel that the application was easy to use.

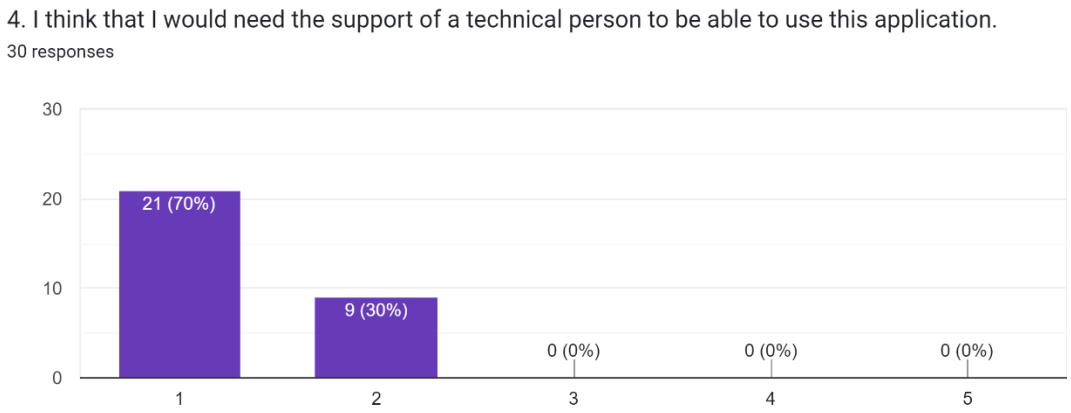


Figure 5.4: User Acceptance Testing Question 4

Figure 5.4 shows the user responses towards the need of technical person guidance to use the application. Majority of the respondents (21 out of 30 respondents) had strongly agreed that they can use this application without any guidance from the technical person.

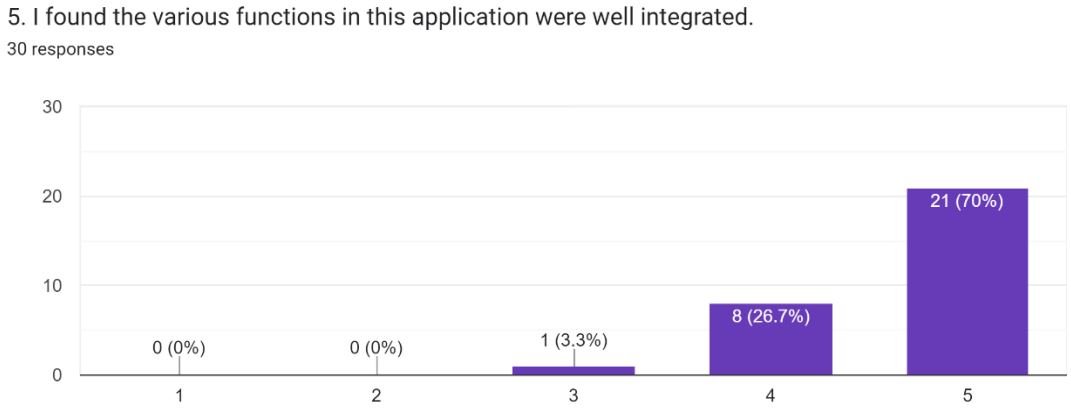


Figure 5.5: User Acceptance Testing Question 5

Figure 5.5 shows the user responses towards the function integration of the application. Majority of the respondents (29 out of 30 respondents) had agreed that various functions of this application were well integrated. Another 1 respondent remain neutral towards this question.

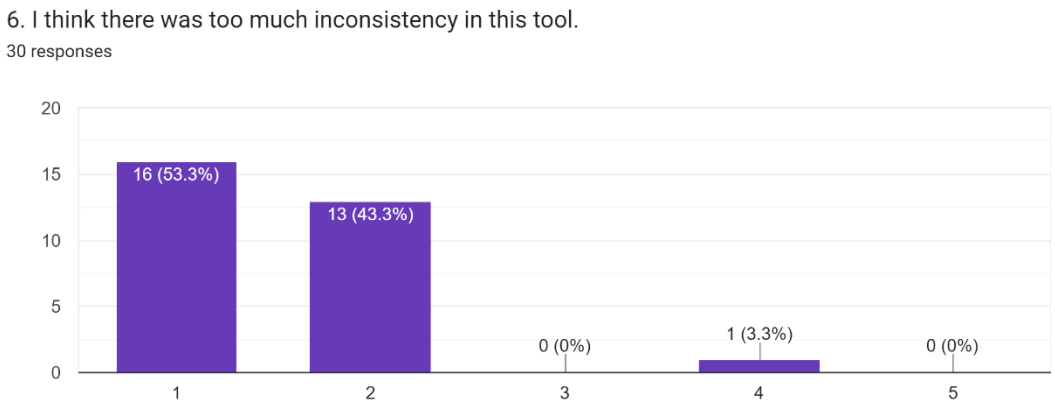


Figure 5.6: User Acceptance Testing Question 6

Figure 5.6 shows the user responses towards the inconsistency of the application. Majority of the respondents (16 out of 30 respondents) had strongly agreed that this application was well design without much inconsistency. Another 1 respondent thinks that there was too much inconsistency in this application while using.

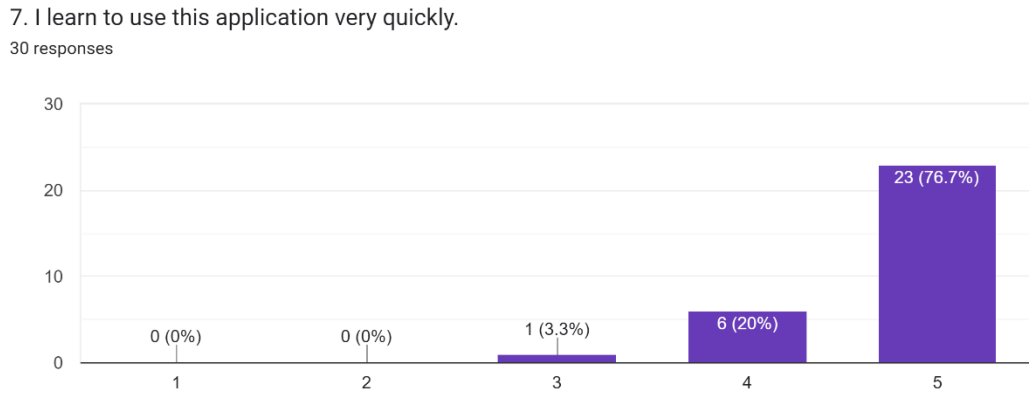


Figure 5.7: User Acceptance Testing Question 7

Figure 5.7 shows the user responses towards the ease of learning of this application. Majority of the respondents (23 out of 30 respondents) had strongly agreed that they can learn to use this application in a very short time. Another 1 respondent remain neutral towards this question.

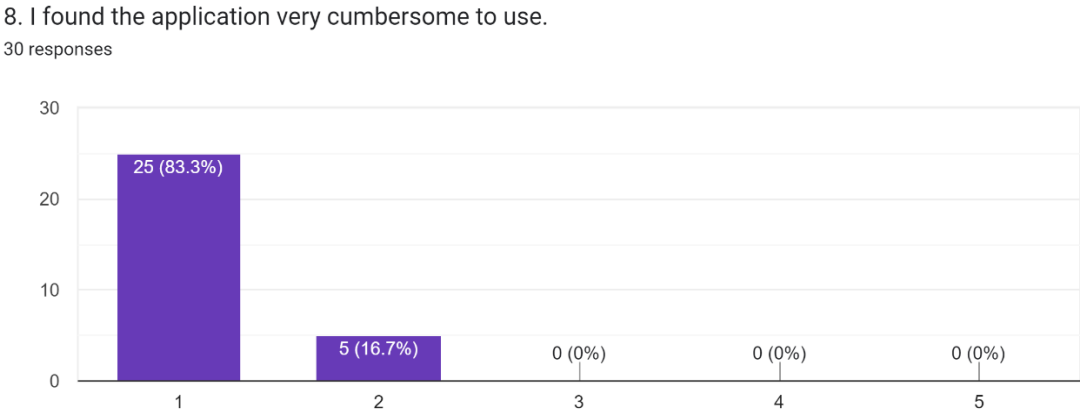


Figure 5.8: User Acceptance Testing Question 8

Figure 5.8 shows the user responses towards the negative user experience while using this application. Majority of the respondents (25 out of 30 respondents) had strongly agreed that they do not think this application was crumbersome to use. Another 5 respondents also agreed based on the previous point of view as they did not feel frustrated while using the application.

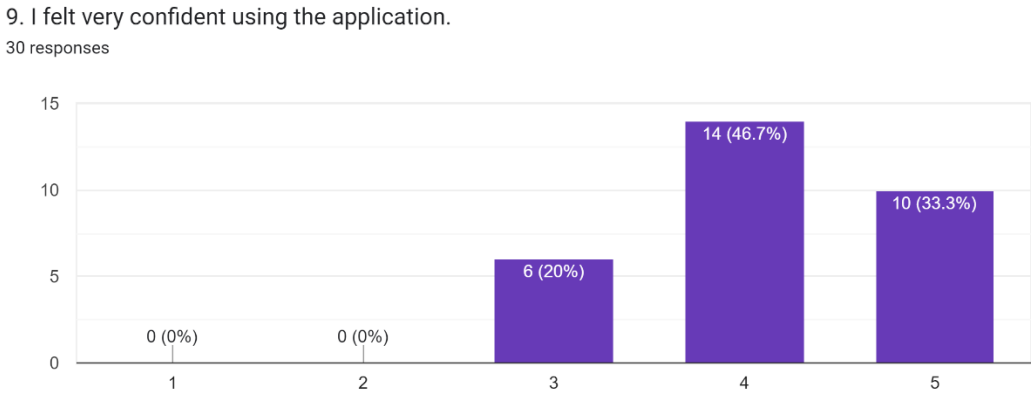


Figure 5.9: User Acceptance Testing Question 9

Figure 5.9 shows the users' confidence level while using this application. Majority of the respondents (24 out of 30 respondents) had agreed that they are confident while using this application. Another 6 respondents remain neutral towards this question.

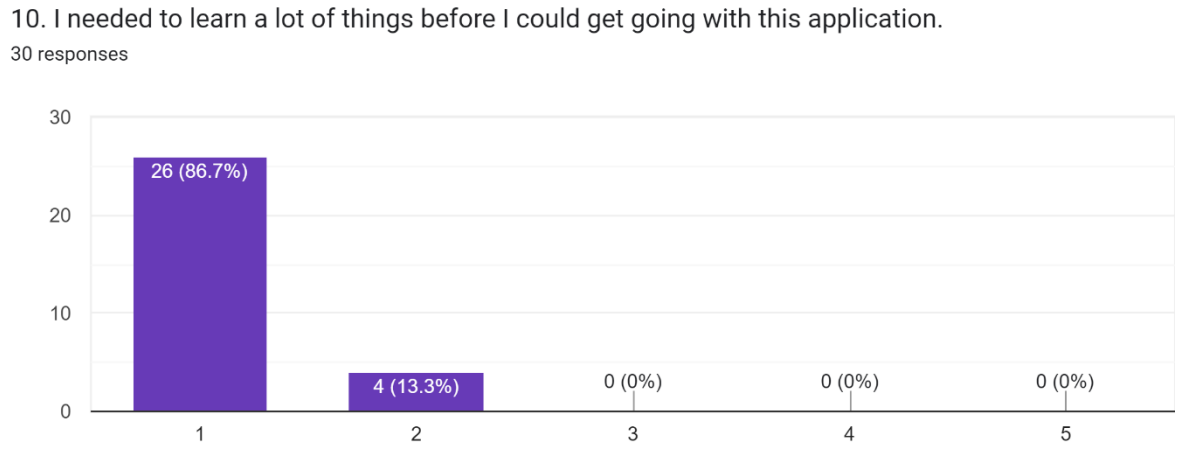


Figure 5.10: User Acceptance Testing Question 10

Figure 5.10 shows user responses towards the need of preparation for the users before use this application. Majority of the respondents (26 out of 30 respondents) had agreed that they do not need to learn anything before using this application. This also reflected the ease of use of the proposed application as the user can straight away use the application without additional learning.

5.3 Unit Testing

The unit testing was carried out by attempting to input erroneous and valid information into each of the designed test cases. The following discussion will be categorised by test case and results are tabulated with the test scenario, test steps, expected result, actual result, and test status.

5.3.1 User Account Sign Up

This section covers the functionality testing for user account sign up.

Table 5.1: Test Case for User Account Sign Up

Test Case Name: User Account Sign Up					
Test Case Description:		To test the user sign-up function of Sign-Up Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC1_1	Sign up user account with invalid information.	1. Tap on 'Sign Up' button. 2. Provide invalid user information. 3. Tap on 'Sign Up' button.	Show error popup and sign-up process is unsuccessful.	As expected	Pass
TC1_2	Sign up user account with valid information.	1. Tap on 'Sign Up' button. 2. Provide valid user information. 3. Tap on 'Sign Up' button.	Show successful sign-up popup.	As expected	Pass

5.2.2 Login User Account & Login Admin Account

Table 5.2: Test Case for User and Admin Account Login

Test Case Name: User and Admin Account Login					
Test Case Description:		To test the user login function of Login Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**

TC2_1	Login user account with wrong information.	1. Input invalid user information. 2. Tap on 'Login' button.	Show error popup and login process is unsuccessful.	As expected	Pass
TC2_2	Login unverified user account with correct registered information.	1. Input registered information. 2. Tap on 'Login' button.	Show not verified popup and login process is unsuccessful.	As expected	Pass
TC2_3	Login verified user account with correct registered information.	1. Input registered information. 2. Tap on 'Login' button.	User logged in successfully.	As expected	Pass
TC2_4	Login admin account with registered information.	1. Input registered admin information. 2. Tap on 'Login' button.	Admin logged in successfully.	As expected	Pass

5.2.3 Extract Data from Scanned Image

Table 5.3: Test Case for Scan Blood Test Report

Test Case Name: Scan Blood Test Report					
Test Case Description:		To test the scanning function and browse file function of Camera Page are working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC3_1	Scan and capture image of blood test report.	1. Tap on 'Scan Report' button. 2. Capture blood test report using device camera function. 3. Tap on next button.	Navigated to Verification Page with extracted data.	As expected	Pass
TC3_2	Browse blood test report in PDF file type.	1. Tap on the blue 'Upload Report' button. 2. Select a blood test report in PDF form. 3. Tap on 'Confirm' button.	Navigated to Verification Page with extracted data.	As expected	Pass
TC3_3	Error handling for invalid document input.	1. Tap on 'Scan Report' button. 2. Capture a book object using device camera function. 3. Tap on next button.	Invalid document popup appeared and asked user to recapture the image or proceed to verify extracted document.	As expected	Pass

5.2.4 Validate and Save Extracted Data

Table 5.4: Test Case for Validate and Save Verified Data

Test Case Name: Validate and Save Verified Data					
Test Case Description:		To test the edit text function, convert unit button and save function of Verification Page are working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC4_1	Verify the extracted data by correcting the showing value.	1. Edit the extracted data shown in the text input field. 2. Tap on each page buttons. 3. Tap on the 'Next' button. 4. Select 'Save' button of the confirm saving popup.	Navigated to the Report Page.	As expected	Pass
TC4_2	Save the verified data to the database.	1. Verify all 4 pages of the extracted data. 2. Tap on 'Next' button.	Successfully saved popup appeared.	As expected	Pass
TC4_3	Error handling for empty required field.	1. Clear the input of patient name shown in Page 1. 2. Verify remaining pages. 3. Tap on 'Next' button.	Invalid patient name popup appears and the data will not be saved to the database.	As expected	Pass

5.2.5 Search Patient by Patient Name and Patient ID

Table 5.5: Test Case for Search Patient by Patient Name and Patient ID

Test Case Name: Search Patient by Patient Name and Patient ID					
Test Case Description:		To test the search patient function of Search Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC5_1	Search patient by invalid patient name.	1. Insert invalid patient name. 2. Tap on the magnifying glass icon button.	Text of No patient record found under "input" will be shown.	As expected	Pass
TC5_2	Search patient by existing patient id.	1. Insert existing Patient ID. 2. Tap on the magnifying glass icon button.	Patient button with matching Patient ID will be shown.	As expected	Pass

5.2.6 View Extracted Blood Test Report

Table 5.6: Test Case for View Extracted Blood Test Report

Test Case Name: View Extracted Blood Test Report					
Test Case Description:		To test the data retrieving functions of Report Page are working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC6_1	View extracted blood test report	1. Navigate to Search Page. 2. Tap on the light orange patient button.	Navigated to the Report Page with report details showing.	As expected	Pass
TC6_2	View saved original scanned blood test report.	1. Tap on the blue button with image icon.	Preview report popup displayed with the correct saved image.	As expected	Pass

5.2.7 Manage Saved Blood Test Report

Table 5.7: Test Case for Manage Saved Blood Test Report

Test Case Name: Manage Saved Blood Test Report					
Test Case Description:		To test the edit report function and delete report function of Report Page are working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC7_1	View and edit report from Report Page.	1. Tap on 'Edit Report' button. 2. Edit the field. 3. Save edited report.	Navigated to Patient Profile Page with the report updated successfully.	As expected	Pass
TC7_2	Delete report from Report Page.	1. Tap on 'Delete Report' button. 2. Confirm delete.	Navigated to Patient Profile Page with the report deleted successfully.	As expected	Pass

5.2.8 Generate Line Chart Visualization

Table 5.8: Test Case for View Generated Line Chart Visualization

Test Case Name: View Generated Chart Visualization					
Test Case Description:		To test the draw chart function and display report function of Visualization Page are working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**

TC8_1	Generate line chart visualization.	1. Select a patient. 2. Select a blood test parameter. 3. Select a target unit. 4. Tap on 'Apply' button.	Line chart with corresponding selection was shown.	As expected	Pass
TC8_2	View selected report from Visualization Page.	1. Tap on one of the markers. 2. Tap on the report button.	Navigated to the Report Page with selected report.	As expected	Pass
TC8_3	Line chart interaction.	1. Generate a line chart. 2. Tap on one of the markers.	Tooltip with corresponding selection was shown.	As expected	Pass
TC8_4	Exception unit handling.	1. Select a patient. 2. Select a blood test parameter from Differential count. 3. Select a target unit. 2. Tap on 'Apply' button.	A note had shown underneath the line chart to notify the user about unit exception.	As expected	Pass

5.2.9 Export Extracted Blood Test Report

Table 5.9: Test Case for Export Extracted Blood Test Report

Test Case Name: Export Extracted Blood Test Report					
Test Case Description:		To test the export report function of Report Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC9_1	Export extracted blood test report.	1. Select a report from Patient Profile Page. 2. Tap on 'Export PDF' button. 3. Insert filename. 4. Tap on 'Export' button.	Navigated back to the Report Page and the PDF file with correct filename had successfully exported.	As expected	Pass
TC9_2	Error handling for empty filename.	1. Select a report from Patient Profile Page. 2. Tap on 'Export PDF' button. 3.Tap on 'Export' button without filename.	Error popup with text 'Filename cannot be empty' had appeared.	As expected	Pass

5.2.10 Admin Verify User Account

Table 5.10: Test Case for Verify User Account

Test Case Name: Verify User Account					
Test Case Description:		To test the verify user function of Verify User Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC10_1	View unverified or limit requested user account.	1. Select 'Verify User' from Admin Main Page sidebar menu. 2. Tap on 'Request' tab.	Unverified user button showing in blue colour while user account with usage limit showing in light orange colour button.	As expected	Pass
TC10_2	View all registered user account.	1. Select 'Verify User' from Admin Main Page sidebar menu. 2. Tap on 'All' tab.	All user was shown in designed colours accordingly.	As expected	Pass
TC10_3	Verify user account.	1. Select a blue user button from 'Request' tab of Verify User Page. 2. Tap on 'Verify' radio button.	Selected user was verified and button colour changed to beige colour showing in 'All' tab.	As expected	Pass

5.2.11 Admin Delete User Account

Table 5.11: Admin Delete User Account

Test Case Name: Admin Increase User Usage Limit					
Test Case Description:		To test the delete user function of Verify User Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC11_1	Delete user account from admin side.	1. Select a user from Verify User Page. 2. Tap on 'Delete User' button. 3. Insert filename. 4. Tap on 'Export' button.	Verify User Popup was closed and admin navigated back to the Verify User Page and the user had successfully deleted.	As expected	Pass

5.2.12 Admin Increase User Usage Limit

Table 5.12: Admin Increase User Usage Limit

Test Case Name: Admin Increase User Usage Limit					
Test Case Description:		To test the increase usage limit function of Verify User Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC12_1	Increase user usage limit from admin side.	1. Select 'Verify User' from Admin Main Page sidebar menu. 2. Select a light orange user button from 'Request' tab. 3. Tap on 'Usage Limit' button. 4. Insert a valid number into the increase usage limit popup. 5. Tap on 'Increase' button.	Verify User Popup was closed and admin navigated back to the Verify User Page and the user usage limit had been increased based on the number inserted.	As expected	Pass
TC12_2	Error handling for empty usage limit input.	1. Select a light orange user button from 'Request' tab. 3. Tap on 'Usage Limit' button. 4. Tap on 'Increase' button without any input.	Error popup with text 'Usage limit cannot be empty' had appeared.	As expected	Pass

5.2.13 Admin View Report Evaluation

Table 5.13: Admin View Report Evaluation

Test Case Name: Admin View Report Evaluation					
Test Case Description:		To test the view report function of View Report Evaluation Page is working.			
Test Case ID	**Test Scenario**	**Test Steps**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
TC13_1	View report from admin side.	1. Select 'View Report Evaluation' from Admin Main Page sidebar menu. 2. Select a report.	Navigated to another page showing all extracted fields with CER and confidence score of the selected report in table form.	As expected	Pass
TC13_2	Search report by valid report ID.	1. Select 'View Report Evaluation' from Admin Main Page sidebar menu. 2. Insert valid report ID to the search field.	Report with matching inserted report ID appeared.	As expected	Pass
TC13_3	Search report by invalid report ID.	1. Select 'View Report Evaluation' from Admin Main Page sidebar menu. 2. Insert invalid report ID to the search field.	Text of No report found under "input" will be shown.	As expected	Pass

5.3 Performance Evaluation

In this section, the implementation of the Blood Test Report Extraction Application will be discussed and analysed. Two evaluation methods will be selected for the application performance evaluation. Character Error Rate is a metric used to evaluate the performance of text recognition systems especially in Optical Character Recognition (OCR) (Graves et al., 2006). It was selected instead of Word Error Rate as the vocabulary is short for blood test parameter value, unit and reference range. Confidence Score calculated by Azure AI Document Intelligence will also use for making data-driven decisions based on the suggested reliability thresholds (Microsoft, 2025).

A total of 30 participants will be involved in the application testing to evaluate the performance of the blood test report extraction application. 10 blood test reports were prepared and categorized into 3 groups matching with different scenarios. The first category was the New Report Layout Category consist of 4 blood test reports in new report layout from Innovative Diagnostic, Klinik Kesihatan Bukit Minyak (KKBK), Lablink (M) Sdn Bhd, Gnosis Laboratories (M) Sdn Bhd. For Existing Report Layout Category, it consists of 3 blood test reports with the same report layout that used for model training from Borneo Medical Centere (BMC), University of Malaya Medical Center (PPUM), and Gribbles Pathology (Malaysia) Sdn Bhd. Lastly, for the Modified Report Layout Category, it consist of 3 blood test reports with modified report layout from Quantum Diagnostic Sdn Bhd, Borneo Medical Centere (BMC), University of Malaya Medical Center (PPUM). These reports were used in model training but had been modified by repositioning the content in a different way compared to the original report layout.

Each participant will be extracting blood test reports from 1 category and the average per report, average per category, and total average of testing were calculated for both CER and

confidence score based on the formula below. The testing results will be tabulated in Table 5.10 for CER and Table 5.11 for confidence score.

$$\text{Average per Report} = \frac{\text{Sum of each CER/confidence score per extracted field}}{\text{Total number of extracted field}}$$

$$\text{Average per Category} = \frac{\text{Sum of all average per report}}{\text{Total number of report with same category}}$$

$$\text{Total Average of Testing} = \frac{\text{Sum of all average per category}}{\text{Total number of category}}$$

5.3.1 Character Error Rate (CER)

Accuracy can be measured by calculating the character error rate, which reflects the difference between the measured value and the true value. CER is calculated based on the Levenshtein distance, which is the minimum number of insertions (I), deletions (D), and substitutions (S) and total number of characters in the string (N).

$$CER = \frac{(I + D + S)}{N}$$

Table 5.10 shows the character error rate result from the application testing. The result was then visualized in a bar chart showing in Figure 5.11.

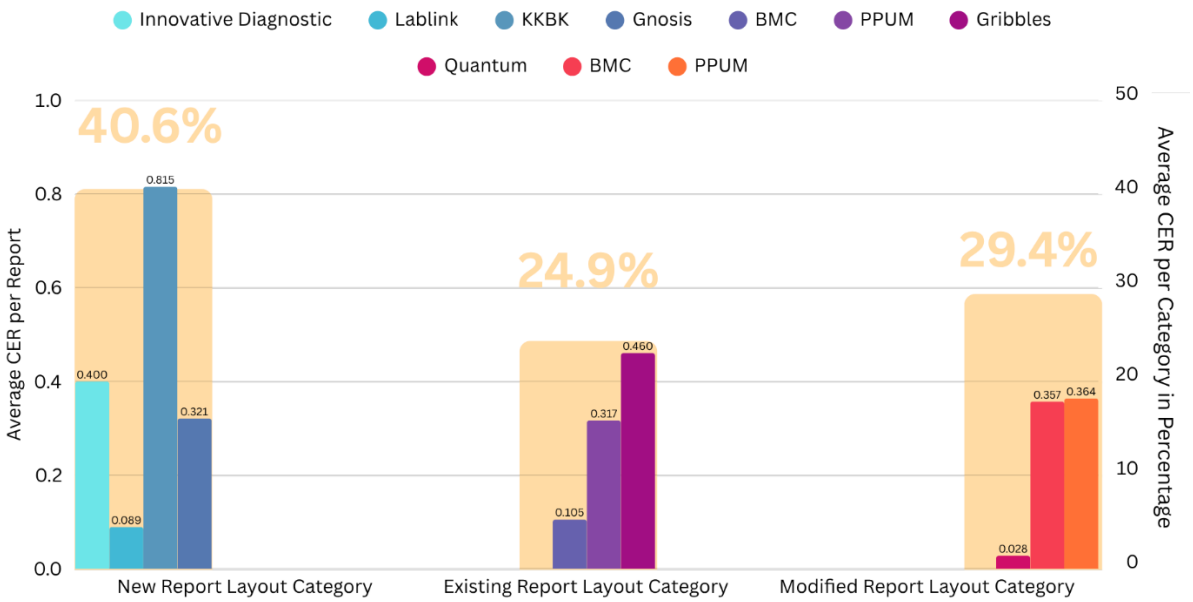


Figure 5.11: Visualization of Testing Result Analysis for CER

Based on the results, the custom extraction model used in the proposed blood test report extraction application performed best for the existing report layout category showing with the lowest average CER per category (24.9%) compared to the second well performed category which is the modified report layout category (29.4%) and the category with highest average CER per category which is the new report layout category (40.6%). As the key field data was not trained during the model training, the model will have higher error rate when analysing blood test report with new layout. Most of the time the OCR performed well during document analysis, but it failed to perform well for key pair matching. This might lead to the high CER for the new report layout data extraction. For the modified report layout category, it had lower average error rate compared to the new report layout category, but higher error rate compared to the existing report layout category as the model will has higher chance to mismatch the key value pairs, for example pairing with values belongs to another key.

Table 5.10: Application Performance Evaluation using Character Error Rate

	Character Error Rate (CER)										
Report / Tester	**New Report Layout Category**				**Existing Report Layout Category**			**Modified Report Layout Category**			**Average**
	Innovative Diagnostic	Lablink	KKBK	Gnosis	BMC	PPUM	Gribbles	Quantum	BMC	PPUM	
T1	0.379	0.000	0.800	0.065							**0.406**
T2	0.401	0.080	0.820	0.043							
T3	0.452	0.207	0.800	0.239							
T4	0.280	0.000	0.800	0.169							
T5	0.495	0.478	0.867	0.807							
T6	0.379	0.020	0.800	0.391							
T7	0.406	0.000	0.867	0.399							
T8	0.418	0.020	0.800	0.318							
T9	0.391	0.040	0.800	0.397							
T10	0.401	0.040	0.800	0.384							
T11					0.007	0.294	0.255				**0.294**
T12					0.005	0.049	0.042				
T13					0.160	0.373	0.614				
T14					0.103	0.056	0.400				
T15					0.039	0.373	0.188				
T16					0.150	0.446	0.614				
T17					0.147	0.339	0.615				
T18					0.147	0.440	0.661				
T19					0.127	0.339	0.629				
T20					0.167	0.456	0.583				
T21								0.020	0.344	0.368	**0.249**
T22								0.020	0.377	0.317	
T23								0.060	0.326	0.385	
T24								0.039	0.295	0.368	
T25								0.020	0.393	0.402	
T26								0.020	0.344	0.385	
T27								0.020	0.344	0.368	
T28								0.039	0.393	0.409	
T29								0.020	0.410	0.317	
T30								0.020	0.344	0.317	
Average	**0.400**	**0.089**	**0.815**	**0.321**	**0.1052**	**0.3165**	**0.4601**	**0.0278**	**0.357**	**0.3636**	**0.316**

5.3.2 Confidence Score by Azure AI Document Intelligence

Azure AI Document Intelligence's confidence score is the estimated probabilities (0 to 1) tied to each extraction element (Microsoft, 2025).

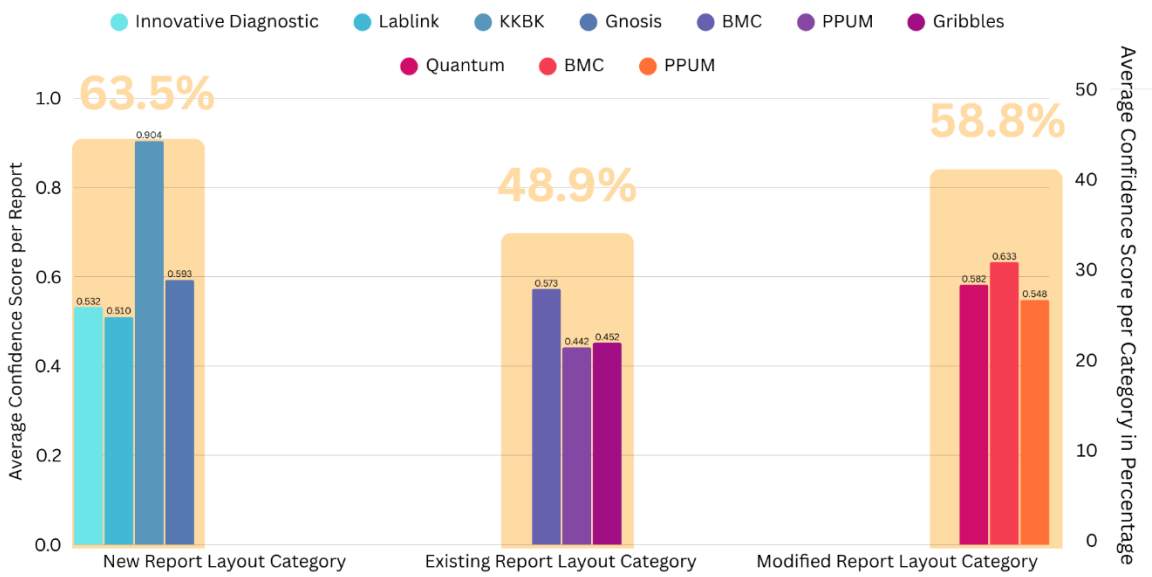


Figure 5.12: Visualization of Testing Result Analysis for Confidence Score

Based on the results showing in Figure 5.12, the custom extraction model used in the proposed blood test report extraction application has the highest average confidence score for the new report layout category (63.5%) compared to the modified report layout category (58.8%) and the existing report layout category showing with the lowest average confidence score per category (48.9%). Based on the documentation of Azure AI Document Intelligence shared by Laujan (n.d.), new report layout category fall under the medium confidence tier which shows a fairly good key pair values matching that should be mostly correct. This analysis was contrast with the CER testing result which indicating the high possibility of no value was matched with the keys during the extraction process. This works the same way to the lowest confidence score with lowest CER of existing report layout category.

Table 5.11 Application Performance Evaluation using Confidence Score

	Confidence Score Calculated by Microsoft Azure AI Document Intelligence										
Report / **Tester**	**New Report Layout Category**				**Existing Report Layout Category**			**Modified Report Layout Category**			**Average**
	Innovative Diagnostic	Lablink	KKBK	Gnosis	BMC	PPUM	Gribbles	Quantum	BMC	PPUM	
T1	0.520	0.518	0.887	0.584							**0.635**
T2	0.530	0.531	0.888	0.599							
T3	0.552	0.456	0.918	0.637							
T4	0.586	0.527	0.888	0.624							
T5	0.507	0.502	0.909	0.579							
T6	0.520	0.518	0.888	0.584							
T7	0.555	0.518	0.908	0.580							
T8	0.514	0.509	0.931	0.584							
T9	0.519	0.505	0.926	0.570							
T10	0.520	0.516	0.900	0.584							
T11					0.319	0.384	0.26073				**0.489**
T12					0.603	0.492	0.502				
T13					0.600	0.438	0.464				
T14					0.587	0.427	0.460				
T15					0.551	0.466	0.493				
T16					0.621	0.430	0.446				
T17					0.577	0.432	0.470				
T18					0.622	0.440	0.491				
T19					0.624	0.456	0.461				
T20					0.629	0.456	0.477				
T21								0.548	0.614	0.526	**0.588**
T22								0.561	0.622	0.537	
T23								0.573	0.657	0.539	
T24								0.621	0.682	0.531	
T25								0.582	0.598	0.588	
T26								0.579	0.618	0.534	
T27								0.556	0.634	0.529	
T28								0.613	0.647	0.531	
T29								0.602	0.663	0.613	
T30								0.584	0.592	0.554	
Average	**0.532**	**0.510**	**0.904**	**0.593**	**0.573**	**0.442**	**0.452**	**0.582**	**0.633**	**0.548**	**0.571**

5.4 Summary

In summary, this chapter covered testing and evaluation of the including user acceptance testing, unit testing, and performance evaluation. As observed from user acceptance testing, the responses towards the usability of the proposed application had proved that the application is ready for deployment with minor improvement. For unit testing, all actual result from the testing is parallel to the expected results. In shorts, the test status for the functionalities and test case of the proposed system has passed, leading to meeting expectation of the development and objectives of the proposed application. As for performance evaluation, Character Error Rate and confidence score were used as the evaluation metrics for the application. Character Error Rate was used to evaluate the error rate of extracted data over the correct data while the confidence score which indicates the probability by measuring the degree of statistical certainty that the extracted result is detected correctly (Laujan, n.d.). The estimated accuracy is calculated by running a few different combinations of the training data to predict the labelled values.

After the testing, it is found that there are several flaws needed to be improved. These flaws will be covered in Chapter 6 of this report.

Chapter 6: Conclusion and Future Work

6.1 Introduction

This chapter will conclude the project's achievements in accordance with the project's objectives, which can be found in following section. Other than that, the limitations of the project will be analysed and discussed along with the future works needed to overcome the stated limitations.

6.2 Achievements

The proposed system has achieved the aim and objectives as shown in Table 6.1.

Table 6.1: Project Objectives and Achievements

Objective	**Achievement**
To study current document processing techniques.	User requirements for the data extraction from blood test report are gathered and studied during requirement planning stage. Solution of using document processing techniques was proposed based on the requirements.
To implement document processing techniques on the application development of automatic data extraction from blood test report.	Using phases of rapid application development methodology, the proposed application was designed in Chapter 3 and implemented in Chapter 4. The proposed application was implemented based on the proposed solution such as using Microsoft Azure AI Document Intelligence to extract data from the blood test report.
To evaluate the application performance using Character Error Rate and Confidence Value calculated by Microsoft Azure Document Intelligence.	Functionality testing and performance evaluation have been done on the proposed application and discussed in Chapter 5. This chapter also explained the test cases for functionality testing and performance evaluation using Character Error Rate and Confidence Value.

6.3 Limitation

There is room for improvement in the proposed application, even though the objectives have been fulfilled. Based on the discussion from chapter 5, several limitations had been discovered during the functionality testing. These limitations are listed below.

- Limited unit conversion in Visualization Page.
- Long waiting time for calling Azure AI Document Intelligence API.
- Low accuracy for data extraction from blood test report.
- Only specified Haematology test metrics can be extracted.

Due to lack of reference to support unit conversion, user only given 2 options to choose to generate the line chart visualization. Table 6.2 shows the specified units that have unit conversion with supporting reference (Online Conversion Calculator for Many Types of Measurement Units in Laboratory and Medicine Practice | UnitsLab.com, n.d.). The user can only choose from the given target unit options to generate the line chart by converting the parameter unit based on Table 6.2. The units that are not in Table 6.2 will be displayed on the line chart and lead to misinterpretation as the line chart will be scaled off when there are multiple values with different units.

Table 6.2: Unit Conversion for Blood Test Parameter

Blood Test Parameter	**Unit**	**Conversion Unit**
Haemoglobin (Hb)	g/dL	g/L
Red Blood Cell (RBC)	x10^12/L	x10^6/µL
Hematocrit (HCT) | Packed Cell Volume (PCV)	%	L/L

Mean Corpuscular Volume (MCV)	fL	µm3
Mean Corpuscular Hemoglobin (MCH)	pg	fmol
Mean Corpuscular Hemoglobin Concentration (MCHC)	g/dL	g/L
Red Cell Distribution Width (RDW)	g/dL	g/L
White Blood Cell (WBC)	x10^3/µL	x10^9/L
Platelets	x10^3/µL	x10^9/L
A.C. Neutrophils	x10^3/µL	x10^9/L
A.C. Lymphocytes	x10^3/µL	x10^9/L
A.C. Monocytes	x10^3/µL	x10^9/L
A.C. Eosinophils	x10^3/µL	x10^9/L
A.C. Basophils	x10^3/µL	x10^9/L

Besides, the processing time for parsing a single page blood test report was longer than expected due to the limitations of the free tier of Azure AI Document Intelligence. It takes around 5 to 10 seconds to complete the data extraction and key field mapping process. The free tier offers limited processing capacity with restricted concurrency, meaning that the document requests may be queued or processed sequentially. Additionally, using custom extraction model takes more time than prebuilt models as it involves extra steps such as key-value pair extraction. This waiting time might cause user frustration as the delay of even a few seconds can feel like an eternity, especially if they're unsure whether the process is working.

The testing had evaluated the proposed application and concluded that the accuracy of the extraction model was lower than expected. A high character error rate with high confidence score in Azure AI Document Intelligence's custom extraction model can cause by several factors including layout drift or key value pairs mismatch. The changes in fonts, tables or overall structure deviate from the training data will confuse the model's interpretation. Additionally, insufficient training data lacking in variation, like different font types or edge cases, may limit the model's ability to adapt to unfamiliar layouts. Lastly, annotation errors, such as inaccurate labeling or misaligned bounding boxes, can compromise the learning process, resulting in suboptimal extraction accuracy.

This proposed application only supports single page document extraction due to the limitations of Azure Document AI Intelligence custom model built under free tier account. The custom model API will only extract the first page of the uploaded blood test report. This might lead to incomplete or unusable output, which could result in or errors during interpretation.

As mentioned in chapter 1 project scope, this proposed application is focusing on extracting Haematology test from the blood test report. The other test including liver function test, renal function test, diabetes mellitus screening and more will not be identified during data extraction.

6.4 Future Work

To enhance the overall functionality and user experience of the proposed application, several future works have been identified. The future works of this project will be addressing the limitations discussed on section 6.3 of this report. First, a model focusing on unit identification and conversion can be trained the expansion of unit conversion capabilities. More supporting references should be studied and analysed to build the data lake for blood test metric unit conversion. By integrating a comprehensive unit conversion module, users will be able to seamlessly convert between a wider range of measurement units, making the application more durable for various laboratory and clinical settings.

Another improvement to enhance application extraction performance involves upgrading to a paid pricing tier which provides faster processing. For integration, optimizing database operations by using batch inserts to Azure SQL can be a way to reduce processing time and resource consumption. Besides, visual user interface widgets should be used to notify the user regarding the current progress for the data extraction process.

Besides, future work should focus on expanding and diversifying the training dataset by including a wider range of document layouts to improve the model's adaptability and extraction accuracy. Additionally, augmenting the dataset with synthetic documents that introduce controlled variations such as font changes, table reflows, or page rotations can simulate layout drift and strengthen the model’s resilience. Layout-aware strategies such as leveraging tools that classify or detect document structures prior to extraction can further optimize performance by enabling dynamic routing to specialized sub-models tailored to different layout types.

Lastly, there is a need to broaden the scope of blood test metrics that can be extracted. Future iterations of the application should include a configurable feature that allows users to customize which test metrics to extract. This will cater to diverse user needs and ensure the tool remains flexible and relevant across various use cases.

6.5 Summary

In conclusion, this chapter analysed the project's achievements, limitations, and future works. The proposed Blood Test Report Data Extraction Application had successfully met its primary goals, which included studying document processing techniques, implementing an automated data extraction application for blood test reports using Microsoft Azure AI Document Intelligence, and evaluating the system’s performance through appropriate metrics. However, several limitations were identified during testing, including restricted unit conversion capabilities, long processing time, low data extraction accuracy, and limited extraction of haematology metrics only. To address these shortcomings, future work will focus on developing a more robust unit conversion model, incorporating streamlined and scaled extraction pipeline, expanding and diversifying the training dataset, and expanding the range

of extractable test metrics to support broader medical use cases. These enhancements aim to improve the system's usability and effectiveness in real-world healthcare environments.

References

Adobe. (2017). *Adobe Scan* (Version 25.01.07-google-dynamic) [Mobile app]. Google Play. https://play.google.com/store/apps/details?id=com.adobe.scan.android&hl=en

Amazon Web Services. (2019). *Amazon Textract* (Version 3.723.0) [Computer software]. Amazon Web Services. https://aws.amazon.com/textract/

Beynon-Davies, P., Carne, C., Mackay, H., & Tudhope, D. (1999). Rapid application development (RAD): an empirical review. *European Journal of Information Systems*, *8*(3), 211–223. https://doi.org/10.1057/palgrave.ejis.3000325

CamSoft Information. (2010). *CamScanner* (Version 6.79.0.2412230000) [Mobile app]. Google Play. https://play.google.com/store/apps/details?id=com.intsig.camscanner&hl=en

Cowie, M. R., Blomster, J. I., Curtis, L. H., Duclaux, S., Ford, I., Fritz, F., Goldman, S., Janmohamed, S., Kreuzer, J., Leenay, M., Michel, A., Ong, S., Pell, J. P., Southworth, M. R., Stough, W. G., Thoenes, M., Zannad, F., & Zalewski, A. (2017). Electronic health records to facilitate clinical research. *Clin Res Cardiol, 106*, 1–9. https://doi.org/10.1007/s00392-016-1025-6

Dash, B. (2021). A hybrid solution for extracting information from unstructured data using optical character recognition (OCR) with natural language processing (NLP). *Proceedings of the Conference on Big Data and Cloud Computing*. https://doi.org/10.358199755

Fausto, M. (2019). *Keras OCR* (Version 0.8.4) [Computer software]. GitHub. https://github.com/faustomorales/keras-ocr

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks*. Proceedings of the 23rd International Conference on Machine Learning (ICML).

Gokhale, K. M., Chandan, J. S., Toulis, K., Gkoutos, G., Tino, P., & Nirantharakumar, K. (2021). Data extraction for epidemiological research (DExtER): a novel tool for automated clinical epidemiology studies. *Eur J Epidemiol, 36*, 165–178. https://doi.org/10.1007/s10654-020-00677-6

Holmes, J. H., Beinlich, J., Boland, M. R., Bowles, K. H., Chen, Y., Cook, T. S., Demiris, G., Draugelis, M., Fluharty, L., Gabriel, P. E., Grundmeier, R., Hanson, C. W., Herman, D. S., Himes, B. E., Hubbard, R. A., Kahn Jr., C. E., Kim, D., Koppel, R., Long, Q., & Mirkovic, N. (2021). Why Is the Electronic Health Record So Challenging for Research and Clinical Care?. *Methods of information in medicine*, *60*(1-02), 32–48. https://doi.org/10.1055/s-0041-1731784

Kaihlanen, A. M., Gluschkoff, K., Hyppönen, H., Kaipio, J., Puttonen, S., Vehko, T., Saranto, K., Karhe, L., & Heponiemi, T. (2020). The associations of electronic health record usability and user age with stress and cognitive failures among Finnish registered nurses: Cross-sectional study. *JMIR Medical Informatics, 8*(11), e23623. https://doi.org/10.2196/23623

Hsu, E., Malagaris, I., Kuo, Y.-F., Sultana, R., & Roberts, K. (2022). Deep learning-based NLP data pipeline for EHR-scanned document information extraction. *JAMIA Open, 5*(2), 1–12. https://doi.org/10.1093/jamiaopen/ooac045

Kaihlanen, A. M., Gluschkoff, K., Hyppönen, H., Kaipio, J., Puttonen, S., Vehko, T., Saranto, K., Karhe, L., & Heponiemi, T. (2020). The associations of electronic health record usability and user age with stress and cognitive failures among Finnish registered nurses: Cross-sectional study. *JMIR Medical Informatics, 8*(11), e23623. https://doi.org/10.2196/23623

Karthikeyan, S., de Herrera, A. G. S., Doctor, F., & Mirza, A. (2022). An OCR post-correction approach using deep learning for processing medical reports. *IEEE Transactions on Circuits and Systems for Video Technology, 32*(5), 2574–2581. https://doi.org/10.1109/TCSVT.2021.3087641

Kormilitzin, A., Vaci, N., Liu, Q., & Nevado-Holgado, A. (2020). Med7: A transferable clinical natural language processing model for electronic health records. *Artificial Intelligence in Medicine, 118*, 102086. https://doi.org/10.1016/j.artmed.2021.102086

Laujan. (n.d.). *Interpret and improve model accuracy and confidence scores - Azure AI services*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/concept/accuracy-confidence?view=doc-intel-4.0.0

Li, Y. (2024). Synergizing Optical Character Recognition: A comparative analysis and integration of Tesseract, Keras, Paddle, and Azure OCR. *University of Sydney, Faculty of Engineering.*

Lufick Technology Private Limited. (2017). *Doc Scanner* (Version 6.8.4) [Mobile app]. Google Play. https://play.google.com/store/apps/details?id=com.cv.docscanner&hl=en

Martin, J. (1991). *Rapid application development*. MacMillan Publishing Company.

Material Design - Version 2. (n.d.). Material Design. https://m2.material.io/design/guidelines-overview

Melnick, E. R., Dyrbye, L. N., Sinsky, C. A., Trockel, M., West, C. P., Nedelec, L., Tutty, M. A., & Shanafelt, T. (2020). The association between perceived electronic health record usability and professional burnout among U.S. physicians. *Mayo Clinic Proceedings, 95*(3), 476–487. https://doi.org/10.1016/j.mayocp.2019.09.024

Microsoft. (n.d.). *Blob storage*. Microsoft Azure. https://azure.microsoft.com/en-gb/products/storage/blobs

Microsoft. (2025, March 3). *Interpret and improve model accuracy and confidence scores*. Azure AI Document Intelligence documentation. https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/concept-confidence-scores

Microsoft Azure. (2021). *Azure AI Document Intelligence* (Version 4.0.0) [Computer software]. Microsoft Azure. https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/?view=doc-intel-4.0.0

Microsoft Corporation. (2015). *Microsoft Lens* (Version 16.0.17425.20158) [Mobile app]. Google Play. https://play.google.com/store/apps/details?id=com.microsoft.office.officelens&hl=en

Mondal, H., & Zubair, M. (2024, October 6). *Hematocrit*. StatPearls - NCBI Bookshelf. https://www.ncbi.nlm.nih.gov/books/NBK542276/

Nandhinee, P. R., Krishnamoorthy, H., Srivatsan, K., Goyal, A., & Santhiappan, S. (2022). DEXTER: An end-to-end system to extract table contents from electronic medical health documents. *Proceedings of the Conference on Document Intelligence*. https://arxiv.org/abs/2207.06823

Online conversion calculator for many types of measurement units in laboratory and medicine practice | UnitsLab.com. (n.d.). https://unitslab.com/

PaddlePaddle. (2024). *PaddleOCR* (Version 2.9.1) [Computer software]. GitHub. https://github.com/PaddlePaddle/PaddleOCR

Putra, M. Y., & Lolly, R. W. R. (2021). Sistem Aplikasi Penjualan Souvenir Berbasis Web Menggunakan Metode Rapid Application Development (RAD). *Inf. Syst. Educ. Prof.*, *5*(2), 1-10. https://doi.org/10.51211/isbi.v5i2.1548

Shafait, F., & Smith, R. (2010). Table detection in heterogeneous documents. In D. S. Doermann, V. Govindaraju, D. P. Lopresti, & P. Natarajan (Eds.), *Document Analysis Systems* (pp. 65-72). ACM. https://dblp.uni-trier.de/db/conf/das/das 2010.html#ShafaitS10

Subramani, N., Matton, A., Greaves, M., & Lam, A. (2020). A survey of deep learning approaches for ocr and document understanding. https://doi.org/10.48550/arxiv.2011.13534

SureSwift Capital Inc. (2018). *DocParser* (Version 1.0) [Computer software]. SureSwift Capital Inc. https://docparser.com/

Tap AI. (2017). *TapScanner* (Version 3.0.51) [Mobile app]. Google Play. https://play.google.com/store/apps/details?id=pdf.tap.scanner&hl=en

Training Custom Extraction model in Azure AI Document Intelligence for medical Data - Microsoft Q&A. (n.d.). https://learn.microsoft.com/en-us/answers/questions/2223444/training-custom-extraction-model-in-azure-ai-docum?

Wang, B., Lai, J., Liu, M., Jin, F., Peng, Y., & Yao, C. (2022). Electronic source data transcription for electronic case report forms in China: Validation of the electronic source record tool in a real-world ophthalmology study. *JMIR Formative Research, 6*(12), e43229. https://doi.org/10.2196/43229

WilliamDAssafMSFT. (n.d.). *What is the Azure SQL Database service? - Azure SQL Database*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql

Xu, T., Zhang, Y., Wu, X., & Ming, W. (2021). Intelligent document processing automate business with fluid workflow. *Konica Minolta Technology Report*, *18*, 86-94. https://research.konicaminolta.com/jp/pdf/technology_report/2021/pdf/18_xu.pdf

Yan, J., Kloecker, G., Fleming, C., Bousamra, M., Hansen, R., Hu, X., Ding, C., Cai, Y., Xiang, D., Donninger, H., Eaton, J. W., & Clark, G. J. (2014). Human polymorphonuclear neutrophils specifically recognize and kill cancerous cells. *OncoImmunology*, *3*(7), e950163. https://doi.org/10.4161/15384101.2014.950163

Appendices

Appendix A: Blood Test Report Health Metrics

Test	**Health Metrics**		**Expected Unit**
	Full Form	Abbreviation	
Haematology	Total Red Blood Cell	RBC	x10^12/L
	Haemoglobin	Hb	g/dL
	Packed Cell Volume	PCV	L/L
	Mean Corpuscular Volume	MCV	fL
	Mean Corpuscular Hemoglobin	MCH	pg
	Mean Corpuscular Hemoglobin Concentration	MCHC	g/L
	Red Blood Cell Distribution Width	RDW	%
	White Blood Cell	WBC	x10^9/L
	Platelet	-	x10^9/L
	Erythrocyte Sedimentation Rate	ESR	mm/hr
Differential Count	Polymorphs	D.C. Polymorphs	%
	Lymphocytes	D.C. Lymphocytes	%
	Monocytes	D.C. Monocytes	%
	Eosinophils	D.C. Eosinophils	%
	Basophils	D.C. Basophils	%
Absolute Count	Polymorphs	A.C Polymorphs	x10^9/L
	Lymphocytes	A.C Lymphocytes	x10^9/L
	Monocytes	A.C Monocytes	x10^9/L
	Eosinophils	A.C Eosinophils	x10^9/L
	Basophils	A.C Basophils	x10^9/L
Blood Bank	ABO (Grouping)	-	A, B, AB, O
	Rh (Anti-D)	Rh(D)	Positive/Negative

Note: Polymorphs and Neutrophils are interchangeable (Yan et al., 2014). Same goes to Packed Cell Volume (PCV) and Haematocrit (HCT) (Mondal & Zubair, 2024).

Appendix B: User Requirement Questionnaire

Automatic Text Extraction from Printed Blood Test Report into Digital Data using AI Technology

Greetings! We are Kueh Hui Tieng (**Bachelor of Computer Science with Honours (Information Systems)**) and Ho Jia Wei (**Bachelor of Computer Science with Honours (Multimedia Computing)**), final-year undergraduate students from **Faculty of Computer Science and Information Technology (FCSIT), University Malaysia Sarawak (UNIMAS)**.

We would like to invite you to fill this questionnaire (5 - 10 minutes), as your input is highly valuable and greatly appreciated.

This questionnaire contains five (5) section:

- **Section A: General Information**
- **Section B: System Features, Usability and Compatibility**
- **Section C: Data Accuracy and Validation**
- **Section D: Reporting and Data Export**
- **Section E: Additional Requirements**

The insights collected through this questionnaire will help to better understand the current needs in extracting blood test data from printed reports, as well as gathering the requirements from medical personnel to be used in implementing the automatic text extraction tool using AI technology (Optical Character Recognition and Natural Language Processing). This tool will ease the digitization of data extraction from printed blood test reports for easy record management and storing purposes.

Your responses will remain confidential and will be used solely for the purpose of this project.

If you have any questions, feel free to contact us through **79802@siswa.unimas.my** (Kueh Hui Tieng) or **79545@siswa.unimas.my** (Ho Jia Wei).

Thank you!

Sincerely,
Ho Jia Wei,
Kueh Hui Tieng

hojiaweii0817@gmail.com Switch accounts

Not shared

* Indicates required question

By proceeding with this survey, you consent to participate in this study and acknowledge that you have read and understood the information provided above. *

◯ I agree

Section A: General Information

Occupation *

- ◯ Doctor
- ◯ Nurse
- ◯ Other: _____

How many blood test reports do you process daily? *

- ◯ 1 - 5
- ◯ 6 - 10
- ◯ 11 - 20
- ◯ 21 and above

Describe the issue faced in extracting data from patients' blood test report? *

Your answer

How would you think an application focus on automatic text extraction from blood test report would be useful in your current/future occupation? *

Your answer

Section B: System Features, Usability and Compatibility

How do you manage and record your patients' blood test reports? *

- [ ] Manually transcript the blood test data on paper
- [ ] Scan and save a hardcopy of the document (eg. photostat, printed report)
- [ ] Scan and save a softcopy of the document (eg. pdf, image)
- [ ] Save the softcopy of the document in local device (mobile phone, tablet, laptop, etc.)
- [ ] Save the softcopy of the document on cloud storage service
- [ ] Other: _____

What key features would you expect from an automatic text extraction from blood test report system/application? (Select all that apply) *

- [ ] Automatic text extraction by scanning using camera
- [ ] Highlight abnormal test results
- [ ] Export data (eg. Excel)
- [ ] Other: _____

Which device do you prefer for using the automatic text extraction tool? *

- [ ] Laptop/Personal computer
- [ ] Tablet
- [ ] Mobile phone

What patient information do you need to store from the blood test reports? *

- [ ] Name
- [ ] IC/Passport Number
- [ ] Age
- [ ] Gender
- [ ] Date Of Birth
- [ ] Blood Collection Date
- [ ] Doctor Details
- [ ] Name of Clinic for requesting blood test
- [ ] Name of Blood Test Laboratory (eg.Pathlab, Innoquest)
- [ ] Other: _____

Section C: Data Accuracy and Validation

Do you prefer a system/application that allows manual corrections after the data extraction process? *

- ◯ Yes, manual correction is needed
- ◯ No, only automated data extraction is sufficient
- ◯ Other: _____

Which action should be taken by the system/application if failed to completely extract the data? (eg. data missing) *

- ☐ Notify the user for manual review
- ☐ Automatically fill up the data using 'NA'
- ☐ Other: _____

Do you think user authentication would help to enhance the security and privacy of the system/application? (eg. user log in/log out feature) *

- ◯ Yes
- ◯ No

For automatic system record deletion, how long should the records to be stored in the system/application? *

- ◯ 6 months
- ◯ 1 year
- ◯ 2 years
- ◯ Other: _____

Section D: Data Export and Reporting

What type of data visualization tools would you prefer? *

- [ ] Table
- [ ] Line Chart
- [ ] Bar Chart
- [ ] Pie Chart
- [ ] Scatter Plot
- [ ] Not needed

Which format of digitised blood test report would be useful for your needs? *

- [ ] Detailed Report, I prefer a detailed report with all test metrics.
- [ ] Summary Report, I want a summary report that shows only the abnormal values.
- [ ] Customizable Report, I would like to have a feature to select metrics appear in the report.
- [ ] Other: _____

How do you search for a specific patient record in Electronic Medical Record(EMR)? *

- [ ] Search by patient name
- [ ] Search by patient IC/passport number
- [ ] Search by patient blood collection date
- [ ] Other: _____

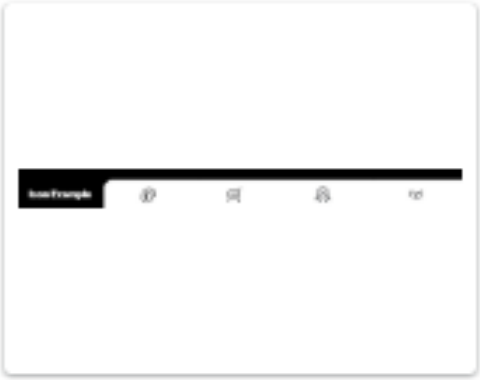
Section E: Additional Requirements

What key data would you like to see at a glance when you first open the dashboard? *

- [ ] Patient count
- [ ] Abnormalities count
- [ ] Recent test result
- [ ] Other: _____

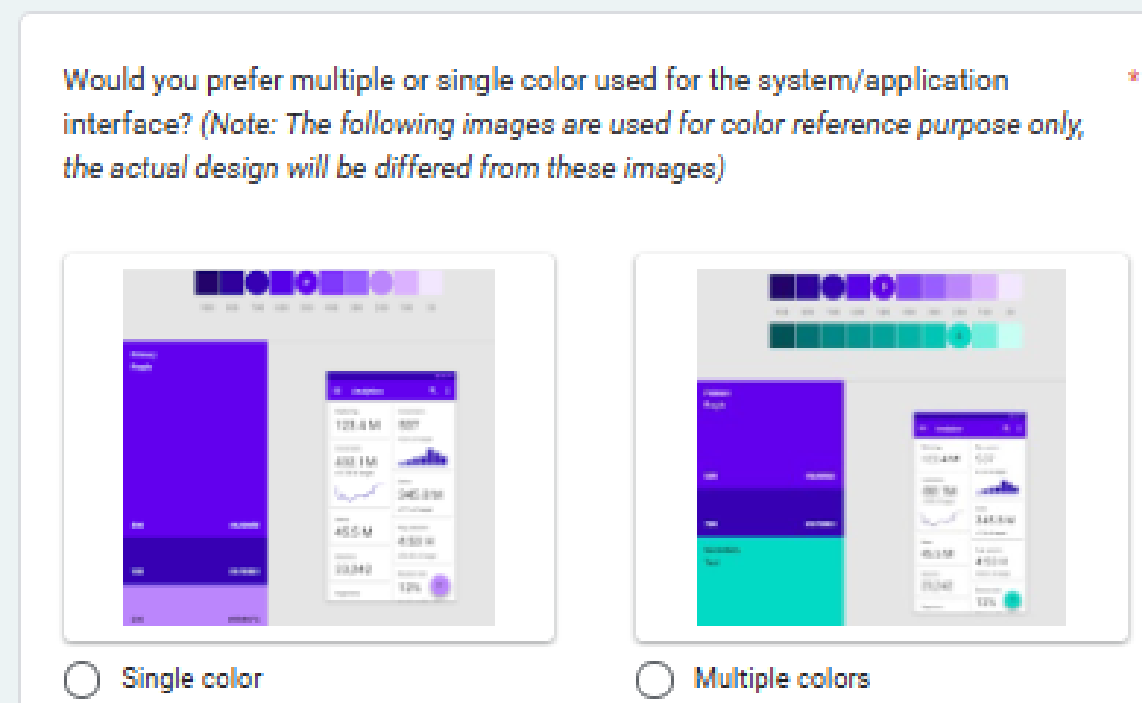
Would you prefer text or icon in the system/application interface? *

- ◯ Text



- ◯ Icon

- ◯ Both



Question	Options
Section A: General Information	
By proceeding with this survey, you consent to participate in this study and acknowledge that you have read and understood the information provided above.	• I agree
Occupation	• Doctor • Nurse • Other:
How many blood test reports do you process daily?	• 1 - 5 • 6 - 10 • 11 - 20 • 21 and above
Describe the issue faced in extracting data from patients' blood test report?	Open-ended question
How would you think an application focus on automatic text extraction from blood test	Open-ended question

report would be useful in your current/future occupation?	
Section B: System Features, Usability and Compatibility	
How do you manage and record your patients' blood test reports? Tick all that apply.	• Manually transcript the blood test data on paper • Scan and save a hardcopy of the document (eg. photostat, printed report) • Scan and save a softcopy of the document (eg. pdf, image) • Save the softcopy of the document in local device (mobile phone, tablet, laptop, etc.) • Save the softcopy of the document on cloud storage service • Other:
What key features would you expect from an automatic text extraction from blood test report system/application? Tick all that apply.	• Automatic text extraction by scanning using camera • Highlight abnormal test results • Export data (eg. Excel) • Other:
Which device do you prefer for using the automatic text extraction tool? Tick all that apply.	• Laptop/Personal computer • Tablet • Mobile phone
What patient information do you need to store from the blood test reports? Tick all that apply.	• Name • IC/Passport Number • Age • Gender • Date Of Birth • Blood Collection Date • Doctor Details • Name of Clinic for requesting blood test • Name of Blood Test Laboratory (eg.Pathlab, Innoquest) • Other:
Section C: Data Accuracy and Validation	
Do you prefer a system/application that allows manual corrections after the data extraction process?	• Yes, manual correction is needed • No, only automated data extraction is sufficient • Other:
Which action should be taken by the system/application if failed to completely extract the data? (eg. data missing) Tick all that apply.	• Notify the user for manual review • Automatically fill up the data using 'NA' • Other:
Do you think user authentication would help to enhance the security and privacy of the system/application? (eg. user log in/log out feature)	• Yes • No

For automatic system record deletion, how long should the records to be stored in the system/application?	• 6 months • 1 year • 2 years • Other
Section D: Data Export and Reporting	
What type of data visualization tools would you prefer? Tick all that apply	• Table • Line Chart • Bar Chart • Pie Chart • Scatter Plot • Not needed
Which format of digitised blood test report would be useful for your needs? Tick all that apply	• Detailed Report, I prefer a detailed report with all test metrics. • Summary Report, I want a summary report that shows only the abnormal values. • Customizable Report, I would like to have a feature to select metrics appear in the • report. • Other:
How do you search for a specific patient record in Electronic Medical Record (EMR)? Tick all that apply.	• Search by patient name • Search by patient IC/passport number • Search by patient blood collection date • Other
Section E: Additional Requirements	
What key data would you like to see at a glance when you first open the dashboard? Tick all that apply.	• Patient count • Abnormalities count • Recent test result • Other:
Would you prefer text or icon in the system/application interface?	• Text • Icon • Both
Would you prefer multiple or single color used for the system/application interface? (Note: The following images are used for color reference purpose only, the actual design will be differed from these images)	• Single color • Multiple colors
What other features would you suggest being included in the automatic text extraction from blood test report system/application?	Open-ended question

Appendix C: List of Blood Test Report Layout Used for Model Training

No.	Blood Test Laboratory	Number of Report
1	Borneo Medical Centre	5
2	BP Lab	6
3	Colombia Asia Hospital Bukit Rimau	1
4	Drlogy Pathology Lab	1
5	Gribbles Pathology (Malaysia) Sdn Bhd	5
6	Innoquest Pathology Sdn Bhd	5
7	Navipath Diagnostics Sdn Bhd	4
8	Pantai Premier Pathology	5
9	Pathlab Pathology & Clinical Laboratory (M) Sdn Bhd	4
10	Quantum Diagnostic Sdn Bhd	1
11	Timberland Medical Centre	3
12	University of Malaya Medical Centre	5

Appendix D: Content of Verification Page

Page No.	Page Title	Content	
		Field/Parameter Name	**Key Extraction Field**
1	Personal Information	• Patient Name • Patient IC/Passport Number • Patient Date of Birth • Patient Age • Patient Gender • Report Date • Report Laboratory Name • Patient Blood Group • Patient Rh(D)	
2	Haematology Test	• Red Blood Cell (RBC) • White Blood Cell (WBC) • Hematocrit (HCT) | Packed Cell Volume (PCV) • Mean Corpuscular Volume (MCV) • Haemoglobin (Hb) • Mean Corpuscular Hemoglobin (MCH) • Mean Corpuscular Hemoglobin Concentration • Platelets • Red Cell Distribution Width (RDW) • Erythrocyte Sedimentation Rate (ESR)	Value, Unit, and Reference Range
3	Absolute Count	• A.C. Neutrophils • A.C. Lymphocytes • A.C. Monocytes • A.C. Eosinophils • A.C. Basophils	Value, Unit, and Reference Range
4	Differential Count	• D.C. Neutrophils • D.C. Lymphocytes • D.C. Monocytes • D.C. Eosinophils • D.C. Basophils	Value, Unit, and Reference Range

Appendix E: User Acceptance Testing Questionnaire

Automatic Data Extraction from Blood Test Report Using Microsoft Azure AI Document Intelligence

Hi, I am Ho Jia Wei (79545), a final year student from Bachelor of Computer Science (Multimedia Computing) course and I am currently working on my final year project. I'd truly appreciate it if you could take a few minutes to test on my application and then fill out a Google Form to share your feedback. Your insights will help me improve the usability and overall experience.

This form is a **System Usability Scale (SUS)** questionnaire consist of 10 questions, which was used to evaluate the overall usability of an application. Kindly reflect your true user experience while using the MediExtract app.

Each statement should be rated on a **5-point Likert scale**, from **Strongly Disagree (1)** to **Strongly Agree (5)**. There are no right or wrong answers — just your personal opinion.

Tested Email *

Short-answer text

1. I think that I would like to use this application frequently. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

2. I found this application was unnecessarily complex. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

3. I think this application was easy to use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

4. I think that I would need the support of a technical person to be able to use this application. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

5. I found the various functions in this application were well integrated. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

6. I think there was too much inconsistency in this tool. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

7. I learn to use this application very quickly. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

8. I found the application very cumbersome to use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

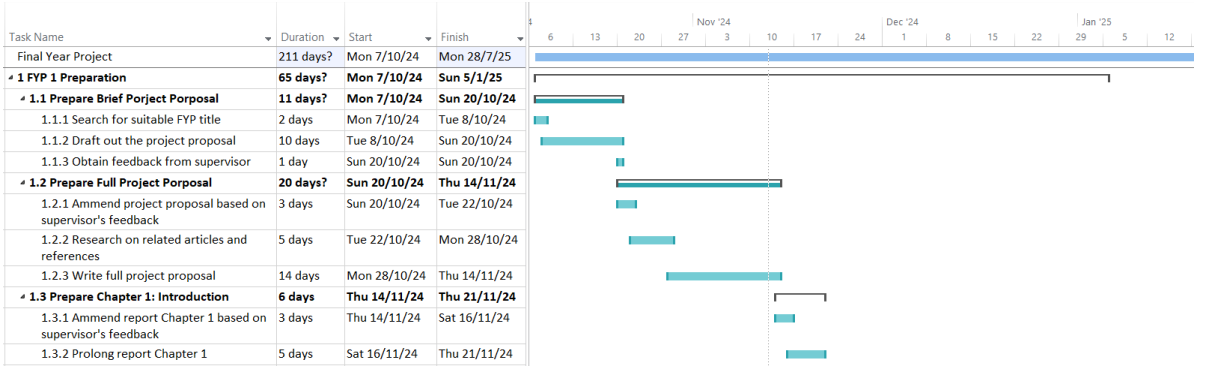
9. I felt very confident using the application. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

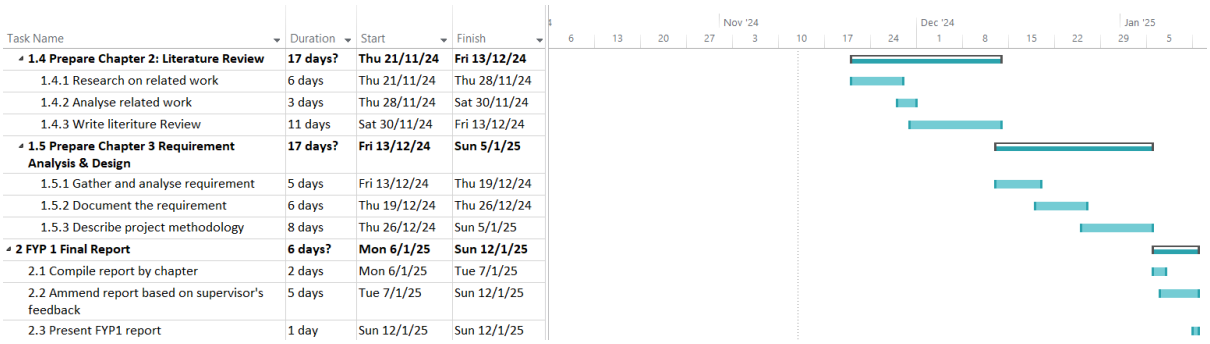
10. I needed to learn a lot of things before I could get going with this application. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

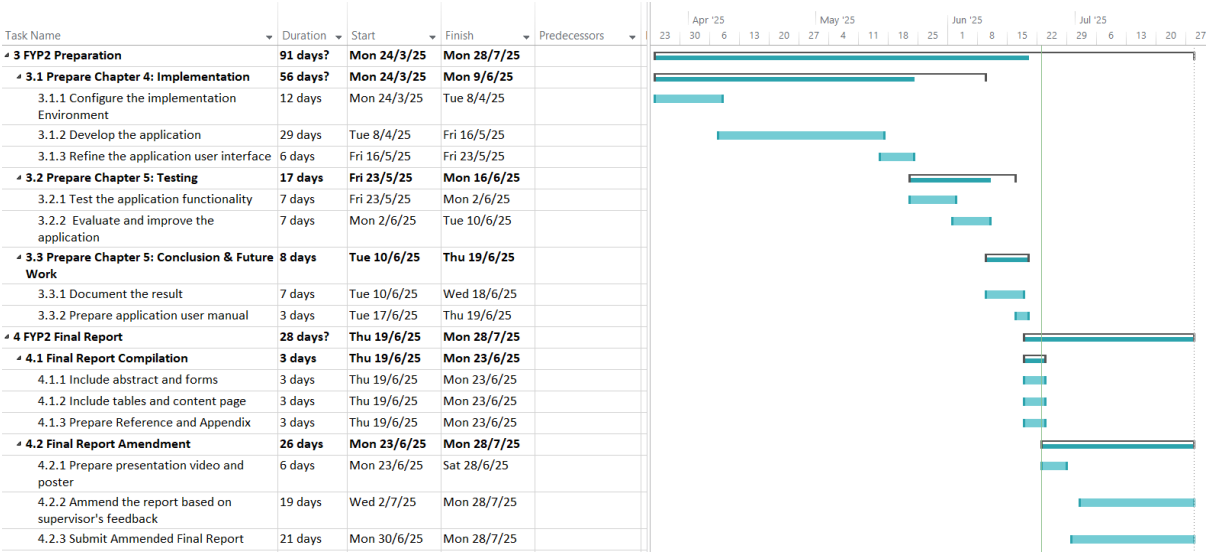
Appendix F: Gantt Chart



Task Name	Duration	Start	Finish
Final Year Project	211 days?	Mon 7/10/24	Mon 28/7/25
1 FYP 1 Preparation	**65 days?**	**Mon 7/10/24**	**Sun 5/1/25**
1.1 Prepare Brief Porject Porposal	**11 days?**	**Mon 7/10/24**	**Sun 20/10/24**
1.1.1 Search for suitable FYP title	2 days	Mon 7/10/24	Tue 8/10/24
1.1.2 Draft out the project proposal	10 days	Tue 8/10/24	Sun 20/10/24
1.1.3 Obtain feedback from supervisor	1 day	Sun 20/10/24	Sun 20/10/24
1.2 Prepare Full Project Porposal	**20 days?**	**Sun 20/10/24**	**Thu 14/11/24**
1.2.1 Ammend project proposal based on supervisor's feedback	3 days	Sun 20/10/24	Tue 22/10/24
1.2.2 Research on related articles and references	5 days	Tue 22/10/24	Mon 28/10/24
1.2.3 Write full project proposal	14 days	Mon 28/10/24	Thu 14/11/24
1.3 Prepare Chapter 1: Introduction	**6 days**	**Thu 14/11/24**	**Thu 21/11/24**
1.3.1 Ammend report Chapter 1 based on supervisor's feedback	3 days	Thu 14/11/24	Sat 16/11/24
1.3.2 Prolong report Chapter 1	5 days	Sat 16/11/24	Thu 21/11/24



Task Name	Duration	Start	Finish
1.4 Prepare Chapter 2: Literature Review	**17 days?**	**Thu 21/11/24**	**Fri 13/12/24**
1.4.1 Research on related work	6 days	Thu 21/11/24	Thu 28/11/24
1.4.2 Analyse related work	3 days	Thu 28/11/24	Sat 30/11/24
1.4.3 Write literiture Review	11 days	Sat 30/11/24	Fri 13/12/24
1.5 Prepare Chapter 3 Requirement Analysis & Design	**17 days?**	**Fri 13/12/24**	**Sun 5/1/25**
1.5.1 Gather and analyse requirement	5 days	Fri 13/12/24	Thu 19/12/24
1.5.2 Document the requirement	6 days	Thu 19/12/24	Thu 26/12/24
1.5.3 Describe project methodology	8 days	Thu 26/12/24	Sun 5/1/25
2 FYP 1 Final Report	**6 days?**	**Mon 6/1/25**	**Sun 12/1/25**
2.1 Compile report by chapter	2 days	Mon 6/1/25	Tue 7/1/25
2.2 Ammend report based on supervisor's feedback	5 days	Tue 7/1/25	Sun 12/1/25
2.3 Present FYP1 report	1 day	Sun 12/1/25	Sun 12/1/25



Task Name	Duration	Start	Finish	Predecessors
3 FYP2 Preparation	**91 days?**	**Mon 24/3/25**	**Mon 28/7/25**	
3.1 Prepare Chapter 4: Implementation	**56 days?**	**Mon 24/3/25**	**Mon 9/6/25**	
3.1.1 Configure the implementation Environment	12 days	Mon 24/3/25	Tue 8/4/25	
3.1.2 Develop the application	29 days	Tue 8/4/25	Fri 16/5/25	
3.1.3 Refine the application user interface	6 days	Fri 16/5/25	Fri 23/5/25	
3.2 Prepare Chapter 5: Testing	**17 days**	**Fri 23/5/25**	**Mon 16/6/25**	
3.2.1 Test the application functionality	7 days	Fri 23/5/25	Mon 2/6/25	
3.2.2 Evaluate and improve the application	7 days	Mon 2/6/25	Tue 10/6/25	
3.3 Prepare Chapter 5: Conclusion & Future Work	**8 days**	**Tue 10/6/25**	**Thu 19/6/25**	
3.3.1 Document the result	7 days	Tue 10/6/25	Wed 18/6/25	
3.3.2 Prepare application user manual	3 days	Tue 17/6/25	Thu 19/6/25	
4 FYP2 Final Report	**28 days?**	**Thu 19/6/25**	**Mon 28/7/25**	
4.1 Final Report Compilation	**3 days**	**Thu 19/6/25**	**Mon 23/6/25**	
4.1.1 Include abstract and forms	3 days	Thu 19/6/25	Mon 23/6/25	
4.1.2 Include tables and content page	3 days	Thu 19/6/25	Mon 23/6/25	
4.1.3 Prepare Reference and Appendix	3 days	Thu 19/6/25	Mon 23/6/25	
4.2 Final Report Amendment	**26 days**	**Mon 23/6/25**	**Mon 28/7/25**	
4.2.1 Prepare presentation video and poster	6 days	Mon 23/6/25	Sat 28/6/25	
4.2.2 Ammend the report based on supervisor's feedback	19 days	Wed 2/7/25	Mon 28/7/25	
4.2.3 Submit Ammended Final Report	21 days	Mon 30/6/25	Mon 28/7/25	