



Faculty of Computer Science and Information Technology

**AIBeautician: An AI-Powered Mobile App for Personalized Cosmetic Ingredient
Analysis and Recommendations**

Nur Syahirah binti Mohamad Abu Bakar

Bachelor of Software Engineering with Honours
2025

**AIBEAUTICIAN: AN AI-POWERED MOBILE APP FOR PERSONALIZED
COSMETIC INGREDIENT ANALYSIS AND RECOMMENDATIONS**

NUR SYAHIRAH BINTI MOHAMAD ABU BAKAR

This project is submitted in partial fulfilment of
The requirements for the degree of Bachelor of Software Engineering with Honours

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

2025

**AIBEAUTICIAN: APLIKASI MUDAH ALIH DIKUASAKAN AI UNTUK
ANALISIS RAMUAN KOSMETIK DAN PENGESYORAN BAHAN KOSMETIK
PERIBADI**

NUR SYAHIRAH BINTI MOHAMAD ABU BAKAR

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Kejuruteraan Perisian dengan Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I hereby declare that this project is my original work. I have not copied from any other student's work or from any other sources except where due reference or acknowledgement is not made explicitly in the text, nor has any part had been written for me by another person.



(NUR SYAHIRAH BINTI MOHAMAD
ABU BAKAR)

18 /7 /2025

Date

ACKNOWLEDGMENT

Foremost, I would like to express my sincere gratitude to my supervisor, Professor Ts. Dr. Dayang Iskandar, for granting me the opportunity to work under her supervision and providing invaluable guidance and support throughout this project. Her passion and encouragement greatly helped me in completing my Final Year Project. In addition, I would like to express my gratitude towards the project's examiner, Dr. Liew Siaw Hong, for her constructive feedback and guidance during the course of the project. Not forgetting, I would like to thank Professor Dr. Wang Yin Chai for the many helpful insightful advice and tips he generously shared with us during the lectures. I would also like to extend my deepest gratitude and deeply grateful to everyone who willingly participated in my research, especially those who contributed by completing my questionnaire and offering valuable insights. Lastly, I would also like to thank and express my heartfelt appreciation to my beloved family and friends for their unwavering support, advice, and encouragement during this project. Without their love and continuous support, both physically and mentally, I would not have been able to complete this project successfully.

ABSTRACT

AIBeautician: An AI-Powered Mobile App for Personalized Cosmetic Ingredient Analysis and Recommendations is mobile application developed to address challenges faced by cosmetic user in understanding product formulations. Differentiating between silicone – based an water – based products remains a challenge for many consumers, often leading to uneven makeup application and dissatisfaction with product performance. Furthermore, concerns about the potential health risks associated with silicone – based ingredients have aroused extensive debate. While some studies claim that silicone may cause hypersensitivity reaction in particular individuals, other research indicates that it is generally safe for use in cosmetics. Despite these discussion, there is a considerable paucity of scientific evidence, particularly in Malaysia, to support or refute these claims conclusively. This knowledge gap contributes to consumer uncertainty, highlighting the importance for reliable and data – driven solutions to enhance informed decision – making in cosmetic product selection. This project also aimed to analyse and provide personalised recommendation tailored to individual preferences. By incorporating advanced text recognition technology, the application simplifies the process of identifying ingredient details directly from product label, thereby enhancing usability and accuracy. Utilising a structured Rapid Application Development (RAD) methodology, this innovation fosters user informed decision making, promotes user satisfaction, and improves the entire experience while looking for cosmetic products.

ABSTRAK

AIBeautician: Aplikasi Mudah Alih Dikuasakan AI untuk Analisis Ramuan Kosmetik dan Pengesyoran Bahan Kosmetik Peribadi ialah aplikasi mudah alih yang dicipta untuk menangani cabaran yang dihadapi oleh pengguna kosmetik dalam memahami formulasi produk. Membezakan antara produk berasaskan silikon dan air masih menjadi cabaran bagi ramai pengguna kosmetik, yang sering mengakibatkan aplikasi solekan yang tidak sekata ke wajah dan ketidakpuasan terhadap prestasi produk tersebut. Selain itu, kebimbangan mengenai potensi risiko kesihatan yang dikaitkan dengan bahan berasaskan silikon telah mencetuskan perdebatan meluas. Walaupun beberapa kajian mendakwa bahawa silikon boleh menyebabkan reaksi hipersensitiviti kepada individu tertentu, penyelidikan lain menunjukkan bahawa bahan ini secara amnya selamat digunakan dalam kosmetik. Namun, terdapat kekurangan bukti saintifik yang ketara, terutamanya di Malaysia, untuk menyokong atau menyangkal dakwaan ini secara konklusif. Jurang pengetahuan ini menyumbang kepada ketidakpastian pengguna, sekaligus menekankan kepentingan penyelesaian yang boleh dipercayai dan berpandukan data bagi meningkatkan pemilihan produk kosmetik yang lebih berinformasi dan tepat. Projek ini juga bertujuan untuk menganalisis dan memberikan cadangan produk kosmetik secara peribadi yang disesuaikan dengan keutamaan individu. Dengan menggabungkan teknologi pengecaman teks yang canggih, aplikasi ini memudahkan proses mengenal pasti butiran bahan ramuan secara langsung daripada label produk, sekali gus meningkatkan kebolehgunaan dan ketepatan. Menggunakan metodologi Pembangunan Aplikasi Rapid (RAD) berstruktur, inovasi ini membantu pengguna dalam membuat keputusan yang lebih tepat, meningkatkan kepuasan, dan memperbaiki pengalaman keseluruhan semasa memilih produk kosmetik.

TABLE OF CONTENTS

DECLARATION	i
ACKNOWLEDGMENT	ii
ABSTRACT	iii
ABSTRAK.....	iv
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Project Description	1
1.3 Problem Statement	2
1.4 Objective	3
1.5 Scope	3
1.6 Brief Methodology	3
1.7 Significance of Project	5
1.8 Project Schedule	6
1.9 Expected Outcome	6
1.10 Report Outline	6
1.11 Summary	8
2 CHAPTER 2: LITERATURE REVIEW	9
2.1 Introduction	9
2.2 Background Study	9
2.3 Background of Existing Application	10
2.3.1 INCI Beauty	10
2.3.2 Safely: Cosmetic Ingredients	14
2.3.3 Little Bean – Cosmetic Checker.....	16
2.4 Comparison between the existing application.....	17
2.5 Limitations Of The Existing Products.....	19
2.6 Comparison Between The Proposed Application With The Existing Applications .	19

2.7	Discussion	20
2.8	Summary	20
3	CHAPTER 3: METHODOLOGY	21
3.1	Introduction	21
3.2	Rapid Application Development	21
3.2.1	Phase 1: Analysis and Quick Design	22
3.2.1.1	Questionnaire	23
3.2.1.2	Software Requirement.....	30
3.2.1.3	Hardware Requirement	31
3.2.2	Phase 2: Prototype Cycles	32
3.2.2.1	Flowchart.....	33
3.2.2.2	Activity Diagram.....	35
3.2.2.3	Sequence Diagram.....	36
3.2.2.4	Class Diagram	37
3.2.2.5	Use Case Diagram	38
3.2.2.6	Proposed User Interface	39
3.2.3	Phase 3: Construction and Testing	41
3.2.4	Phase 4: Implementation	41
3.3	Summary	42
4	CHAPTER 4: SYSTEM IMPLEMENTATION	43
4.1	Introduction	43
4.2	Implementation of the Application.....	43
4.2.1	Software Tools Setup.....	44
4.2.1.1	Flutter	44
4.2.1.2	Android Studio	46
4.2.1.3	Xcode	49
4.2.1.4	CocoaPods.....	49
4.2.2	Mobile Application Development	49
4.3	Summary	61
5	CHAPTER 5: SYSTEM TESTING.....	62
5.1	Introduction	62

5.2	Functional Testing	62
5.2.1	Test Case for Live Text Recognition	63
5.2.2	Test Case for Gallery Text Recognition	64
5.2.3	Test Case for Ingredient Processing And Classification	64
5.2.4	Test Case for Product Recommendation	65
5.2.5	Test Case for Category Filtering	65
5.3	Non-Functional Testing	66
5.4	Summary	66
6	CHAPTER 6: CONCLUSION AND FUTURE WORK.....	68
6.1	Introduction	68
6.2	Achievements	68
6.3	Limitation	69
6.4	Future works.....	69
	REFERENCES.....	71
	APPENDIX.....	74

LIST OF FIGURES

Figure 1.1 Rapid Application Development (RAD) Methodology Cycle.....	4
Figure 1.2 Gantt Chart for AI Beautician Application (FYP 1 & FYP 2)	6
Figure 2.1. Statistic of the general population vs. tech executives that are aware of various AI uses.	10
Figure 2.2 Home page of INCI beauty application	11
Figure 2.3 Example of barcode	12
Figure 2.4 Statistic of the user using search engine compared to barcode scanner	12
Figure 2.5 The coloured system used by the INCI beauty application	13
Figure 2.6 The application built-in camera and scanner	14
Figure 2.7 Example of harmful substances and where the concerns found results	15
Figure 2.8 The application homepage with three features	16
Figure 2.9 Built-in camera used in the app	17
Figure 3.1 Rapid Application Development (RAD) Methodology Cycle.....	22
Figure 3.2 Pie chart of distribution of consumers by age group	23
Figure 3.3 Pie chart of the distribution of consumers by gender	24
Figure 3.4 Distribution of respondents by role or occupation.....	24
Figure 3.5 Frequency of cosmetic product use among respondents	25
Figure 3.6 Criteria for choosing cosmetic products among respondents	25
Figure 3.7 Pie chart of respondent's experiences with patchy or uneven makeup	26
Figure 3.8 Awareness of differences between silicone-based and water-based cosmetic	26
Figure 3.9 Difficulty in analysing the differences between cosmetic product based	27
Figure 3.10 Respondent's habits of checking ingredient lists when purchasing	27
Figure 3.11 Usage of mobile application for scanning and analysing cosmetic product among respondents.....	28

Figure 3.12 Most useful features in a mobile application for cosmetic analysis	28
Figure 3.13 Preference for a simple vs. featured rich cosmetic application	29
Figure 3.14 Main frustration with the current checker applications or tools	30
Figure 3.15 Flowchart of the proposed AIBeautician application	33
Figure 3.16 Activity Diagram of workflows within the application	35
Figure 3.17. Sequence Diagram of the proposed application	36
Figure 3.18 Class Diagram of the proposed application	37
Figure 3.19 Use Case Diagram of the proposed application	38
Figure 3.20 Homepage of the application	39
Figure 3.21 Result and product recommendation interface	40
Figure 4.1 Flutter Download Page	45
Figure 4.2 Installation Bundle For Ios	45
Figure 4.3 Flutter path code	46
Figure 4.4 Flutter version info.....	46
Figure 4.5 Android Studio IDE Download Page.....	47
Figure 4.6 Android Studio version	47
Figure 4.7 Flutter plugin installation.....	47
Figure 4.8 New Flutter project creation	48
Figure 4.9 Successful Flutter project creation.....	48
Figure 4.10. Cocoapods Version.....	49
Figure 4.11. Recognizer Main Screen	50
Figure 4.12. Camera Plugin.....	50
Figure 4.13. Capture Picture Code Segment.....	50
Figure 4.14. Camera Permission Code.....	51
Figure 4.15. Gallery	51
Figure 4.16. Image Picker Plugin.....	51

Figure 4.17. Image Picker Code Segment.....	52
Figure 4.18. Gallery Permission Code	52
Figure 4.19. Flip Camera Code Segment	53
Figure 4.20. Google ML Kit Plugin	53
Figure 4.21. Image Process Code Segment.....	53
Figure 4.22. Text Recognition Result Page	54
Figure 4.23. First 6 Ingredient Code Segment	54
Figure 4.24. First 6 Ingredient Result Page.....	55
Figure 4.25. Displaying Result Code Segment	55
Figure 4.26. Base Type Classification Code Segment	56
Figure 4.27. Water-Based Base Type Result	56
Figure 4.28. Silicone-Based Base Type Result	57
Figure 4.29. Recommendation Card Code Segment	58
Figure 4.30 Example Of Images	58
Figure 4.31. SQLite Plugin	59
Figure 4.32. SQLite Database Code Segment.....	59
Figure 4.33. Product Model Code Segment	60
Figure 4.34. Data Sample Code Segment	61

LIST OF TABLES

Table 2.3.1 Comparison Table Between Existing Applications	18
Table 2.5.1 Comparison Between the Existing Application and Proposed Application	19
Table 3.2.1 Software Requirement	30
Table 3.2.2 Hardware Requirement.....	32
Table 5.2.1. Test Case Module With Description	62
Table 5.2.2 Test Case For Live Test Recognition	63
Table 5.2.3 Test Case For Gallery Text Recognition.....	64
Table 5.2.4 Test Case For Ingredient Processing And Classification.....	64
Table 5.2.5 Test Case For Product Recommendation.....	65
Table 5.2.6 Test Case For Category Filtering.....	65
Table 5.3.1 Non-Functional Test	66
Table 6.2.1 Project Achievement.....	68

LIST OF ABBREVIATIONS

API	:	Application Programming Interface
FYP	:	Final Year Project
IDE	:	Integration Development Environment
iOS	:	iPhone Operating System
IoT	:	Internet of things
ML	:	Machine Learning
OCR	:	Optical Character Recognition
RAD	:	Rapid Application Development
SDK	:	Software Development Kit
SQL	:	Structured Query Language
UI	:	User Interface

CHAPTER 1: INTRODUCTION

1.1 Background

The AIBeautician project was inspired by a social media personality who posted a video on TikTok in early 2023. The video claimed that silicone-based and water-based makeup, when combined, result in patchy makeup. Within the beauty community, this theory gained significant attention, sparking widespread discussion. The notion was further reinforced by the opinions of numerous beauty product users.

This assertion was recently refuted by another influencer, Alexis, who stated in her TikTok video that water-based and silicone-based makeup can, in fact, be used together. This conflicting information has caused confusion among beauty enthusiasts regarding whether these two types of makeup can be safely mixed.

Furthermore, there has been ongoing discussion regarding the possible negative effects of cosmetic products containing silicone, particularly to specific skin type. Many claim that such products are harmless, non-toxic, and have several benefits because it contained common ingredients. However, silicone may cause skin irritation, especially in those that have silicone allergy (ADA Cosmetic, 2024).

With that in mind, the aim of this project is to design a system that can detect and analyse the ingredients of makeup products, specifically identifying whether they are water-based or silicone-based, in order to provide users with accurate and reliable guidance.

1.2 Project Description

AIBeautician is an intelligent application to identify whether beauty products are silicone-based or water-based. It also recommends similar makeup products based on the same product-based formulation. This system enhances users' understanding of product compositions, resulting in better makeup outcomes and better choices according to their skin type conditions.

1.3 Problem Statement

There have been numerous discussions within the beauty industry, about the downside of silicone-based cosmetic products, particularly on social media. However, there is a lack of studies or any evidence to support this particular issue, both within Malaysia and internationally. While, some individuals claim that it can be resulting in uneven makeup result when mixed both silicon-based and water-based product, but others disagree, which gives rise to mixed feelings and further confusion among the mass consumer of cosmetic product (C. Basias, 2023).

However, makeup can still be patchy on skin even when using a quality formulas and there's reason this could be happening because the makeup ingredient either water-based or silicone-based, and mixing the two will result in a spotty mess (Spectrum Collections, 2023). This difficulty can result in the selection of unsuitable cosmetics products, affecting product performance and user satisfaction.

In addition to the issue of combining silicones with water-based products, consumers also wary of the potential harm related to silicone ingredients. Indeed, research indicates that silicone can be harmful to certain individuals. For instance, a study on allergic and non-allergic hypersensitivity reactions to silicone outlined the capability of silicone to show side effects. Again, silicone might still prove to be harmful according to A. Rubio, C. Ponvert, O. Goulet, P. Scheinmann, and J. De Blic, silicone has been implicated as the causative agent of chronic local or systemic reactions such as inflammatory processes, foreign-body reactions or rheumatologic diseases (A. Rubio, C. Ponvert, O. Goulet, P. Scheinmann, J. De Blic. 2009).

Meanwhile, on the 'Medical News Today' website mentions that silicone is safe for most people as according to MaryAnn De Pietro, "Silicone for the skin may have benefits, such as promoting wound healing, locking in moisture, and smoothing the skin. Silicone is safe for most people" (MaryAnn De Pietro, 2022).

Therefore, given the amount of unconfirmed and contradictory data, this AI Beauty application is crucial to enable cosmetic consumer to make appropriate and prudent choices regarding cosmetic product by their self needs. The application not only helps users distinguish ingredient for a better makeup result, but also allows the individuals to identify the ingredients

that they have to avoid and thus kill two birds with one stone by effectively treating a number of concerns.

1.4 Objective

The objectives of this project are as follows:

- i. To design a user-friendly mobile application interface, ensuring seamless consumer engagement and real-time feedback delivery for accessing cosmetic products.
- ii. To develop an advanced application capable of scanning and analysing the ingredient list of cosmetic products, accurately classifying them as silicone-based or water-based.
- iii. To evaluate the mobile application's performance, effectiveness, and user satisfaction while providing tailored recommendations based on the analysis results.

1.5 Scope

The scope of the application, AIBeautician, is to provide an intelligent platform for the beauty community, enabling users to analyse and evaluate cosmetic products ingredients by distinguishing between formulations specifically silicon-based and water-based through ingredient list analysis.

The application provides real – time results along with recommendations for related cosmetic products based on formulation type. Designed with a user-friendly interface, AI Beautician aims to enhance user experience.

1.6 Brief Methodology

The methodology chosen for this project is Rapid Application Development (RAD). This approach focuses on developing applications quickly through frequent iterations and continuous feedback, which is particularly suitable for a final year project with a limited timeline of approximately one year. The RAD methodology emphasizes customer satisfaction by enabling the early and continuous delivery of functional software. Active customer

involvement throughout the development cycle helps reduce the risk of non-conformance with user requirements, ultimately saving time and enhancing alignment with project objectives (Chien, 2020).

The following figure shows the brief methodology of this project which is segmented into four phases as follows:

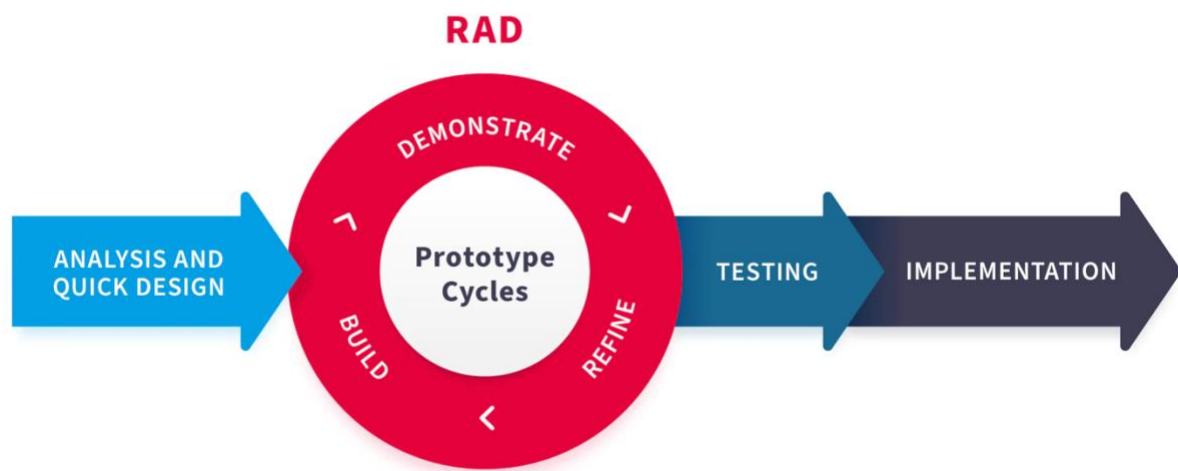


Figure 1.1 Rapid Application Development (RAD) Methodology Cycle

Phase 1: Analysis and Quick Design

During this phase, a detailed project scope is established by thoroughly collecting and evaluating user requirements. The application's core features are identified, including ingredient analysis, product classification, recommendations, and user interface design. Following this, a well-structured system architecture and a user-friendly interface are integrated into the web application design to achieve the project's goals.

Phase 2: Prototype Cycles

Once a project has been scoped, initial models and prototypes are developed. As necessary, this process is repeated, enabling the project to change with frequent feedback. The prototype is continuously improved by user feedback until it satisfies all project requirements, ensuring a functional and user-focused end product.

Phase 3: Testing

After the prototype has been developed into a working model, thorough testing will be conducted to make sure that the final product satisfies the goals and expectations of the project. The AIBeautician system's basic functionalities will be evaluated through functional testing during this phase. Additionally, user acceptance testing will be performed to gather real-world feedback, and observation of user interactions, and results will be analysed to pinpoint areas that require enhancement. Any new needs or enhancements found will be integrated into the system, enabling any necessary modifications and enhancements, followed by iterative testing as required.

Phase 4: Implementation

Finally, after the working model is developed, full-scale testing and user training will be carried out to ensure that all features function as expected and meet the project requirements. Any issues that are found will be fixed, and the system will be improved by making the required changes. At the end of this stage, a fully functional web application that offers an effective foundation for the beauty community will be prepared for deployment.

1.7 Significance of Project

The significance of the AIBeautician project lies in addressing a common challenge faced by consumers in the beauty community on understanding the composition of cosmetic products to select suitable items and achieve better product performance and application result. Many users struggle to differentiate between silicone-based and water-based products due to complex ingredient lists, which often leads to unsuitable product combinations that can compromise makeup performance, resulting in patchiness or uneven application. What's more, these incompatible products ingredient may also be unsuitable for the user's skin condition. Furthermore, the database comparison feature in the system will provide personalised recommendations for comparable product formulas, fostering greater consumer confidence in product selection. By the end of the project, users will be able to use the classified product formulations to make well-informed choices, helping them discover alternatives that align with their preferences and needs.

1.8 Project Schedule

Figure 1.2 below illustrates the project timeline.

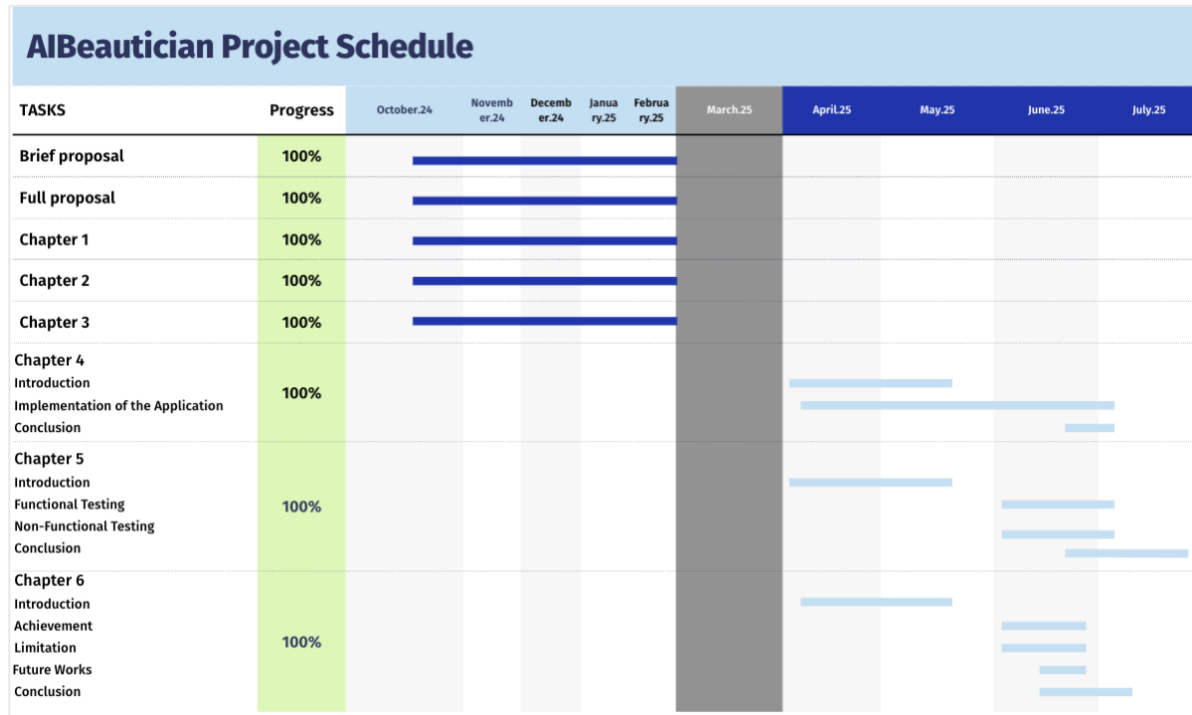


Figure 1.2 Gantt Chart for AI Beautician Application (FYP 1 & FYP 2)

1.9 Expected Outcome

The expected outcomes of this project are:

1. The AIBeautician application will classify beauty products as either silicone-based or water-based by analysing and scanning the product's ingredients.
2. The system will recommend makeup products with similar formulations to users.

1.10 Report Outline

This paper will focus on how the AI Beautician projects offers an innovative and user-friendly solution, enable the user to better understand cosmetic product and make suitable choice. The flow of the paper is described as follows:

Chapter 1: Introduction outlines the background of the project in terms of the definition of each elements from the title “AIBeautician”. This chapter provides a comprehensive overview of the key aspects of the project, including the problem statement, project scope, objectives, brief methodology, the significance of project, the project schedule, expected outcomes, and report outline.

Chapter 2: Literature review describes the relation of the proposed application to other related works. The background research, the comparison of the qualities, strength, limitations along with the suggested improvement to the proposed application will be clearly identify in this chapter.

Chapter 3: Rapid Application Development Methodology explains the steps taken to achieve the end goal of this project in detail. The chapter comprises of several steps and process of the methodology including analysis and quick design, prototype cycles, testing, and implementation.

Chapter 4: The thorough explanation of the implementation of AIBeautician application is discussed in this chapter, following the phases outlined in the methodology described in Chapter 3. The detailed utilization of Google ML Kit for text recognition, along with the approach used to classify the product base type is also thoroughly explained.

Chapter 5: The success of the text recognition, the accuracy of the product base type classification result, and the effectiveness of the recommendations will be evaluated during the testing phase to determine the functionality of each module. The results of this evaluation will be shown in this chapter.

Chapter 6: The conclusion of the project is summarized in this chapter, where the achievement all the objectives is reaffirmed. This chapter also outlines the limitations encountered in the application throughout the development process, as well as proposed future works to address the issues.

1.11 Summary

This chapter lays the foundation for the project by first identifying challenges and showing the relevance of the proposed solution. It also provides the first framework on how the AIBeautician application can contribute to improving user experience and fostering greater transparency in the beauty industry.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this chapter, the related theoretical foundation for the proposed application is covered. The primary objective of this literature review chapter is to review existing application background study related text scanner application within the beauty industry. It compares its functionality, technology features, and user experiences with the proposed application.

2.2 Background Study

In recent years, many companies and product owners have introduced ingredient list scanners to ensure that their products are suitable for individual skin conditions. Nowadays, we see many smart consumers making prudent choice. A lack of knowledge about makeup product ingredients can potentially harm users. Insufficient understanding has frequently made it more challenging consumers to make wise purchasing decisions (P. Elisa, 2019). Technological advances these days are expected to be able to solve problems in various fields (Sitanggang, D., Juan, A., Doa, G. D., & Indra, E., 2024).

However, the proposed application described in Chapter 1 is an AI-powered mobile application called AIBeautician: An AI-Powered Mobile App for Personalized Cosmetic Ingredient Analysis and Recommendations. Artificial intelligence is a method of making a computer, software, or robot think intelligently. It involves creating algorithms to classify, analyse, and make predictions from data (Elder, A., Cappelli, 2024). In this case, AI is used for text recognition to analyse product ingredients through a scanner and make predictions based on data to provide recommendations to the user. Figure 2.1 below shows statistics retrieved from Statista, sourced from Edelman, about the areas where AI is utilized. These statistics compare the awareness of various AI applications between the general population and tech executives.

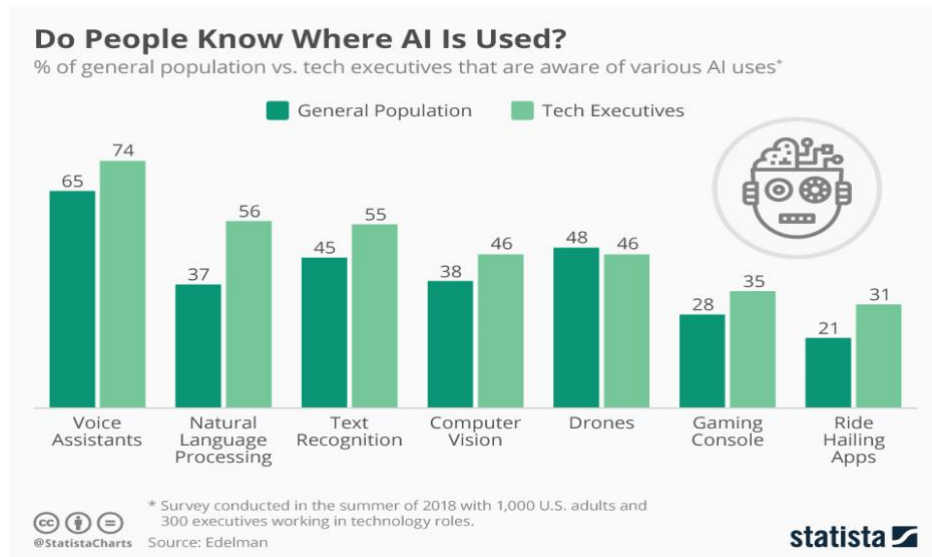


Figure 2.1. Statistic of the general population vs. tech executives that are aware of various AI uses.

In this chapter, three existing application will be analyse. According to Vithani and Kumar (2014), a thorough review and comparison of the features and user base of the existing application and the proposed application is advised if a comparable application is currently available on the market. This could involve researching consumer demand for the existing application, evaluating its features and user experience, and comparing them with the features and potential user experience of the proposed application.

2.3 Background of Existing Application

This section does a background study of three existing applications and compares them in order to develop the proposed application. The functionality, technology features, accuracy, and user experiences of the current application are all evaluated.

2.3.1 INCI Beauty

The INCI Beauty Application was developed and launched at the end of 2017 to allow users to analyse the composition of cosmetic products, as shown in Figure 2.2. The primary objective of the INCI Beauty Application is to provide users with clear and accessible information about cosmetic product ingredients. It aims to enhance consumer awareness,

improve the product selection process, and promote safer choices by highlighting potentially harmful components. This application evaluate the products ingredients based on safety, environmental impact, and health concerns. INCI Beauty categorized the products and rates using predefined scoring system derived from scientific literature and regulatory standards (INCI Beauty, n.d.).

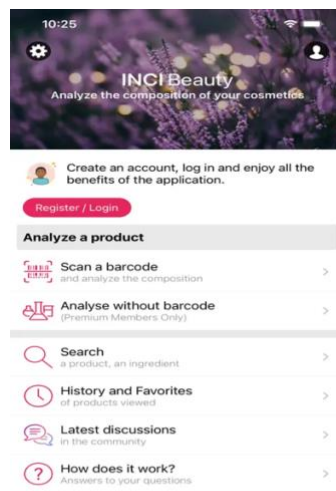


Figure 2.2 Home page of INCI beauty application

INCI Beauty works through barcode or ingredients scans and search for a product or ingredients from the home page. Barcodes as shown in Figure 2.3 are used as identifiers for products which will allow user to recognise the product. Barcodes are a collection of black and white lines of different widths to store special data such as production code, expiration date, identification number, etc. So that the computer system can easily identify the information encoded in the barcode. Barcode has several types of barcodes, namely 1D linear barcode (1 dimension) which is a collection of lines with different thicknesses and is rectangular. And the type of 2D (2-dimensional) matrix barcode is that the data is represented by symbols in the form of squares, dots, hexagons, and other geometric shapes. After the barcode is scanned with a barcode scanner, the code is automatically connected to the product data that has been stored in the database (Sitanggang, D., Juan, A., Doa, G. D., & Indra, E., 2024).



Figure 2.3 Example of barcode

This application uses an algorithm to parse ingredients. Users can use the 'Search for a product or ingredient' menu on the application's home page, as shown in Figure 2.2, to search for a product name or ingredient in the application's database. According to INCI Beauty (n.d.), their application use search engine more than barcode scanner as it relies on the extensive database. From Figure 2.4 below, we can see that 55% of the users tends to used search engine compare to barcode scanner with 45% with 5% comparison to INCI Beauty (n.d.).

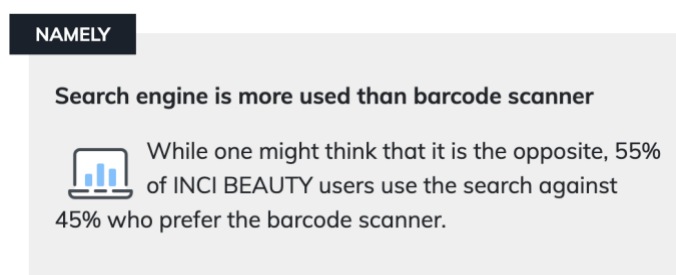


Figure 2.4 Statistic of the user using search engine compared to barcode scanner

As a results, this application provide user ingredient information in a user-friendly interface, showing a detailed description and safety rating are provided for each product. Products are ranked on a scale of 0-20, with a color-coded indicator, as illustrated in Figure 2.5. The ranking is based on elements that could cause an ingredient in a cosmetic product to pose varying levels of risk, as they represent potential danger. The color-coded indicator uses a 'colour flower' system to reveal penalties of varying significance assigned to each ingredient, which are factored into the product score. Each ingredient is identified by a specific colour: green, yellow, orange, or red. However, an ingredient can have different colour flower, and therefore different penalties related to the risk it may represent for the consumer. The ingredient can switch from one colour to another based on updated knowledge that the application may have. The principle being that the more the colour tends towards red, the more controversial the ingredient is and conversely, the more it tends towards, the less so it is. This application provides with the alternatives with better composition based on the category with the better

rating, the point of sale which is the price level of the product, and similar compositions INCI Beauty (n.d.).

Based on the risk element ingredient in cosmetic product, here are 6 categories that this application focus on, namely:

1. The product category: Depending on the type of product in which it is used
2. Rinse-off or non-rinse: Nature of the product
3. Texture of the product
4. Particle size:
5. Destination of the product: The target audience for a particular product
6. Certification: Information on the content and origin of certain ingredients

The coloured flower system:

1. Green: a risk-free ingredient - No penalty
2. Yellow: a regulated ingredient / rather irritating / allergenic - Low penalty
3. Orange: an ingredient derived from petrochemicals and/or which does not deserve a red - Average penalty
4. Red: A controversial or potentially risky ingredient - Heavy penalty

	<i>Green: a risk-free ingredient - No penalty</i> This type of ingredient does not pose any problem for human health and its impact on the environment is less. Most natural extracts are green if they are not regulated.
	<i>Yellow: a regulated ingredient / rather irritating / allergenic - Low penalty</i> This type of ingredient does not pose much more of a problem for most of us than the previous ones. However, in this group we are starting to find some ingredients with irritating or even allergenic power. Others can also be regulated (Annex III of the European regulations): in general, these are concentration limits that are imposed by Europe. We indicate this type of ingredient with a "small" penalty in the products.
	<i>Orange: an ingredient derived from petrochemicals and/or which does not deserve a red - Average penalty</i> Under this color, we mainly reference synthetic, petrochemical or environmentally unfriendly ingredients given their polluting manufacturing process and/or their low biodegradability. Ingredients of this type can have an indirect effect on human health.
	<i>Red: A controversial or potentially risky ingredient - Heavy penalty</i> This is the highest level of penalty for an ingredient. As soon as a sufficiently serious controversy exists around an ingredient, it is positioned in this color level. However, before placing it in such a zone, we make sure that it is possible to do otherwise.

Figure 2.5 The coloured system used by the INCI beauty application

2.3.2 Safely: Cosmetic Ingredients

Safely: Cosmetic Ingredients is an application designed to enhance consumer knowledge about the safety of cosmetic products. The main purpose of this platform is to provide users with immediate insights into the ingredients of cosmetic products, helping them make informed decisions. This application identifies substances that are not recommended for pregnant women, breastfeeding mothers, vegans, allergy sufferers, or those that are generally considered harmful to health (appasion, 2024). There are several ingredients that can be harmful to certain individuals, including parabens, retinoids, and salicylic acid, as these ingredients can potentially be absorbed into breast milk and may harm the baby. According to Marie, C., Garlantézec, R., Béranger, R., & Ficheux, A. (2022), several studies have examined the association between occupational exposure to these products and reproductive and developmental abnormalities.

This application used built-in camera in the device for ingredient scanner analysis the listed ingredients on a cosmetic's label example as shown in Figure 2.6 below where the application will analyse and user will received comprehensive breakdown of any detrimental ingredients, including specific warning about their use (appasion, 2024).



Figure 2.6 The application built-in camera and scanner

Moreover, in Figure 2.7 below, this system application utilizes fixed algorithms and a comprehensive database to evaluate the safety of cosmetic ingredients. It will scanned and analyse products with less safe ingredient then provides the user with field where it should be concerns (appasion, 2024).

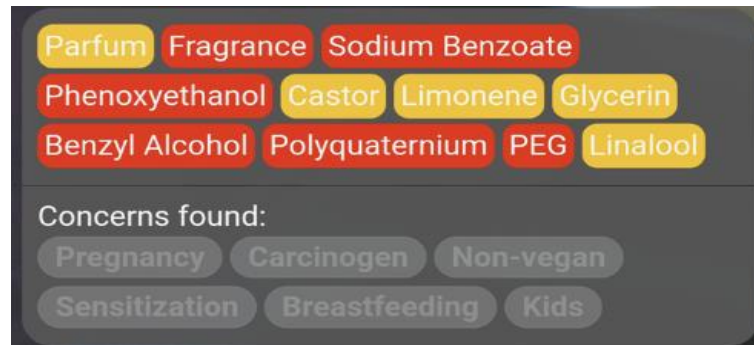


Figure 2.7 Example of harmful substances and where the concerns found results

Based on the application, these are the lists of example ingredient that might contain harmful substances:

1. Fragrance
2. Styrene
3. Phenoxyethanol
4. Sodium Benzoate
5. PEG
6. Benzyl Alcohol

Area where the concern is found:

1. Allergic
2. Sensitive skin
3. Pregnancy
4. Breastfeeding
5. Vegan/ Non-vegan
6. Kids
7. Carcinogenic

2.3.3 Little Bean – Cosmetic Checker

Little bean application was founded by Mirte Maas in 2021. With plenty of cosmetics at home and so much ambiguous data online, she was unsure about which ones were safe to use. It was difficult to identify potentially hazardous ingredients on product labels, even given an extensive list. That's when Little Bean was born, with platform that is for aggregates information on the safety of using certain cosmetic ingredients in one place and that speeds up every day decision-making by allowing users to quickly and effortlessly call up the research relevant to the product of concern. Little Bean also committed to providing reliable information sourced from reputable outlets like the American College of Obstetricians and Gynaecologists (ACOG), European Medicines Agency (EMA), PubMed, and the National Institutes of Health (NIH) says Little Bean (2024).

This application comes with features that allow the user to capture the ingredient list using the device's built-in camera, upload a photo, or paste text from a website into the application. As illustrated in Figures 2.8 and 2.9 below, the app will analyse and evaluate the product's ingredients.



Figure 2.8 The application homepage with three features



Figure 2.9 Built-in camera used in the app

As a result, users will be presented with three categories: high risk, medium risk, and low risk, depending on the harmfulness of the products. Additionally, the app provides detailed research on the ingredients for those seeking more information. Moreover, the app offers alternative or recommended products that are available or have similar formulations.

2.4 Comparison between the existing application

In this section, the similarities and differences of the applications are identified. In the following table shows the comparison between the three existing application that has been discussed in the previous section.

Table 2.4.1 Comparison Table Between Existing Applications

Application Features	Existing Application		
	INCI Beauty Application	Safely: Cosmetic Ingredients	Little Bean – Cosmetic Checker
Scanning Method	Available for premium member only	√	√
Barcode Scanner	√	x	x
Accessibility: Free / Premium Member	• Free features • Premium features	x	x
Recommendation	√	x	√
Platform Availability: Android/ iOS	Both	Android only	Both

From the comparison table of the three existing applications above, we can conclude that these applications share the same main objective: to analyse and evaluate the ingredients of cosmetic products, providing consumers with better-informed choices.

Based on the comparison table, all three applications share a similarity in using a scanner feature to analyse product ingredient lists. However, the INCI Beauty app requires users to upgrade to a premium membership to access this feature. In the free version, INCI Beauty only offers barcode scanning and search functionality, which can be inconvenient as barcode scanning becomes ineffective when certain brands update their products, rendering the old barcodes irrelevant. According to INCI Beauty it happens more frequently that the brand changes the formula of a product, without changing its barcode, which generates a certain confusion on the market. Barcode technology is prone to human error besides it is easy to ripped or damaged there is no way to scan the library materials (Chanda, Dr. A., 2019).

In terms of offering suggestion, two out of three existing applications, INCI Beauty and Little Bean – Cosmetic Checker, are capable of offering an alternative or recommending similar products with the same formulation. On the other hand, the Safety: Cosmetic Ingredient application is not able to provide that feature.

2.5 Limitations Of The Existing Products

As mentioned, INCI Beauty offers the feature to scan the ingredient list of cosmetic products, but users need to upgrade to a premium plan to access this feature. Without it, users are limited to using the barcode scanning option, which is not available for all products, thereby restricting access to information.

2.6 Comparison Between The Proposed Application With The Existing Applications

Table 2.6.1 Comparison Between the Existing Application and Proposed Application

Application		Existing Application			
		INCI Beauty Application	Safely: Cosmetic Ingredients	Little Bean – Cosmetic Checker	Proposed Application: <i>AI Beautician: An AI-Powered Mobile App for Personalized Cosmetic Ingredient Analysis and Recommendations</i>
Features					
Scanning Method		Available for premium member only	√	√	√
Barcode Scanner		√	x	x	x
Accessibility: Free / Premium Member		- Free features - Premium features	Free	Free	Free
Platform Availability: Android/ iOS		Both	Android Only	Both	Both
Recommendation	Safe ingredients	√	x	√	√
	Silicon/Water based product	x	x	x	√
Scanning: Silicon/ Water based product		x	x	x	√

√ : The features provided in the application
x: The features is not provided in the application

The table above shows a comparison between the proposed application and the existing applications. All four applications are capable of scanning and analysing the ingredient lists of cosmetic products. However, the proposed application, *AI Beautician: An AI-Powered Mobile App for Personalized Cosmetic Ingredient Analysis and Recommendations*, focuses on scanning and analysing ingredients to classify them as silicone-based or water-based, a feature not available in any of the existing applications.

Additionally, while two of the existing applications are capable of recommending alternative products that prioritize safe ingredients for health, the proposed application goes a step further by offering alternatives or recommendations specifically based on silicone-based or water-based formulations. These recommendations are tailored to suit the user's skin condition, ensuring greater user satisfaction and helping them make more informed choices.

2.7 Discussion

After reviewing the three existing applications, one feature that can be implemented in the proposed system is the barcode scanning feature. With this feature, consumers can access more information provided by the brand itself, offering greater accuracy regarding the ingredients of their cosmetic products.

2.8 Summary

Conclusively, this chapter has discussed related existing applications and research, further subcategorized into the methodology used, results, implications, and evaluation of the applications. As discussed, different applications have various features and functions to achieve their objectives. However, the purpose of this comparison is to identify the strengths and weaknesses of these applications and to use this information to enhance the design of the proposed application. The comparison highlights the pros and cons of each application, aiming to ensure that the proposed application is more efficient and effective than the existing ones.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter focuses on the development methodology, requirement analysis, and designing of AIBeautician: An AI-Powered Mobile App for Personalized Cosmetic Ingredient Analysis and Recommendations. According to Soobia (2019), “ The Methodology Software Development process is a set of rules that are used to address all of the stages in order to succeed in software development” (p.447). The software development life cycle can be managed in a different way with each methodology, which defines project steps, roles, responsibilities, activities, communication standards, and deliverables. As mentioned in Chapter 1, the proposed system is built using Rapid Application Development (RAD), approach. RAD is an agile software development methodology that places less focus on precise planning and more on rapid feedback and experimentation. In general, the RAD methodology places more emphasis on development and building prototypes than on planning.

3.2 Rapid Application Development

Quick application development and prototyping are the main goals of the RAD methodology in order to produce products faster. Rapid Application Development is a development lifecycle that aims to produce solutions that are both higher quality and significantly faster than those obtained with conventional approaches, where time is a constraint during the building of this project (Despa, M. L., 2014). This method works particularly well when there is a lot of ambiguity in the requirements or when the project has a tight deadline. The iterative nature of RAD allows for changes to be made at any stage of the software development process where the cycles are time boxed and multiple cycles can be developed at the same time (Despa, M. L., 2014).

This method is applied in this project because it works well for software development and small to medium-sized projects. This approach is also effective for small, medium, and large businesses, but it requires that the project be broken into smaller components (Soobia, 2019). Moreover, RAD often employs automated tools, reusable components, and visual development environments to accelerate the development process. Quick evaluations can be

completed in less time and with fewer workforce resources, increasing productivity. Rapid Application Development (RAD) aims to expedite the development process and provide the client with a functional product as quickly as feasible. It is important to divide the project into units, and as shown in Figure 12 below RAD paradigm consist of the phases of analysis and quick design, prototype cycles, testing and implementing.

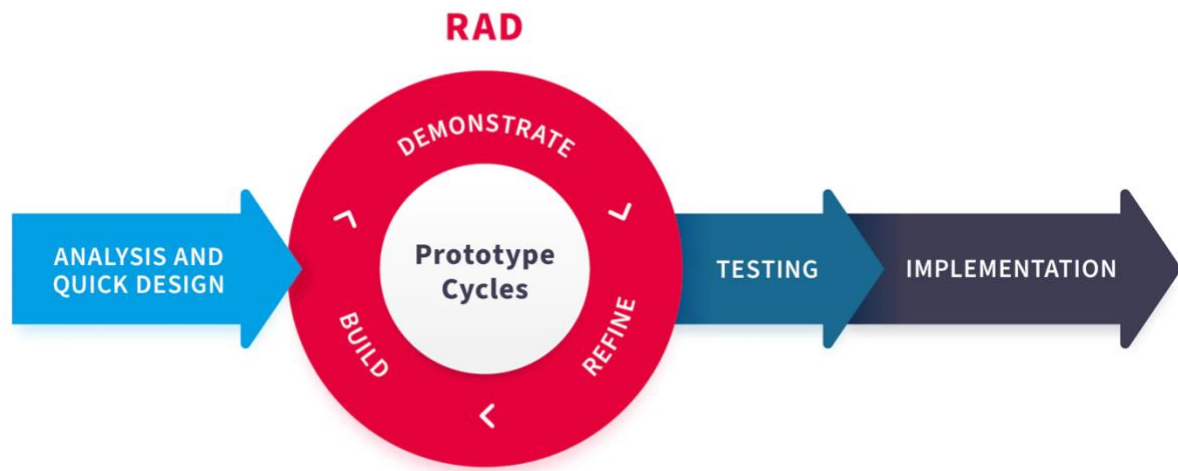


Figure 3.1 Rapid Application Development (RAD) Methodology Cycle

3.2.1 Phase 1: Analysis and Quick Design

The initial phase of the project is outlines the project's scope, objectives, timeline, and gathering information to define and finalise the requirement and expectation. The stage where requirements are gathered through elicitation techniques (e.g. brainstorming, interview, and so forth) (Nurhaida, I., Ayumi, V., 2020). The initial action in the phase is to acquire an overall overview of the proposed application by consulting with the FYP supervisor and gathering information about it.

A thorough research effort was conducted to support the relevance of the project and provide context for the related information. Existing applications were studied to identify their advantages and disadvantages, which were then compared and used as a basis for improvements in the proposed application. At the end of this phase, the student and the FYP supervisor have established and approved the requirements.

3.2.1.1 Questionnaire

Questionnaire is conducted through online as a requirement gathering method used in this project. It is a structured list of questions designed to collect specific information relevant to the project. The questionnaire was distributed online to various types of cosmetic product users. The questions range greatly, from multiple-choice to open-ended, and they all collect important data for well-informed decision-making. They were roughly 23 respondents involved in this survey.

Section 1: Demographics What is your age group?

23 responses

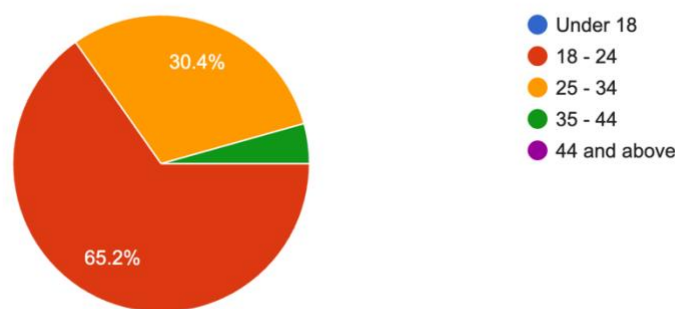


Figure 3.2 Pie chart of distribution of consumers by age group

Figure 3.2 shows the distribution of survey respondents based on age groups. The largest proportion of the cosmetic product consumer that took this survey, 65.2% are between the age of 18 and 24. 30.4% of the respondents are between the age of 25 and 34, while only 4.3% of respondent belong to the 35 - 44 age group. From the data, indicates that younger consumers are the primary drivers of the demand for cosmetic products. This reflect that these age of groups are more likely engage with beauty products due to trends and evolving beauty consumption habits.

What is your gender?

23 responses

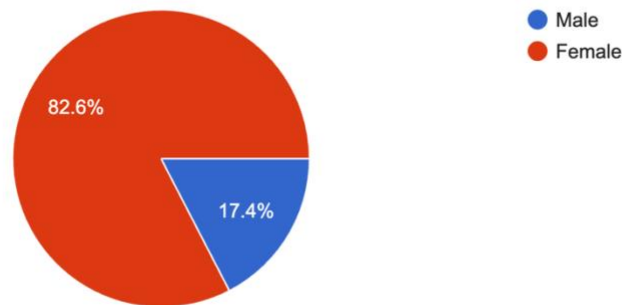


Figure 3.3 Pie chart of the distribution of consumers by gender

Figure 3.3 shows the distribution of survey respondents based on gender. With percentage of 82.6% of respondents are female compared to 17.4% are male. This demonstrate that larger female demographic indicates a higher level of participation in beauty-related surveys as well as a general interest in cosmetic compared to male.

What is your current role or occupation?

23 responses

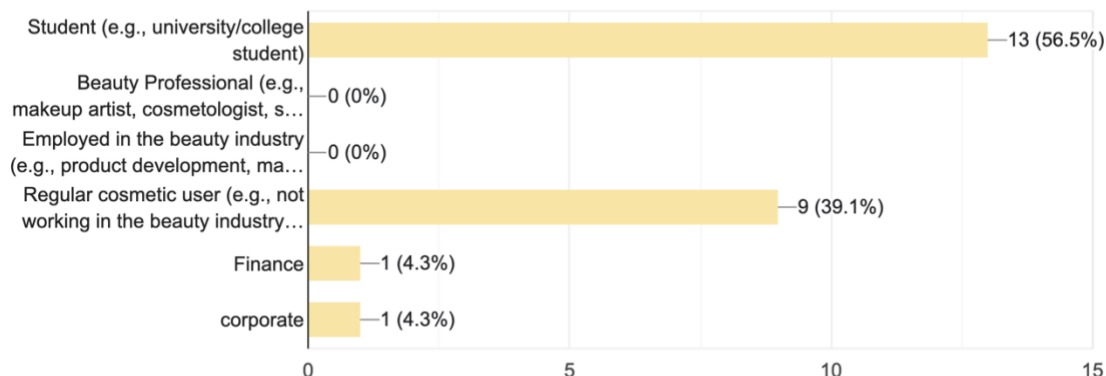


Figure 3.4 Distribution of respondents by role or occupation

Figure 3.4 shows the distribution of survey respondents based on their role or occupation. The majority of the respondents are among students with 56.5%, followed by 39.1% who are regular cosmetic user that are not working in the beauty industry, and remaining with 8.6% are corporate users engaged with cosmetic.

How often do you use cosmetic products? (e.g., makeup,skincare)

23 responses

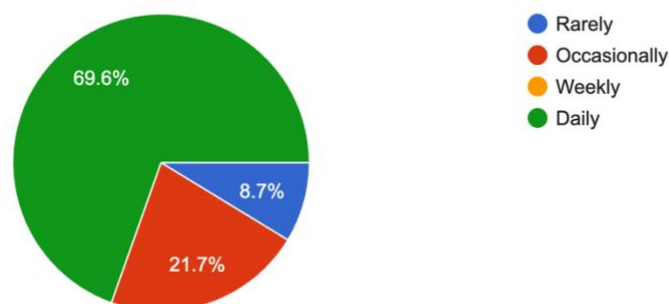


Figure 3.5 Frequency of cosmetic product use among respondents

Figure 3.5 shows the frequency of cosmetic products use among respondents. 69.6% respondents use it as part of their daily routine, 21.7% use them occasionally user and 8.7% respondents rarely use cosmetic product.

How do you typically choose cosmetic products? (Select all that apply)

23 responses

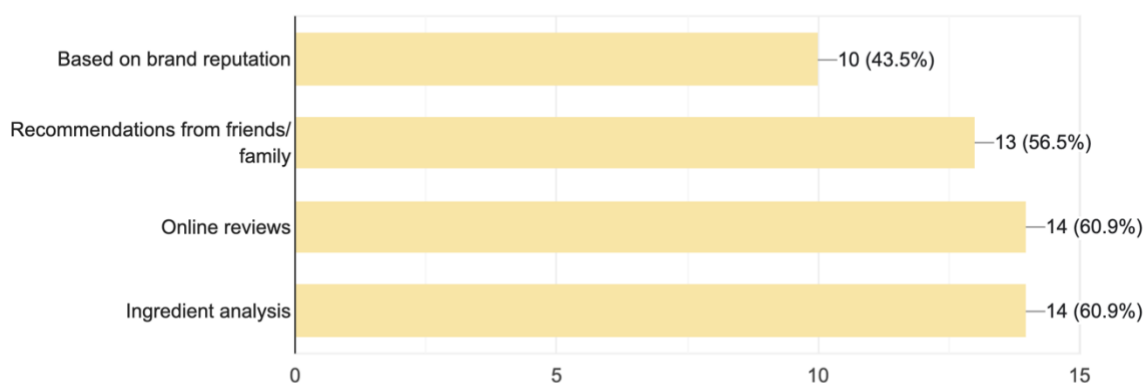


Figure 3.6 Criteria for choosing cosmetic products among respondents

Figure 3.6 shows criteria for choosing cosmetic products among respondents. A majority, with 60.9% of respondents indicated that online reviews and ingredients analysis are the primary criteria most commonly use when purchasing cosmetic product. 56.5% of respondents reported relying on the recommendation from friends and family, while 43.5% respondents are based on the brand reputation. The data indicates that the majority of the respondents are relying on the ingredients analysis in their decision-making process.

Have you ever experienced patchy or uneven makeup due to product incompatibility (e.g., mixing silicone-based and water-based products)?

23 responses

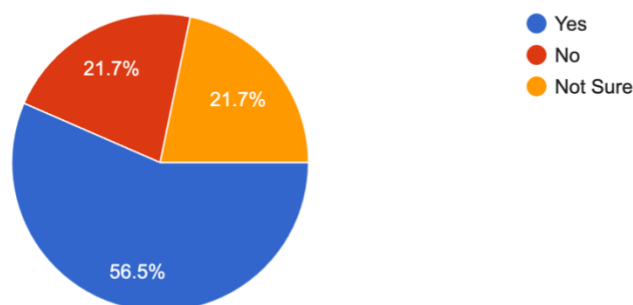


Figure 3.7 Pie chart of respondent's experiences with patchy or uneven makeup

The Figure 3.7 pie chart above depicts the percentage of respondents who have had patchy or uneven makeup due to product incompatibility, such as blending silicone-based and water-based products. A total of 56.5% of respondents stated having experienced patchy or uneven makeup, while another 21.7% stated had never encountered this issue. Another 21.7% were uncertainty about whether they had experienced similar issue.

Are you aware of the differences between silicone-based and water-based cosmetics?

23 responses

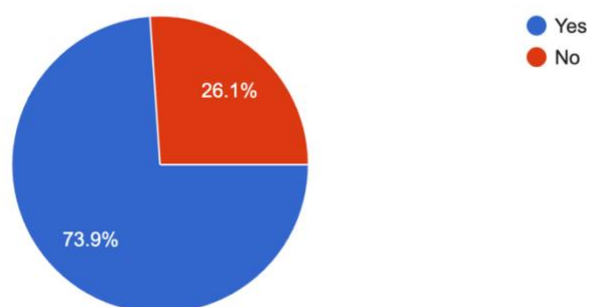


Figure 3.8 Awareness of differences between silicone-based and water-based cosmetic

Figure 3.8 illustrates respondents' awareness of the differences between silicone-based and water-based cosmetics. 73.9% of respondents are aware, while 26.1% are not aware of the differences between silicone-based and water-based products.

Do you find it hard to analyse the difference between silicone-based and water-based products?

23 responses

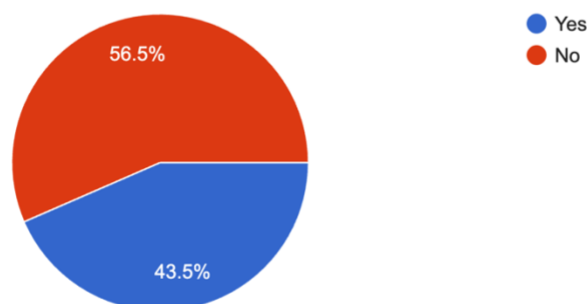


Figure 3.9 Difficulty in analysing the differences between cosmetic product based

Figure 3.9 depicts respondent's views on the difficulty in distinguishing between these two cosmetic product based. According to Figure 3.7, most of the respondents aware of the differences, but 43.5% find it difficult to analyse the products bases, meanwhile 56.5% stated that it is not hard to distinguish between these two product based. This data indicates that, there are still cosmetic products consumer out there that are still struggling to differentiate between these two bases, to ensure the ingredients are suitable to certain individual.

Do you currently check ingredient lists when purchasing cosmetic products?

23 responses

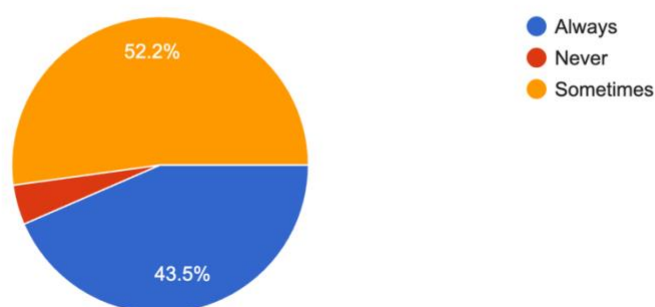


Figure 3.10 Respondent's habits of checking ingredient lists when purchasing

The figure 3.10 above illustrates the percentage of respondents who analyse ingredient lists while purchasing cosmetic items. A majority of 52.2% of respondents reported analysing ingredient lists occasionally, while a number of 43.5% stated that they do so every time they make a purchase, and 4.3% of respondents had never analysed ingredient lists before

purchasing. The above data shows that, some user prefer to thoroughly analyse the ingredient lists before making inform purchasing decision.

Have you ever used a mobile app to scan or analyze your cosmetic products? If yes, which one?
23 responses

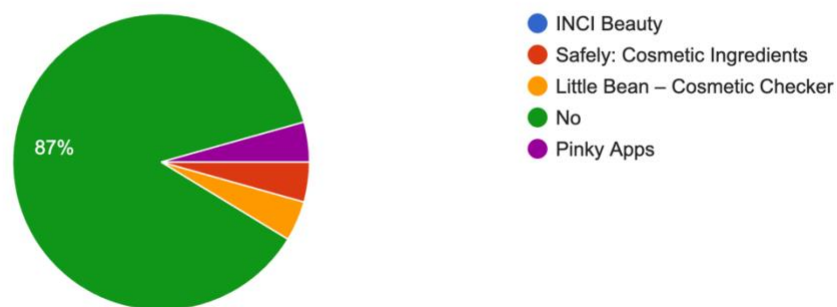


Figure 3.11 Usage of mobile application for scanning and analysing cosmetic product among respondents

The Figure 3.11 above illustrates the respondent's usage of mobile apps to scan or analyse cosmetic products. A majority of 87% of respondents never use any mobile application to scan or analyse the cosmetic product. In contrast, 4.3% of respondents respectively indicated that they have used application such as Pinky Apps, Safely: Cosmetic Ingredients and Little Bean – Cosmetic Checker.

Which features would you find most useful in a mobile app for cosmetic analysis? (Select all that apply)
23 responses

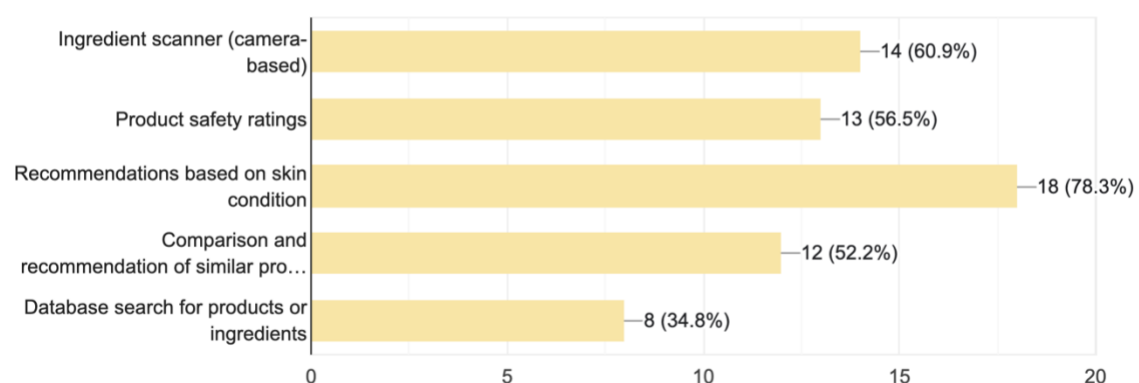


Figure 3.12 Most useful features in a mobile application for cosmetic analysis

As for Figure 3.12 the bar chart depicts that the most preferred features in mobile application are the recommendation feature together with scanner feature, with 78.3% and 60.9% of respondents selecting these options, respectively. The least preferred feature is the database search for products or ingredients, with only 34.8% of respondents indicating interest. This data point out that user are more inclined towards real-time analysis and consider database search to be time-consuming.

Would you prefer a simple and straightforward app or a feature-rich application?

23 responses

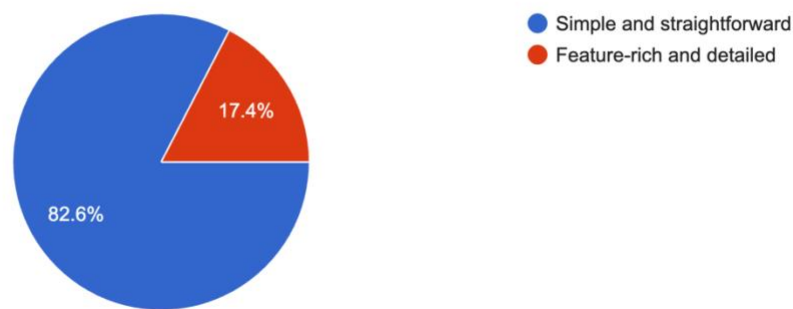


Figure 3.13 Preference for a simple vs. featured rich cosmetic application

Figure 3.13 above illustrates the percentage of the respondent's preference for the features of mobile application. 82.6% of respondents prefer for a simpler feature over feature rich with 17.4% only. This data shows given that many cosmetic product consumers have never used any mobile application for scanning and analysing ingredient lists before.

What are your main frustrations with current cosmetic apps or tools? (Select all that apply)

23 responses

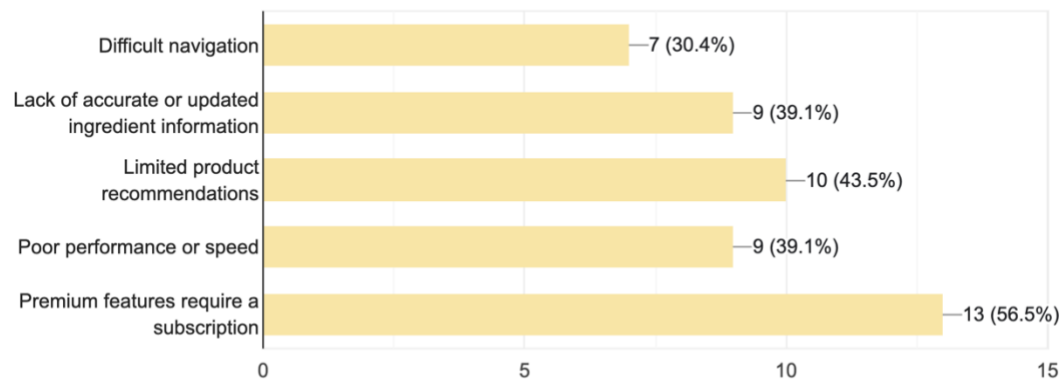


Figure 3.14 Main frustration with the current checker applications or tools

The chart 3.14 above depicts respondents' main complications with the use of current cosmetic applications or tools. Premium features that require a subscription are the main concerns of 56.5% of the respondents with the current checker application. Following this, 43.5% of respondents stated that limitation of product recommendations is an issue, while 39.1% of respondents mentioned that lack of accurate recommendation and poor performance are encountered with the current application. Lastly, 30.4% had challenges with the difficult navigation. This data indicates that most users choose free subscriptions, as shown in Figure 3.4 above, where students comprise the largest proportion of cosmetic product consumers or users, likely due to their age.

3.2.1.2 Software Requirement

Table 3.2.1 Software Requirement

Software tools	Description
Google ML kit	Google ML Kit is a software development kit that enables on-device machine learning capabilities in Android and iOS applications. It includes a selection of pre-built APIs for tasks including text recognition, facial identification, barcode scanning, and

	language translation. By leveraging Google's powerful machine learning models, ML Kit enables the seamless integration of AI-driven features into mobile applications without requiring extensive machine learning skills or relying on cloud-based resources. One of its important features is real-time text recognition, which is designed to perform well across a wide range of devices (Google., (2024).
Android Studio	Android Studio is the official Integrated Development Environment (IDE) for application by Google. It is mainly used for developing Android application. For this project, Android Studio is also support Flutter based IOS application development that enable codebase for both Android and iOS platforms.
SQLite	SQLite is a C-language library that implements a SQL database engine that is small, quick, self-contained, highly reliable, and fully featured (SQLite., <i>n.d.</i>). It works directly on local storage, is fully ACID compliant, and requires no further setup or administration. Widely used in mobile and IoT devices.

3.2.1.3 Hardware Requirement

This project requires the utilisation of two important hardware tools. A laptop will be utilised as the primary development tool for the mobile application, with mobile devices used for testing, debugging, and evaluating its functionality.

Table 3.2.2 Hardware Requirement Specification

Hardware tools	Description
Laptop	MacBook Air Apple chip 1 with 8-core CPU and 7 or 8 GPU suitable for application development.
Mobile devices	iPhone 15 plus A16 Bionic (4 nm), Hexa-core CPU (2×3.46 GHz + 4×2.02 GHz), and 5-core GPU. This device suitable for application development where it ensure smooth simulations performance.

3.2.2 Phase 2: Prototype Cycles

This stage is designation of the interfaces and creates system architecture. During this phase, an iterative cycle consisting of prototype, test and refinement phases. A prototype is an initial model of a product constructed to test a design. Design and engineering professionals frequently use prototypes to refine products and procedures before putting them into production on a big scale. According to Hurwitz (2014), a prototype is a functioning prototype of a product that replicates the visual and tactile characteristics of the final product. This step is often repeated as many times as necessary as the project evolves over time. Rapid prototype development promotes user input, testing, and feedback on an actual system until project requirements are sufficiently fulfilled. The rapid nature of releases means errors are far more likely to be discovered earlier, which reduces errors and debugging (Chien, C.,2020). System's flow and activity are determined and constructure in this phase.

3.2.2.1 Flowchart

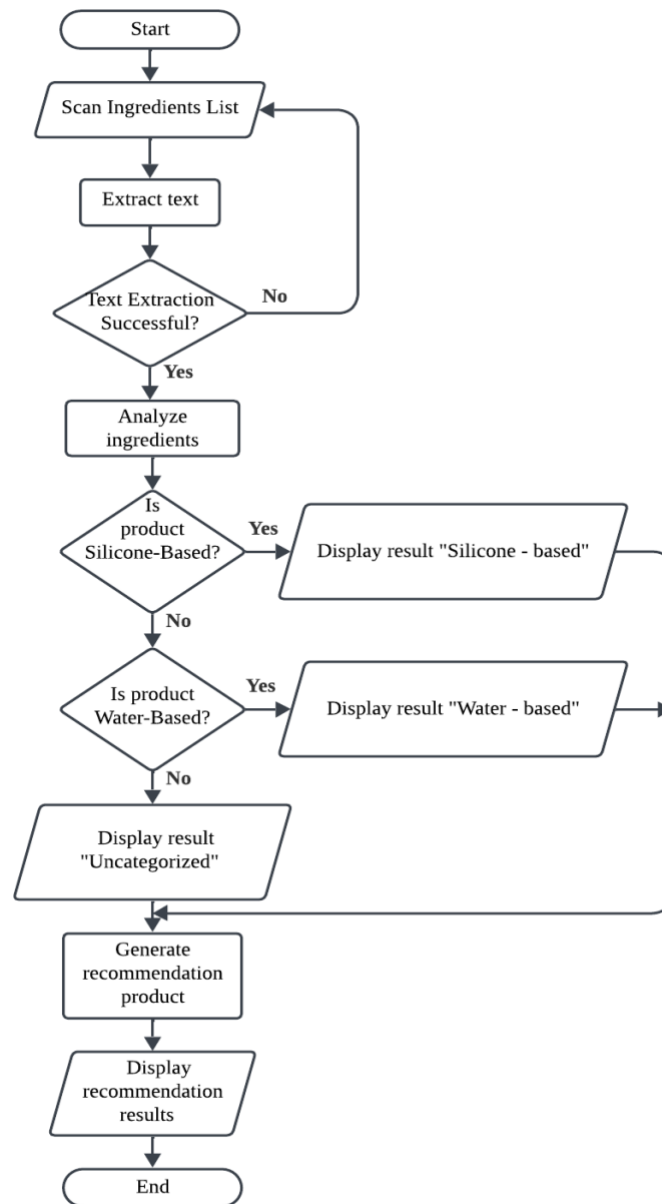


Figure 3.15 Flowchart of the proposed AIBeautician application

As shown in Figure 3.15 illustrates the workflow of the proposed AIBeautician application which begins with scanning the text of the ingredient lists using the established Google ML Kit. This stage utilises Google ML Kit's text recognition capabilities to extract text from the scanned image in real - time. If the text extraction fails, the system prompts the user to rescan the ingredient list to ensure that the information provided is accurate. Once the text has been successfully extracted, the system will proceed to analyse the ingredients.

The analysis is performed by referencing the data stored in SQLite. The system then checked whether the products is silicone – based, water – based, or provides an output of ‘Uncategorised’ for products that do not fit into either category. Following categorisation, the algorithm offers personalised product recommendations. This stage involves implementing machine learning techniques in conjunction with ingredient data stored in SQLite to identify and recommend items with comparable formulations. These recommendations are subsequently displayed to the user, enabling them to make informed purchasing decisions and enhance their overall user experience. The process concludes with the user having the option to analyse another product or exit the application altogether.

3.2.2.2 Activity Diagram

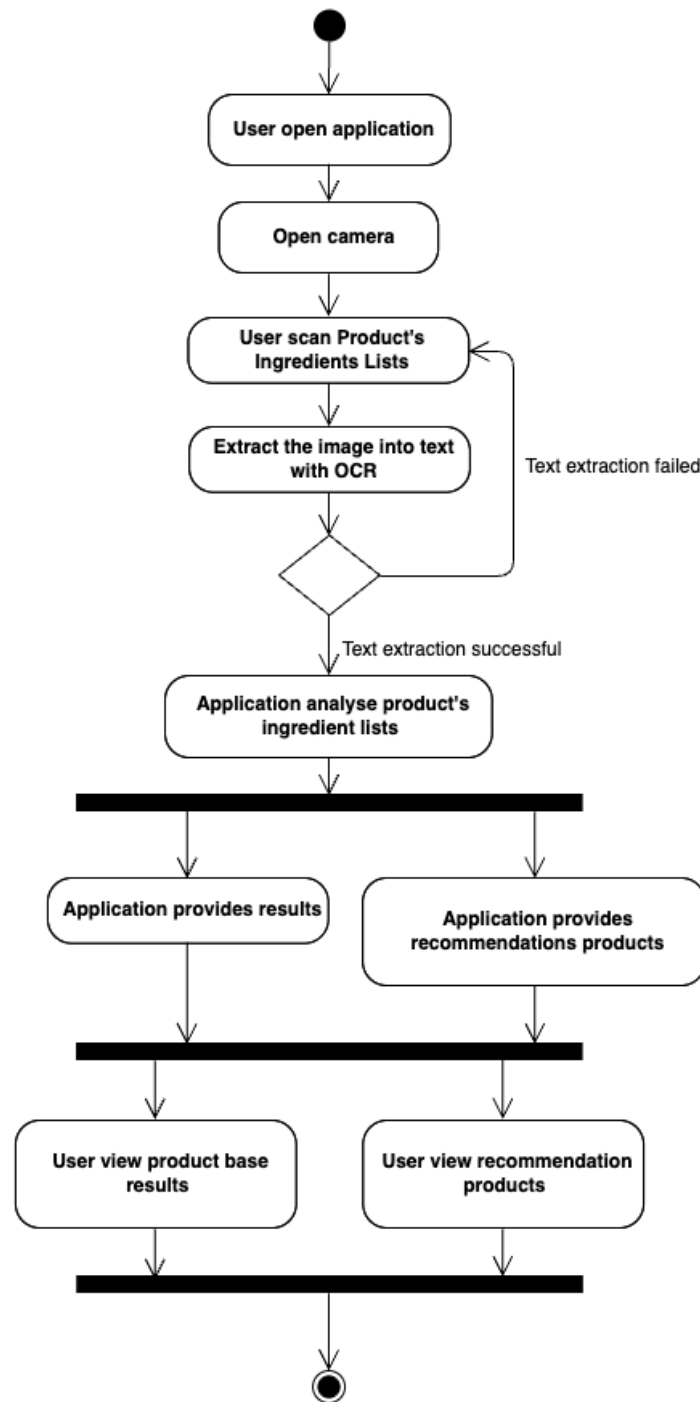


Figure 3.16 Activity Diagram of workflows within the application

Figure 3.16 shows the activity diagram of the proposed application. The activity diagram emphasizes the first objectives of the projects, which are to design a user – friendly mobile application with a simpler user interface.

3.2.2.3 Sequence Diagram

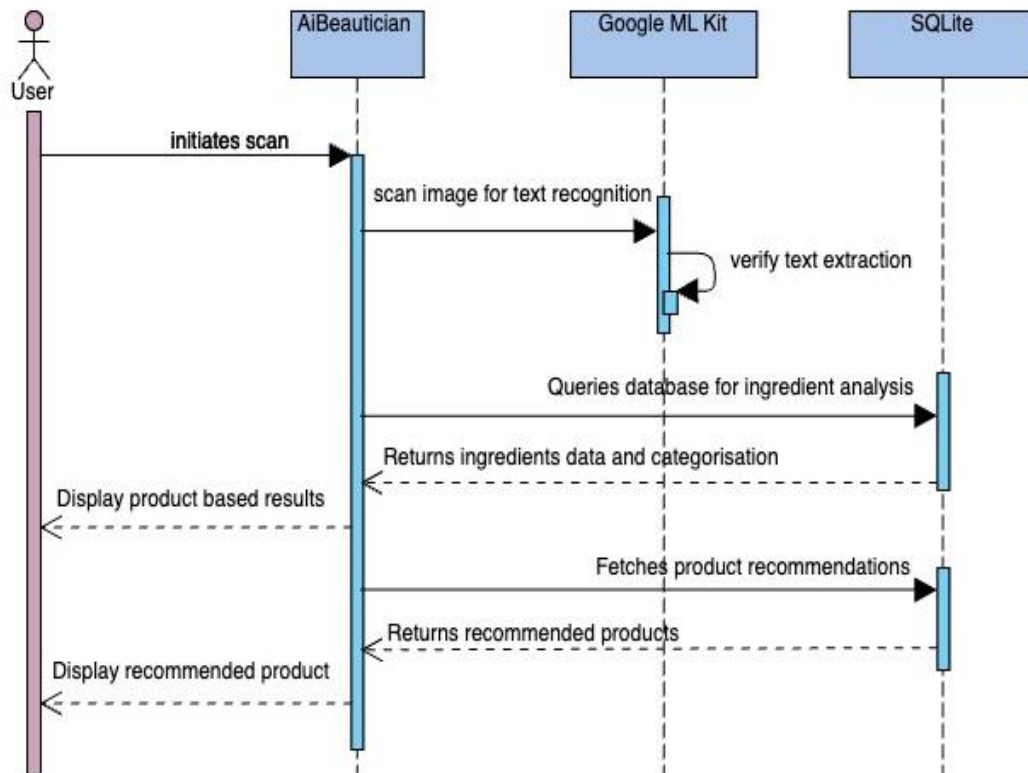


Figure 3.17. Sequence Diagram of the proposed application

Figure 3.17 shows the sequence diagram of the proposed application. Above sequence diagram illustrates the workflow process of the application as it interacts within a system over time.

3.2.2.4 Class Diagram

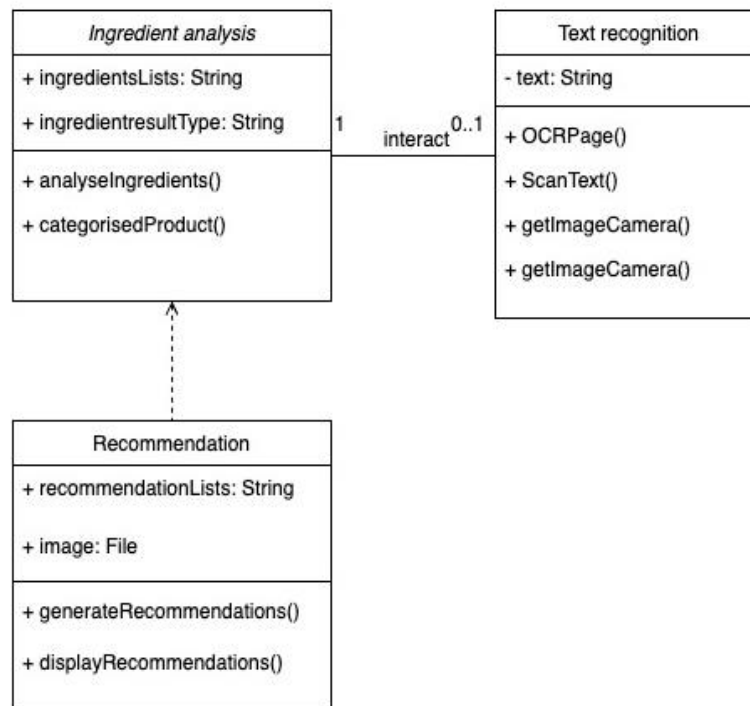


Figure 3.18 Class Diagram of the proposed application

Figure 3.18 above shows the class diagram of the proposed application. Class diagram is used to construct and visualize object oriented system of the project. As shown in the figure, the relation between the classes are evident. Text recognition class feed extracted text to the ingredients analysis class. Recommendation class relies on the results of the ingredient analysis to generate tailored recommendations for the user.

3.2.2.5 Use Case Diagram

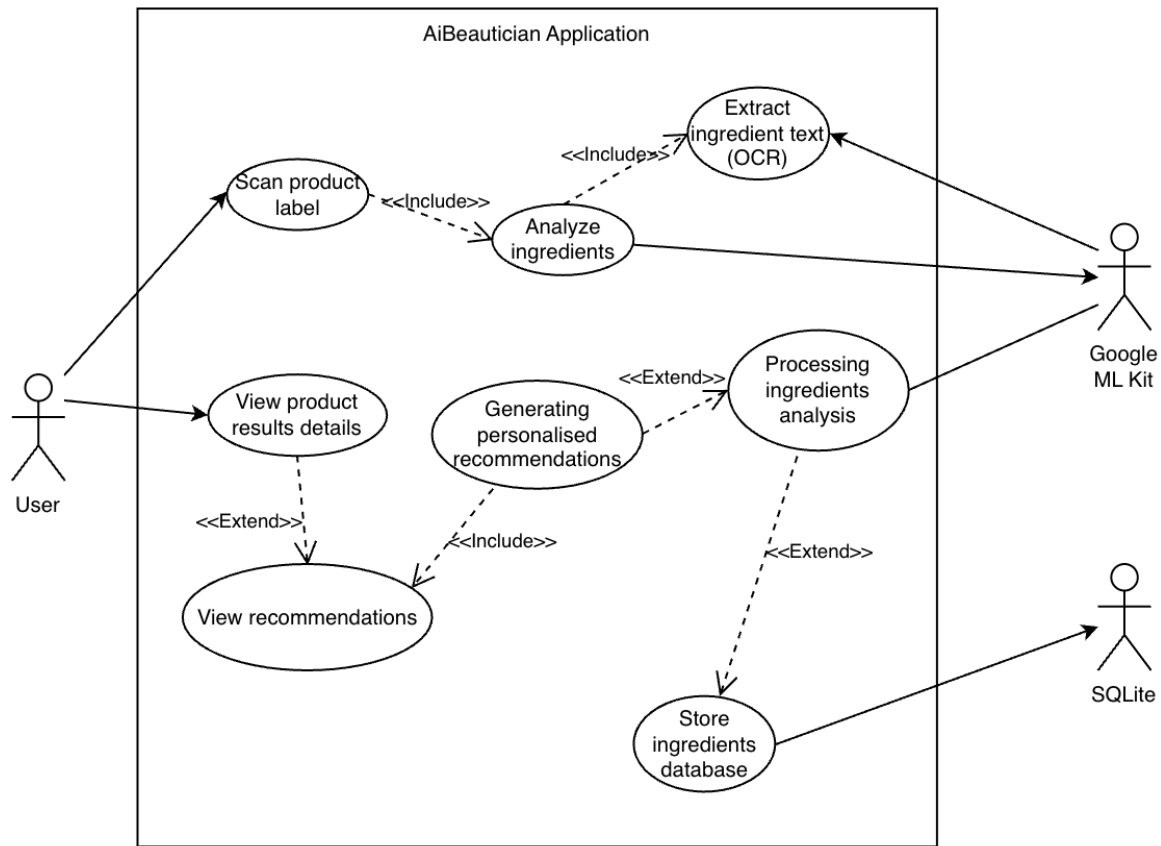


Figure 3.19 Use Case Diagram of the proposed application

Figure 3.19 above illustrates the use case diagram of the proposed application. Use case diagram is used to show the interactions between the application system and its actors. This project involves three actors which are the User, who interact with the application, Google ML Kit, and SQLite. The primary use cases for the User include scan product label, view product result detail and view recommendations. Google ML Kit is responsible for extracting ingredient text (OCR), and analyse ingredient text. Lastly, SQLite is utilized for storing ingredient databases of the products and preloaded product information for the recommendation engine. The entry point for user interaction is the user initiates the scan product label by capturing the ingredients list or uploading an image from the device gallery. This use case has an include relationship with analyse ingredient and OCR use cases indicating that in order the engine analyse the ingredient, user must provide the image. The user is able to view the base type classification result and first six main ingredients after the analysing process done by Google ML Kit. User must be able to view the recommended product that have been generated for

personalised recommendation. These products are fetch from the internal database stored using SQLite.

3.2.2.6 Proposed User Interface

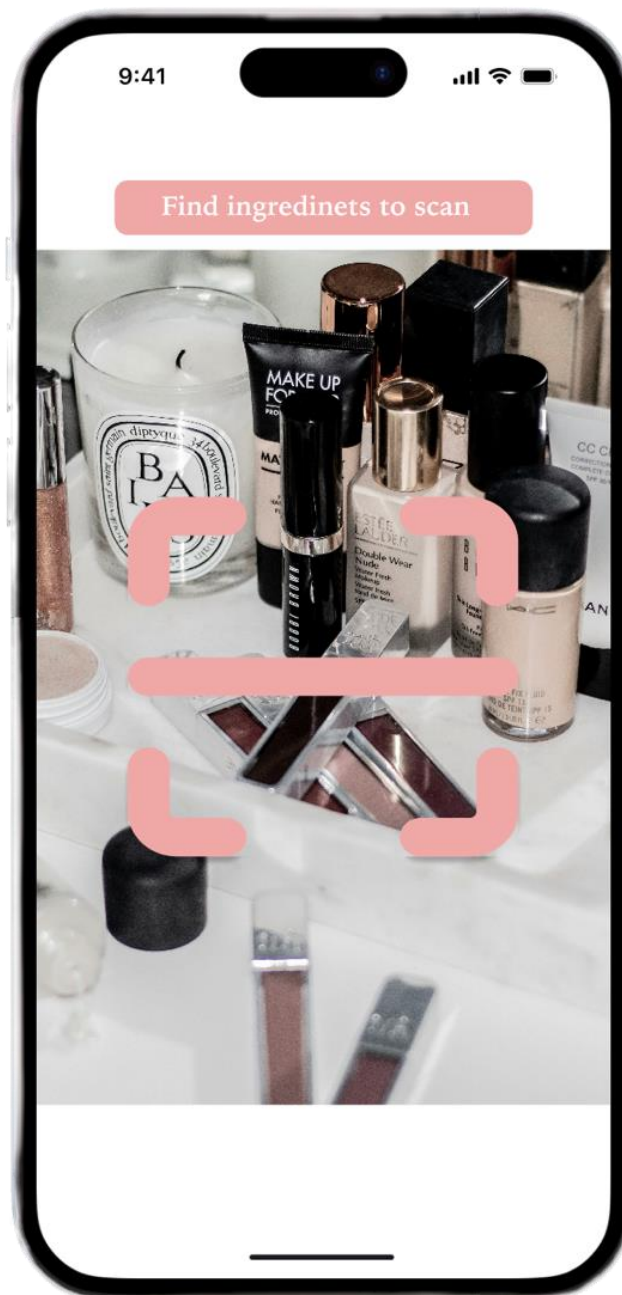


Figure 3.20 Homepage of the application

Figure 3.20 above illustrate the scanning feature interface of the proposed application once they click the application in mobile device. This feature enables users to scan the ingredient lists of cosmetic products directly using their mobile devices. The design adopts a minimalist and intuitive approach, ensuring accessibility and ease of use, particularly for users who may have limited experience with similar application.

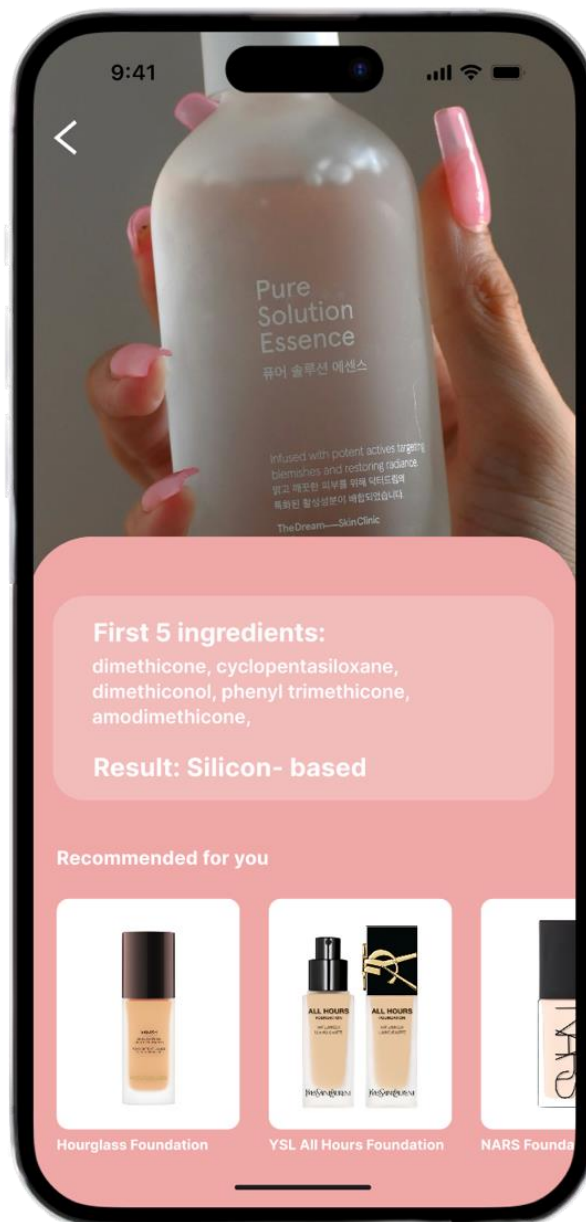


Figure 3.21 Result and product recommendation interface

Figure 3.21 illustrates the results and recommendations of the analysed product in the proposed AIBeautician application. From the figure above, the screen will display the first five ingredients and the result of the product base. The screen also shows the recommendation product allowing user to compare and choose the from recommended product options.

3.2.3 Phase 3: Construction and Testing

During this stage, the construction steps and core functionalities of the mobile application are carried out enabling changes and improvements in order to testing outcomes or newly discovered requirements. This stage requires a lot of code programming for integrating the machine learning models with the front-end and back-end parts of the mobile application. Implementing the machine learning models that drive the app's AI-driven analytical engine and recommendation system is one of the most important tasks in this phase. The system is trained on data related to cosmetic ingredients and formulations, enabling it to efficiently analyse product's ingredients. As a result of this training process, the AI is able to distinguish between different formulation types, such as water-based and silicon-based products, and provides users with individual recommendations according to their needs and preferences. By the end of this phase, a functional prototype of the AIBeautician application is ready for further testing and refinement.

3.2.4 Phase 4: Implementation

The final phase of the RAD lifecycle is implementation, which involves putting the designed solution into production. This phase includes extensive testing, final changes, and the official launch of the mobile application. Testing is carried out to guarantee that the application fulfils all functional and non-functional requirements. Any errors or anomalies identified during testing are addressed immediately to ensure the application functions smoothly. Once testing completes and the app passes all of the requirements, the AIBeautician application is launched.

3.3 Summary

To conclude, the development of AIBeautician application is developed using Rapid Application Development (RAD) methodology, which approach that is suitable for faster development cycles that flexible with accommodating changes of requirement simultaneously. All relevant requirement are gathered during the early stage of the development to ensure that the application can be constructed completely. A questionnaire was published to gather information and preferences from respondents, ensuring the an application to be developed based on the preferences of the majority. This methodology offers continuous user feedbacks, testing throughout the process ensures that the final product meets user expectations.

In conclusion, this chapter discusses the step-by-step approach of the Rapid Application Development methodology based on the literature review in Chapter 2.

CHAPTER 4: SYSTEM IMPLEMENTATION

4.1 Introduction

This chapter focuses on the implementation of the AIBeautician application. For the implementation phase, it focuses more on the how the proposed application is developed as outlined in the previous chapter's software requirement. This is to ensure that the proposed application is intended to fully accomplish its objectives.

4.2 Implementation of the Application

The software implementation is a process where a system operating in its environment which includes the installation and configuration. Based on the Rapid Application Development (RAD) methodology, this is the construction phase where by the implementation method that this process is repeated, enabling the project to change with frequent feedback as mentioned in Chapter 3. The details, and structured that are required to be installed for this project are further explained with screen captures along with a brief description of the application.

For the testing phase which been repeated during this phase is to determine whether the function of the application has meet the project's aims and objectives. It is a crucial part of the cycle for determining the functionality and effectiveness of the application. According to Arkin, B. (2005), software penetration testing is an integral part of the quality assurance and testing organizations in a software application's effort to meet its functional business requirements. White-Box testing and Black-Box testing are the two ways of procedures. However, Black-Box Testing is more concentrated on in this project.

To develop the proposed AIBeautician mobile application, the Flutter cross-platform UI toolkit was used to build the UI. Android Studio is the official Integrated Development Environment (IDE) that created by Google and Xcode were employed for coding, building, and running the application on the device. CocoaPods, a dependency manager also used for IOS to manage native plugins. A detailed overview of the other additional software tools used for mobile application development is provided and explained below, supported by images as evidence.

4.2.1 Software Tools Setup

4.2.1.1 Flutter

Flutter is an open source user interface (UI) software development toolkit by Google, designed to support efficient performance for application development. It was used as development framework in this project due to its capability to streamline the development of a user-friendly mobile application for both iOS and Android platforms. The framework support integration with platform-specific features through platform channel, hence ensuring access to native APIs and functionality. Furthermore, Flutter's Hot Reload feature allows to view change instantly without restarting the application, amply fast development and facilitating rapid implementation, which was suitable for the project required.

1. Flutter Framework Installation

To download the Flutter framework, browse to <https://docs.flutter.dev/get-started/install> to get it from official website. As shown in Figure 4.1 and Figure 4.2 below, Flutter framework is available for multiple platforms, including Windows, macOS, Linux and ChromeOS. This cross-platform, make it versatile to use for developing an application from various operating system. The Flutter SDK of 'flutter_macos_arm64_3.32.4-stable.zip' is used for M1 chip model of MacBook.

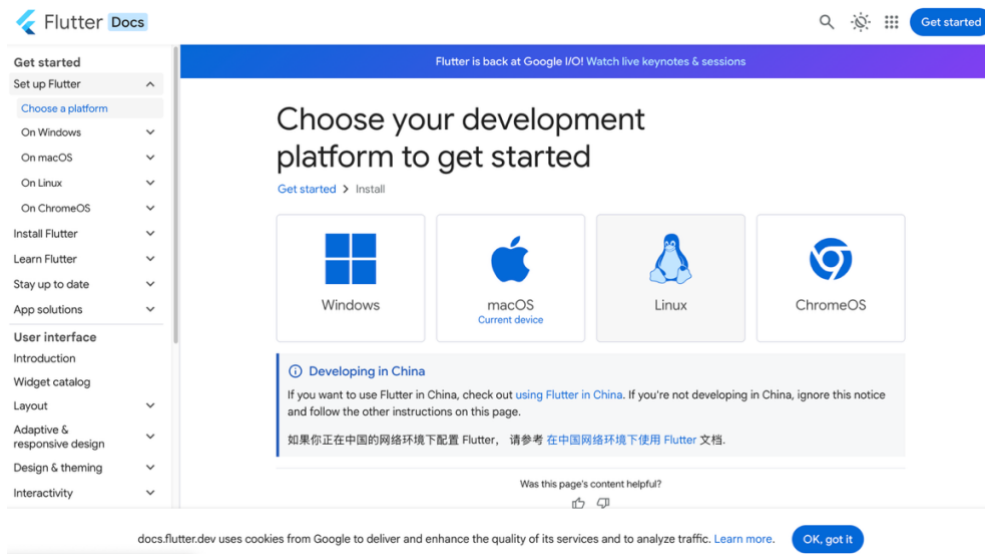


Figure 4.1 Flutter Download Page

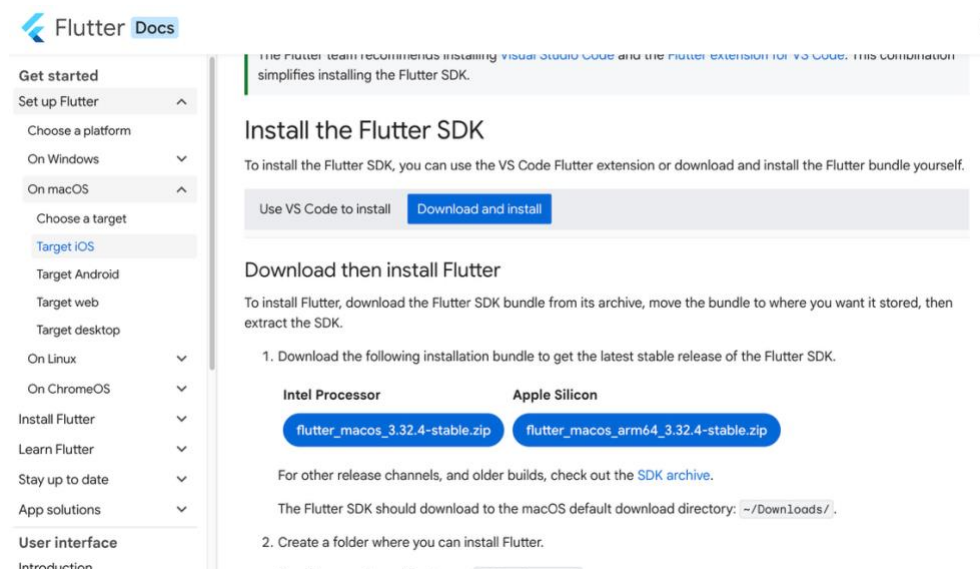


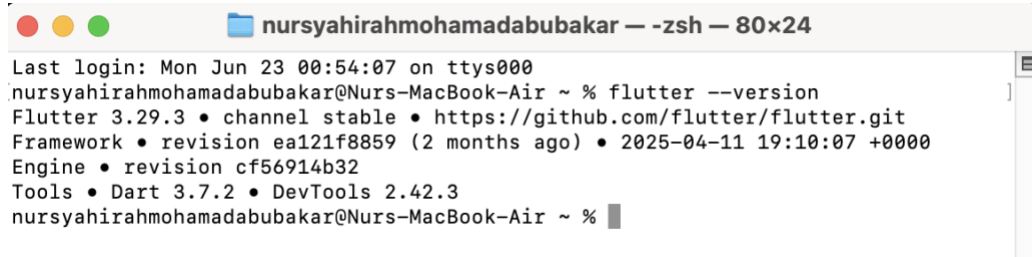
Figure 4.2 Installation Bundle For Ios

2. Flutter Framework Setup

To run Flutter commands in the terminal, Flutter is added to the PATH environment variable. Line as shown in figure 4.3 below is copied and paste at the end of the ‘~/zshenv’ file. To check whether the file is already exist, run a command ‘ls -al’ in the terminal. Figure 4.4 below shows that Flutter has been successfully install to the device where version of 3.29.3 is used for this project.

```
export PATH=$HOME/development/flutter/bin:$PATH
```

Figure 4.3 Flutter path code

A terminal window titled 'nursyahirahmohamadabubakar — -zsh — 80x24'. The terminal shows the output of the command 'flutter --version'. The output includes the Flutter version (3.29.3), channel (stable), framework revision (ea121f8859), engine revision (cf56914b32), and tools (Dart 3.7.2, DevTools 2.42.3).

```
Last login: Mon Jun 23 00:54:07 on ttys000
nursyahirahmohamadabubakar@Nurs-MacBook-Air ~ % flutter --version
Flutter 3.29.3 • channel stable • https://github.com/flutter/flutter.git
Framework • revision ea121f8859 (2 months ago) • 2025-04-11 19:10:07 +0000
Engine • revision cf56914b32
Tools • Dart 3.7.2 • DevTools 2.42.3
nursyahirahmohamadabubakar@Nurs-MacBook-Air ~ %
```

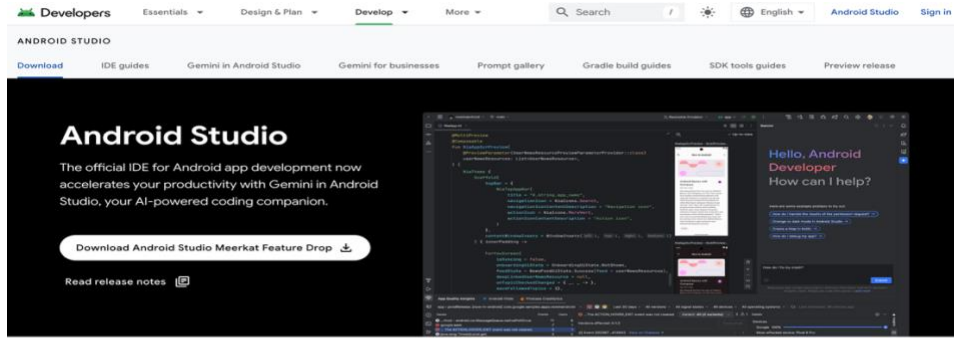
Figure 4.4 Flutter version info

4.2.1.2 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for application by Google. It is mainly used for developing Android application. For this project, Android Studio is also support Flutter based IOS application development that enable codebase for both Android and IOS platforms. It is must support by Xcode to compile and built the iOS version of the application because it itself does not build iOS application independently.

1. Android Studio IDE Installation

To download the Android Studio application, navigate to the <https://developer.android.com/studio> to download from official website. And in this project, Mac with Apple Chip is used because it is compatible with the current device use as per figure 4.6 displayed.



New features

Figure 4.5 Android Studio IDE Download Page

Select the version of Android Studio that's right for your Mac:

Android Studio Meerkat Feature Drop | 2024.3.2 Patch 1

Mac with Apple chip

Mac with Intel chip

android-studio-2024.3.2.15-mac_arm.dmg android-studio-2024.3.2.15-mac.dmg

Figure 4.6 Android Studio version

2. Android Studio IDE Setup

Launch the application upon the success of the download step to start setup the Android Studio. By following the setup and installation step, ready to create a flutter project after install the Flutter plugin as illustrates in figure 4.7.

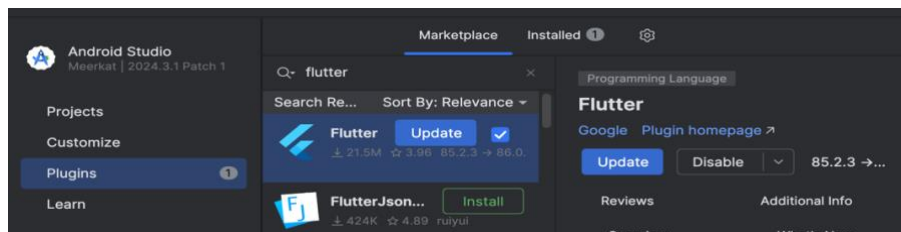


Figure 4.7 Flutter plugin installation

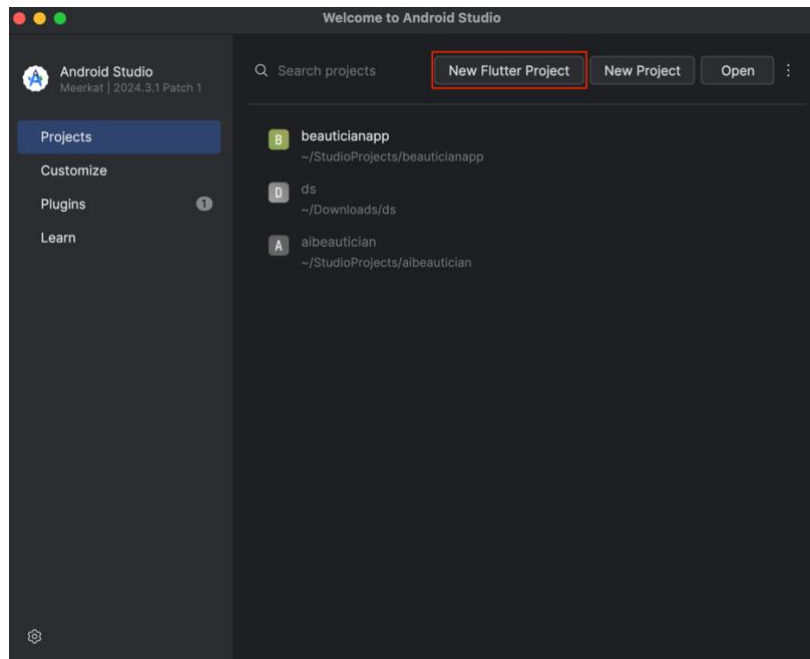


Figure 4.8 New Flutter project creation

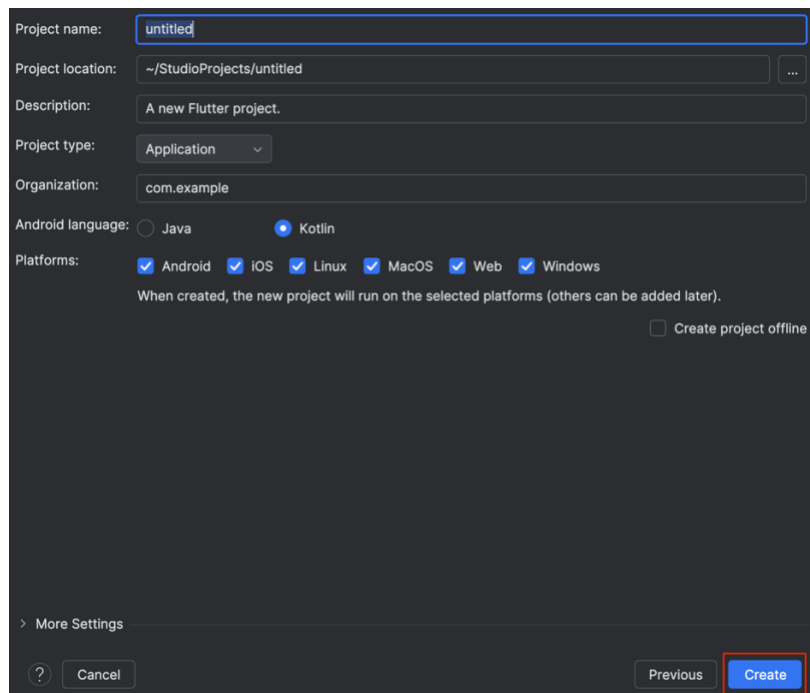


Figure 4.9 Successful Flutter project creation

In figure 4.8 and 4.9 is the steps to create a new flutter project inside the Android Studio IDE application.

4.2.1.3 Xcode

Xcode is Apple's IDE for building application for Apple's platform such as iOS, iPadOS and more. It offers the tools to develop, test, and distribute apps for Apple platforms, including predictive code completion, generative intelligence powered by the best coding models, advanced profiling and debugging tools, and simulators for Apple devices.

1. Xcode IDE Installation

To download the Xcode IDE, navigate to the <https://apps.apple.com/us/app/xcode/id497799835?mt=12/> in app store to download it from official website.

2. Xcode IDE Setup

Agree with all terms shown and follows the installation step given.

4.2.1.4 CocoaPods

1. CocoaPods Installation

To install CocoaPods, open terminal on the computer and prompt command 'sudo gem install cocoapods'. Run the command 'pod --version' to verify the installation. This project used 1.16.2 version of CocoaPods.

```
Last login: Mon Jun 23 01:00:11 on ttys000
nursyahirahmohamadabubakar@Nurs-MacBook-Air ~ % pod --version
1.16.2
nursyahirahmohamadabubakar@Nurs-MacBook-Air ~ % █
```

Figure 4.10. Cocoapods Version

4.2.2 Mobile Application Development

The development of the mobile application involves in two parts which are the functionality and the user interface of the application. The official programming language for Flutter framework called Dart, was used to develop the mobile application. Dart is the programming language that object-oriented specifically designed to support the high-performance, cross-platform that used single codebase development. The language also offers good tools, hot reload functionality, and seamless integration with native APIs. This features enable the development of responsive and user-friendly application interface on both platforms.

The main functionality for this application is the text recognition feature through image capture and load image from gallery. The application displayed the recognizer screen right after launching the application to capture the image. Upon launching the application users enable to capture image, open gallery to load the image, and flip camera to back or front camera.

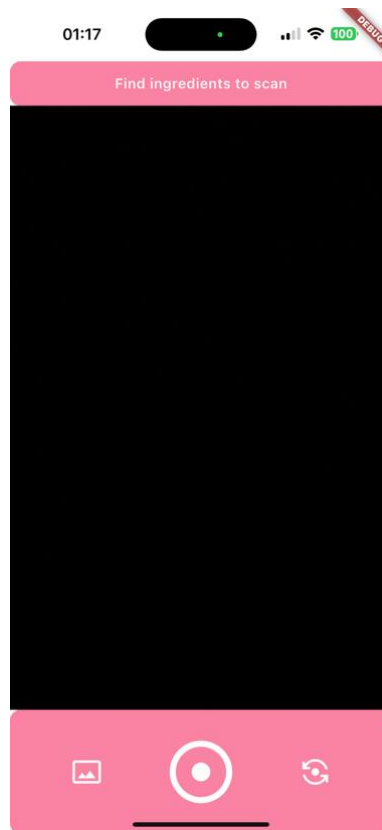


Figure 4.11. Recognizer Main Screen

```
camera: ^0.11.1
```

Figure 4.12. Camera Plugin

```
), // InkWell
InkWell(
  child: const Icon(
    Icons.adjust_rounded,
    size: 85,
    color: Colors.white,
  ), onTap: () async { // Icon
    await controller.takePicture().then((value){
      if(value != null){
        File image = File(value.path);
        processImage(image);
      }
    })
  }
)
```

Figure 4.13. Capture Picture Code Segment

```
<key>NSCameraUsageDescription</key>
<string>Would like to access your camera</string>
```

Figure 4.14. Camera Permission Code

To recognize and analyse the image, 2 options is provided which are to capture image from the live camera or to open the gallery from the user device. Camera plugin in integrate to access the camera function for the flutter project. The user received a permission before able to open the camera as for sensitive information. Once the permission is given, the user can use the camera function. When the shutter button is tapped, it capture the image from the camera and sent for further text recognition process.

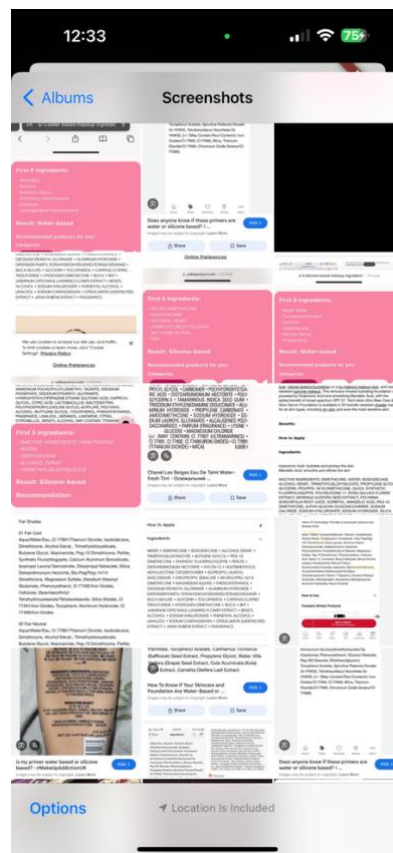


Figure 4.15. Gallery

```
image_picker: ^1.1.2
```

Figure 4.16. Image Picker Plugin

```

InkWell(
  child: const Icon(
    Icons.photo_size_select_actual_outlined,
    size: 35,
    color: Colors.white,
  ), onTap: () async { // Icon
    XFile? xfile = await imagePicker.pickImage(source: ImageSource.gallery);
    if(xfile != null){
      File image = File(xfile.path);
      processImage(image);
    }
  },
)

```

Figure 4.17. Image Picker Code Segment

```

<key>NSPhotoLibraryUsageDescription</key>
<string>Would like to access your photos</string>

```

Figure 4.18. Gallery Permission Code

Image_picker plugin is used to in this project to access the open gallery function. If the user tapped the gallery icon on the left side bar of the recognizer screen, the user need to approve the permission before they have the access to the gallery. Then they able to access and choose any image from gallery after given permission. The image is analysed for the text recognition process, then displayed to the next recognizer result page.

```

InkWell(
  child: const Icon(
    Icons.flip_camera_android_sharp,
    size: 35,
    color: Colors.white,
  ), onTap: () async { // Icon
    if (_cameras.length < 2) return; // If only 1 camera, don't switch

    selectedCameraIndex = selectedCameraIndex == 0 ? 1 : 0; // Toggle
    controller = CameraController(
      _cameras[selectedCameraIndex],
      ResolutionPreset.max,
    ); // CameraController
    await controller!.initialize();
    setState(() {}); // Refresh the preview
  },) // InkWell

```

Figure 4.19. Flip Camera Code Segment

Other than back camera, when the user click the flip camera button, it allows the camera to flip from back camera to front camera.

```

google_mlkit_text_recognition: ^0.15.0

```

Figure 4.20. Google ML Kit Plugin

```

Future<void> doTextRecognition() async {
  final inputImage = InputImage.fromFile(widget.image);
  final recognizedText = await textRecognizer.processImage(inputImage);
  final extractedText = recognizedText.text;
}

```

Figure 4.21. Image Process Code Segment

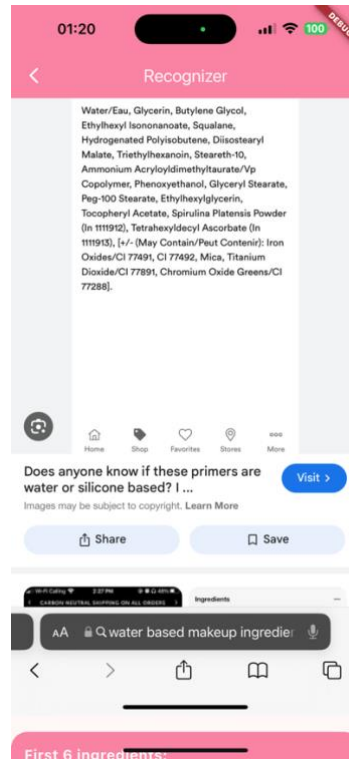


Figure 4.22. Text Recognition Result Page

Figure 4.20 above shows plugin `google_mlkit_text_recognition` version 0.15 was used for text recognition feature during the implementation of the application. This plugin allow the user to use the text recognition feature. The code segment displayed in figure 4.21 shows that the application perform the text recognition using ML Google Kits from image captured or image from the gallery.

```
Widget build(BuildContext context) {
  final ingredients = results
    .split(RegExp(r'[,-]'))
    .map((e) => e.trim())
    .where((e) => e.isNotEmpty)
    .take(6)
    .toList();
}
```

Figure 4.23. First 6 Ingredient Code Segment

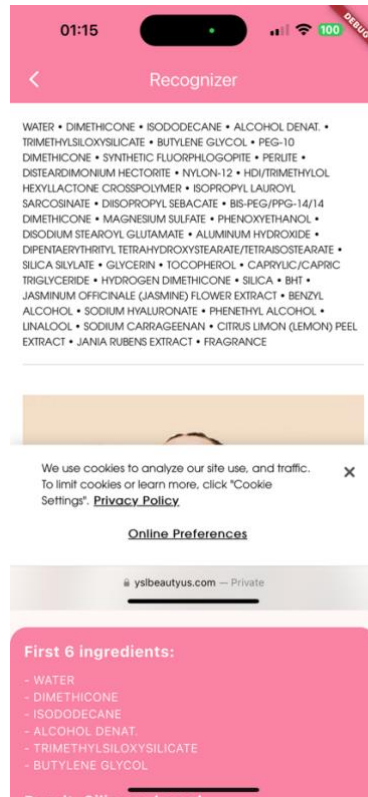


Figure 4.24. First 6 Ingredient Result Page

```
Text('First 6 ingredients:', style: TextStyle(color: Colors.white, fontSize: 18, fontWeight: FontWeight.bold)),
  SizedBox(height: 10),
  ...ingredients.map((item) => Text("- $item", style: TextStyle(color: Colors.white))),
  SizedBox(height: 20),
  Text('Result: $baseType', style: TextStyle(color: Colors.white, fontSize: 18, fontWeight: FontWeight.bold)),
```

Figure 4.25. Displaying Result Code Segment

The figure 4.25 above shows the code segment of process where the application processed the image and split the result it into string parts based on symbols comma (,), bullet, and hyphens (-) and the result is turn into list of string. After the process done, the ingredient list that has undergo the text recognition process will be listed by the first 6 ingredient as shown in figure 4.23, the code function is used for focusing more on the most prominent or initial ingredient which typically most influential in the cosmetic product. The function will filter any empty string to ensure that only valid, non-empty ingredient are listed and displayed inside the recognizer screen as illustrate in figure 4.24 and the result seen as in figure 4.25 is the code segment to display the result.

```
String classifyBaseType(List<String> ingredients) {
    final siliconeKeywords = [
        'dimethicone', 'cyclopentasiloxane', 'cyclohexasiloxane', 'cyclomethicone',
        'phenyl trimethicone', 'dimethiconol', 'amodimethicone', 'trimethylsiloxysilicate',
        'vinyl dimethicone', 'methicone', 'siloxane', 'silicone', 'silica', 'polysilicone',
        'dimethicone crosspolymer', 'cetearyl methicone', 'cetyl dimethicone', 'stearyl dimethicone'
    ];

    int siliconeCount = ingredients.take(6).where((ingredient) =>
        siliconeKeywords.any((keyword) => ingredient.toLowerCase().contains(keyword))).length;

    bool startsWithWater = ingredients.isNotEmpty && ingredients.first.toLowerCase().contains('water');
    return (startsWithWater && siliconeCount < 2) ? 'Water-based' : 'Silicone-based';
}
```

Figure 4.26. Base Type Classification Code Segment

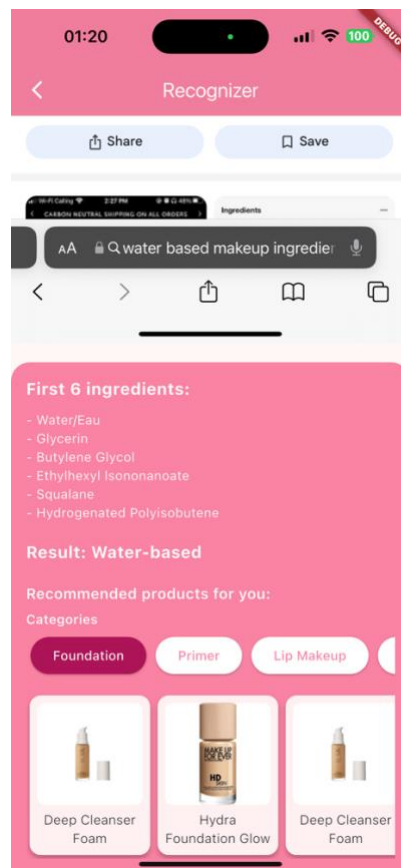


Figure 4.27. Water-Based Base Type Result

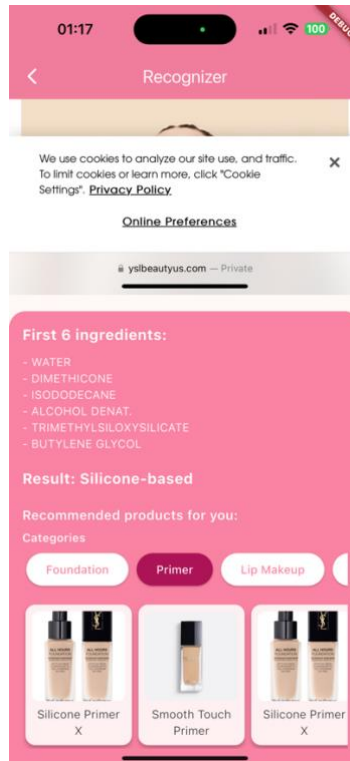


Figure 4.28. Silicone-Based Base Type Result

The provided code segment display in above figure is a core segment of the application, which function for classifying the cosmetic products based on their based type formulation either ‘Silicon-based’ or ‘Water-based’. The classification is based on the analysing the first 6 ingredient that extract from the product label during earlier process of text recognition using Optical Character Recognition (OCR). A predefined list of silicone-related keywords are declared including the common silicone bases which are demethicone, cyclopentasiloxane, silica and more. The algorithm will check when the first keyword is water and the rest of the 6 ingredient contains two or more silicone-related keyword, the product is classified as ‘Silicone-based’ product. Otherwise, it is classified as ‘Water-based’ product.

```

final products = snapshot.data!;
print(' $baseType, $selectedCategory');
products.forEach((p) => print('${p.name}, ${p.imagePath}'));
return SizedBox(
  height: 180,
  child: ListView.builder(
    scrollDirection: Axis.horizontal,
    itemCount: products.length,
    itemBuilder: (context, index) {
      final product = products[index];
      return Card(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(12),
        ), // RoundedRectangleBorder
        elevation: 4,
        child: Container(
          width: 130,
          padding: EdgeInsets.all(8),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
              ClipRRect(
                borderRadius: BorderRadius.circular(8),
                child: Image.asset(
                  product.imagePath,
                  height: 110,
                  width: double.infinity,
                  fit: BoxFit.cover,
                ), // Image.asset
              ), // ClipRRect
              SizedBox(height: 6),
              Text(product.name,
                style: TextStyle(

```

Figure 4.29. Recommendation Card Code Segment

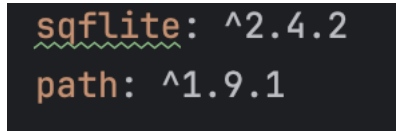
assets:

- assets/images/silicone1.jpg
- assets/images/silicone2.jpg
- assets/images/water1.jpg
- assets/images/water2.jpg
- assets/images/water3.jpg
- assets/images/water4.jpg
- assets/images/water5.jpg

Figure 4.30 Example Of Images

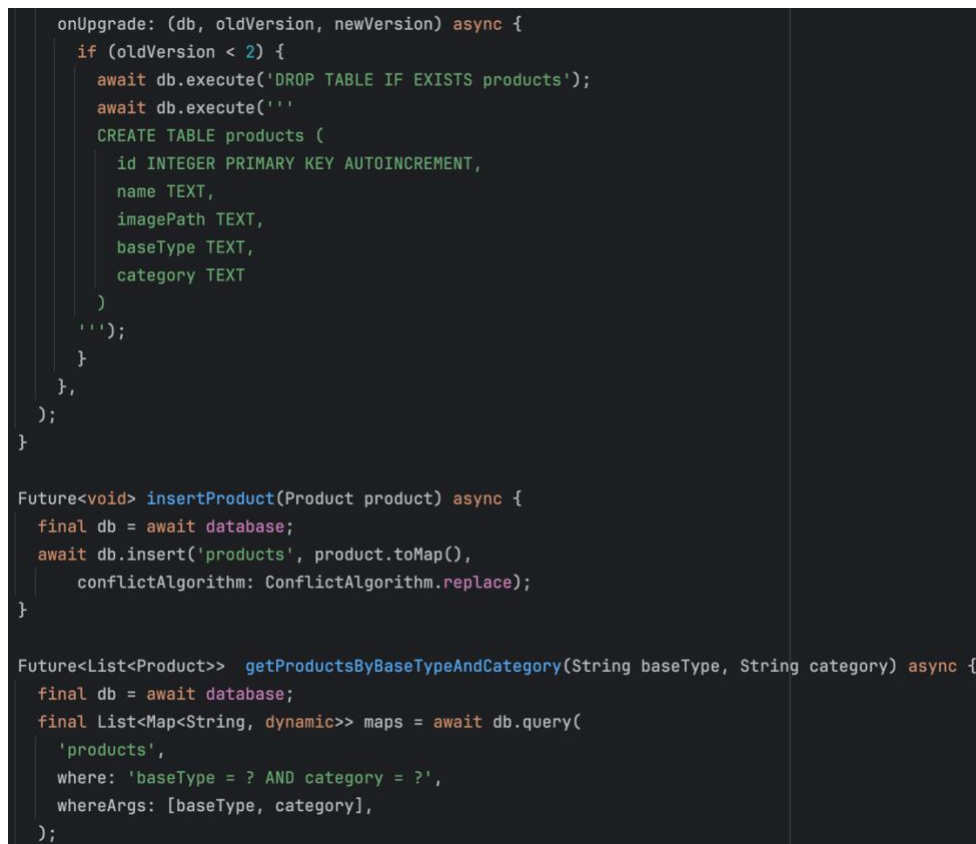
Above code segment in figure 4.29 is implemented for dynamic user interface component within the mobile application. A `ListView.builder` widget is implement to efficiently render a horizontal scrollable list of product cards. Each product image displayed using the `image.asset` widget will load image from local assets which example display in figure

4.30. The UI component delivered a personalized and user-friendly experience by providing a product recommendation that align with the user's skin preferences.



```
sqflite: ^2.4.2
path: ^1.9.1
```

Figure 4.31. SQLite Plugin



```
onUpgrade: (db, oldVersion, newVersion) async {
  if (oldVersion < 2) {
    await db.execute('DROP TABLE IF EXISTS products');
    await db.execute('''
      CREATE TABLE products (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT,
        imagePath TEXT,
        baseType TEXT,
        category TEXT
      )
    ''');
  },
};

Future<void> insertProduct(Product product) async {
  final db = await database;
  await db.insert('products', product.toMap(),
    conflictAlgorithm: ConflictAlgorithm.replace);
}

Future<List<Product>> getProductsByBaseTypeAndCategory(String baseType, String category) async {
  final db = await database;
  final List<Map<String, dynamic>> maps = await db.query(
    'products',
    where: 'baseType = ? AND category = ?',
    whereArgs: [baseType, category],
  );
}
```

Figure 4.32. SQLite Database Code Segment

This project implement SQLite database that focus on data schema versioning, data insertion, and data retrieval mechanisms. The recommended product data will be prompt manually into the database table using 'insertProduct' method. To ensure that any existing entries data with the same primary 'ConflictAlgorithm.replace' is implemented to maintaining data integrity and consistency. Furthermore, the 'getProductsByBaseTypeAndCategory' method implements a conditional query that retrieves a filtered list of products matching both a specified base type and category for user preferences.

```

class Product {
    final int? id;
    final String name;
    final String imagePath;
    final String baseType; // "Silicone-based" or "Water-based"
    final String category;

    Product({
        this.id,
        required this.name,
        required this.imagePath,
        required this.baseType,
        required this.category,
    });

    Map<String, dynamic> toMap() {
        return {
            'id': id,
            'name': name,
            'imagePath': imagePath,
            'baseType': baseType,
            'category': category,
        };
    }
}

factory Product.fromMap(Map<String, dynamic> map) {
    return Product(
        id: map['id'],
        name: map['name'],
        imagePath: map['imagePath'],
        baseType: map['baseType'],
        category: map['category'],
    );
}

```

Figure 4.33. Product Model Code Segment

The product class as illustrate figure 4.33 above provides efficient and clean structure that managing the product data. It support the functionality of the application by enabling retrieval and manipulation of product data records, which used to deliver accurate, personalized recommendations to the user based on their needs.

```

import 'db_helper.dart';
import 'product_model.dart';

Future<void> insertSampleProducts() async {
  final dbHelper = DBHelper();

  // 🧴 Skin Care
  await dbHelper.insertProduct(Product(
    name: 'Hydrating Gel Cream',
    imagePath: 'assets/images/silicone1.jpg',
    baseType: 'Silicone-based',
    category: 'Foundation',
  ));

  await dbHelper.insertProduct(Product(
    name: 'Deep Cleanser Foam',
    imagePath: 'assets/images/water1.jpg',
    baseType: 'Water-based',
    category: 'Foundation',
  ));

  // 🌟 Primer
  await dbHelper.insertProduct(Product(
    name: 'Silicone Primer X',
    imagePath: 'assets/images/silicone1.jpg',
    baseType: 'Silicone-based',
    category: 'Primer',
  ));

  await dbHelper.insertProduct(Product(
    name: 'Smooth Touch Primer',
    imagePath: 'assets/images/water3.jpg',
    baseType: 'Silicone-based',
    category: 'Foundation',
  ));
}

```

Figure 4.34. Data Sample Code Segment

The function of the `insertSampleProducts()` is to populate the local SQLite database with cosmetic product data. Installing `DBHelper` class in Dart to provide an interface for interacting with the database. Each product encapsulates attributes including product's name, image, path, base type, and also category of the product. By incorporating visual assets, the application enhance user's engagement through a visually informative interface.

4.3 Summary

Conclusively, this chapter has described the system's entire implementation cycle in accordance with the guidelines provided in Chapter 3. Google ML Kit was integrated for text recognition in real-time that change the image into textual data. The features of the application are gallery scanning and real-time scanning. Both features proved to be functional and ensure that users received real-time recognition results. Then, the classification of the ingredient base type is analysed to generate the results of the product base. The mobile application then provides a recommendation according to the product based type in various categories.

CHAPTER 5: SYSTEM TESTING

5.1 Introduction

In this chapter, the mobile application was tested to validate the functionality of the AIBeautician application. Testing is required to validate the effectiveness of the proposed features in the application as if it achieving its objectives. By performing thorough testing, potential issues and vulnerabilities can be identified and resolved, ensuring the quality of the mobile application.

5.2 Functional Testing

Black-box testing was mainly employed to test the application in this project, as mentioned in the previous chapter. The black-box testing focuses on assessing the functionality of the application without requiring knowledge of its internal code or structure. According to Murray (2020), the input and output of the application is being study without understanding the core or internal logic design of the software project. Two types of Black-Box Testing was used which are Functional Testing and Non-Functional Testing.

Functional testing is a type of Black-Box software testing that assesses an application's or system's features and functions in order to ensure the application works in accordance with the requirement. By comparing input-output behaviour with anticipated outcomes, its test the software's functionality against expected result. This testing can be done both automatically and manually. There are six features that were tested, and the results are shown in the test cases tables below.

Table 5.2.1. Test Case Module With Description

Test Case ID	Test Modules	Description
TC01	Live Text Recognition	User can scan the product's ingredient in real-time using phone camera.
TC02	Gallery Text Recognition	Users can upload an image from the gallery to extract ingredient text.

TC03	Ingredient Processing and Classification	The application process and classified the product based type into Silicone-based and Water-based.
TC04	Product Recommendation	Recommends products that match the user's detected base type.
TC05	Category Filtering	Allows filtering product recommendations by categories

A detailed test case for each of the identified modules shown in the next section.

5.2.1 Test Case for Live Text Recognition

Table 5.2.2 Test Case For Live Test Recognition

Test Case ID		TC01		
Test Description		User can scan the product's ingredient in real-time using phone camera.		
Objectives		Enable user to use real-time scanning.		
Pre-Condition		User reached the main page of the application.		
Post- Condition		User scan the image of the product's ingredient in real-time using their phone camera.		
#	Test Scenario	Test Steps	Expected Result	Status (Pass/Fail)
1.	Access camera	Launch the app and go to the recognizer screen for activate the camera view.	Camera view is visible.	Pass
2.	1. Scan Text 2. Text Recognition	1. Aim camera at product ingredients. 2. Capture the image. 3. Crop the product ingredient image. 4. Click the capture image button.	1. Application able to capture and crop the image. 2. Text is recognized in real time. 3. Text from product label is captured and displayed correctly.	Pass
3.	Display result	Display the first 6 ingredient result in the recognizer screen.	The first 6 ingredient is displayed.	Pass

5.2.2 Test Case for Gallery Text Recognition

Table 5.2.3 Test Case For Gallery Text Recognition

Test Case ID		TC02		
Test Description		Users can upload an image from the gallery to extract ingredient text.		
Objectives		Enable the user to perform text recognition on uploaded image from gallery.		
Pre-Condition		1. Application has permission to access the gallery. 2. The image contains a readable product ingredient list.		
Post- Condition		User scan the image of the product's ingredient that uploaded from gallery.		
#	Test Scenario	Test Steps	Expected Result	Status (Pass/Fail)
1.	Open Image Upload	Launch the app and go to the recognizer screen.	Recognizer screen opens.	Pass
2.	Upload Image	Tap the upload icon on the bottom left side and select an image from the gallery. Crop the product ingredient image.	Image is successfully loaded and displayed.	Pass
3.	Text Recognition	App processes the uploaded image using ML Kit.	Text is extracted from the image.	Pass
4.	Display Result	Display the first 6 ingredient result in the recognizer screen.	The first 6 ingredient is displayed.	Pass

5.2.3 Test Case for Ingredient Processing And Classification

Table 5.2.4 Test Case For Ingredient Processing And Classification

Test Case ID	TC03
Test Description	The application process and classified the product based type into Silicone-based and Water-based.
Objectives	Application able to identify and classify base type using predefined keywords.
Pre-Condition	Recognized text and the first 6 ingredient result is displayed.

Post- Condition		Classification result of ‘Silicone-based’ or ‘Water-based’ is displayed.		
#	Test Scenario	Test Steps	Expected Result	Status (Pass/Fail)
1.	Parse Ingredients	View extracted ingredients after text recognition.	First 6 ingredients are parsed.	Pass
2.	Classify Type	System uses logic to detect base type. Base type is displayed on the screen.	Base type (Silicone-based/Water-based) is displayed.	Pass

5.2.4 Test Case for Product Recommendation

Table 5.2.5 Test Case For Product Recommendation

Test Case ID		TC04		
Test Description		Recommends products that match the user’s detected base type.		
Objectives		User is provided with products recommendation according to their base type.		
Pre-Condition		Product base type and category are identified.		
Post- Condition		Product recommendation cards are visible on screen.		
#	Test Scenario	Test Steps	Expected Result	Status (Pass/Fail)
1.	Load Products	System loads products for the base type.	Relevant products are loaded.	Pass
2.	Display Cards	View horizontal card list.	Product cards are shown with image and name.	Pass

5.2.5 Test Case for Category Filtering

Table 5.2.6 Test Case For Category Filtering

Test Case ID		TC05		
Test Description		Allows filtering product recommendations by categories		
Objectives		User able to select category tab to filter results.		
Pre-Condition		Products for selected base type are loaded.		
Post- Condition		Only products matching selected category are shown.		
#	Test Scenario	Test Steps	Expected Result	Status (Pass/Fail)

1.	Select Category	Tap a category filter (Foundation, Primer, Lip Product, Skin Care).	Button highlights as selected.	Pass
2.	Load Filtered	Application fetches products for the base type and category chosen.	Filtered product list appears.	Pass

5.3 Non-Functional Testing

Non-functional testing is a type of black-box software testing that performed to verify the non-functional requirement of the application. It assesses whether the behaviour of the AIBeautician application is aligns with the requirement or not. This form of testing evaluate all aspects that are not tested in the functional testing, as discussed in the previous section. Table below are the detailed of the Non-Functional Testing of the application.

Table 5.3.1 Non-Functional Test

Objective: To evaluate the system's performances.	
Performance Testing	1. The application process recognition and displayed base type results, recommendation products within 2-3 seconds. 2. The user is provided with the first 6 ingredient of the product.
Usability Testing	1. User complete the ingredients scan and received results in less than 5 interactions. 2. The application interface is designed to be straightforward, ensuring a smooth and user-friendly experience.
Reliability Testing	1. The application handles invalid images gracefully by displaying an error message when no recommended products are found in the dataset.

5.4 Summary

In conclusion, this phase outlines the testing methodology used in this project, specifically 'Black-Box Testing', which include both functional testing and also non-functional testing. The section provide an overview of validation process the mobile application and its functions

through testing the application. The functional testing takes place in testing the core functions of the application, which are Live Text Recognition, Gallery Text Recognition, Ingredient Processing and Classification, Product Recommendation, and Category Filtering module. For Non-Functional Testing test the performance, usability and reliability of the application. Through this test, all the functions of the mobile application has been implemented successfully, thereby fulfils the project objectives.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Introduction

This chapter examines the conclusion of the project, that focuses on the prototype's performance as well as future work to improve the project of AIBeautician: An AI-Powered Mobile App for Personalized Cosmetic Ingredient Analysis and Recommendations. This chapter is broken down into three subsection. The proposed objectives is reviewed with the outcome of the developed application to ensure that the prior goals have been achieved through out this project. The limitations of the application and the potential of improvement in future works related to this project also discussed in this chapter.

6.2 Achievements

As shown in table below, the proposed mobile application created to achieve the goals outlined in Chapter 1. The project's achievement is elaborated as follows:

Table 6.2.1 Project Achievement

	Objectives	Objectives Fulfilment
i.	To design a user-friendly mobile application interface, ensuring seamless consumer engagement and real-time feedback delivery for accessing cosmetic products.	Successfully developed a mobile application with an intuitive and responsive mobile interface using Flutter. As the interface implements a user-friendly design, it offers straightforward navigation, minimal cognitive load, and pleasing layout.
ii.	To develop an advanced application capable of scanning and analysing the ingredient list of cosmetic products, accurately classifying them as silicone-based or water-based.	The implementation of Google ML Kit's Text Recognition API within the AIBeautician application successfully integrate text extraction from both gallery images and live camera feature. Classification algorithm was implement based on the keywords found in silicone-based and water-based products. This objective is considered effectively achieved

		when the results is correctly distinguish according to based type on real-world product.
iii.	To evaluate the mobile application's performance, effectiveness, and user satisfaction while providing tailored recommendations based on the analysis results.	By providing recommendation engine reliably suggested products aligned with the classified base type and allow user to select product based on selected categories given, thereby achieving the third objective.

6.3 Limitation

Throughout the development of this project, numbers of limitations were identified during the implementation phase of the mobile application. Although all the project objectives have been achieved as shown in Chapter 5, these limitations may develop barrier that could affect the quality and the overall performance of the application.

- i. The accuracy of the text recognition may be in accurate due to the environment aspects such as poor lighting and blurry images, that can lead to incorrect or incomplete ingredient extraction.
- ii. The recognition engine extract all visible text from the images. If the scan portion include unrelated content to the ingredient list, it may be captured and listed among the first 6 ingredient. Thus this can lead to inaccurate base type classification.
- iii. The application relies on a predefined list of silicone-related keywords, may limit the effectiveness of the engine in detecting new or uncommon chemical variants, resulting in misclassification of the product.
- iv. Recommendation products are based on static product database. Hence, new cosmetic product or updated formulations cannot be auto added unless they manually added to the database.

6.4 Future works

Future improvements can be implemented in order to overcame the limitations of the application. Firstly, the accuracy of Optical Character Recognition (OCR) can be improve the by incorporate advanced image processing techniques and real-time guidance to improve text recognition under poor or blurry conditions. Secondly, to focus solely on the part of the

ingredients image, implement a region-of-interest (ROI) feature can reducing misclassification from irrelevant text.

Furthermore, the predefined silicone-related keyword list can be expand through automated updates or machine learning models to detect the chemical compound keywords. To address limitation of the recommendation product being based on static product database, it can be improve by linking the application to an online cosmetic database to provides the user with up-to-date product recommendations, thereby eliminating reliance on static data.

In addition, to achieve more usability function and interactivity of the application, an AI-powered chatbot feature is integrate to provide users with real-time, personalized beauty advice and product alternatives based on their needs.

REFERENCES

- About Our pregnancy-safe ingredient Checker App - Little Bean App.* Pregnancy Safe Ingredient Checker- Little Bean App. (2024, December 6). <https://littlebeanapp.com/about-little-bean-app/>
- ADA Cosmetics. (2024, June 26). *The Truth about Silicones in Cosmetics.* <https://ada-cosmetics.com/expert-stories/silicones/>
- Analyze the composition of your cosmetics.* INCI Beauty. (n.d.). <https://incibeauty.com/theme/2-1-application-inci-beauty>
- appasion. (2024, October 4). *Safely: Cosmetic Ingredients.* Soft112 Reviews. <https://safely-cosmetic-ingredients.soft112.com/>
- Arkin, B., McGraw, G., & Stender, S. (2005). Software Penetration Testing. *IEEE Secur. Priv.*, 3, 84-87. <https://doi.org/10.1109/MSP.2005.23>.
- Basias, C. (2023, June 7). *DEBUNKING.* TikTok. <https://www.tiktok.com/@thelipsticklesbians/video/7241771806120709422?lang=en>
- Chanda, Dr. A. (2019). Barcode Technology and its application in libraries. *SSRN Electronic Journal.* <https://doi.org/10.2139/ssrn.3649957>
- Chien, C. (2020, February 4). *What is Rapid Application Development (RAD)?* Codebots. <https://codebots.com/app-development/what-is-rapid-application-development-rad>
- Despa, M. L. (2014). Comparative study on software development methodologies. *Database Systems Journal*, 5(3).

- Elder, A., Cappelli, M. O., Ring, C., & Saedi, N. (2024). Artificial Intelligence in cosmetic dermatology: An update on current trends. *Clinics in Dermatology*, 42(3), 216–220. <https://doi.org/10.1016/j.clindermatol.2023.12.015>
- Google. (2024, July 10). *Text recognition v2 | ML kit | google for developers*. Google. <https://developers.google.com/ml-kit/vision/text-recognition/v2>
- Hurwitz, K. (2020, October 30). *What is software prototyping and why is it so important?*. Praxent. <https://praxent.com/blog/dont-trust-project-estimate-without-ux-prototype>
- Marie, C., Garlantézec, R., Béranger, R., & Ficheux, A. (2022). Use of cosmetic products in pregnant and breastfeeding women and young children: Guidelines for interventions during the perinatal period from the French National College of Midwives. *Journal of Midwifery & Women's Health*, 67(S1). <https://doi.org/10.1111/jmwh.13428>
- MaryAnn, M. (2022, April 29). *Silicone for skin: Uses, benefits, risks, and more*. Medical News Today. <https://www.medicalnewstoday.com/articles/silicone-for-skin#alternatives>
- Nurhaida, I., Ayumi, V., Noprisson, H., Ratnasari, A., Utami, M., & Putra, E. D. (2020). Web development using WISDM and Rad. *2020 International Conference on Information Technology Systems and Innovation (ICITSI)*, 57–61. <https://doi.org/10.1109/icitsi50517.2020.9264939>
- Pitton, E. (2019, October 23). *Disclosing product ingredients information through scanning mobile apps: Impacts on consumer purchasing behavior*. Retrieved from <http://hdl.handle.net/10579/15766>.
- Rubio, A, Amandine & Ponvert. C, Claude & Goulet. O, O & Scheinmann, P & Blic, Jacques. (2009). Allergic and nonallergic hypersensitivity reactions to silicone: a report of one case <https://doi.org/10.1111/j.1398-9995.2009.02086.x>

- Sitanggang, D., Juan, A., Doa, G. D., & Indra, E. (2024). DESIGNING A COUNTERFEIT-PRODUCT-CHECK APPLICATION USING ANDROID BASED BARCODE SCANNER IN COSMETICS PRODUCTS. *JURNAL INFOKUM*, 9, 364–364. <https://doi.org/https://seaninstitute.org/infor/index.php/infokum/article/view/137>
- Soobia.et.al., S. (2019). Analysis of Software Development Methodologies. *International Journal of Computing and Digital Systems*, 8(5), 445–460. <https://doi.org/10.12785/ijcds/080502>
- Spectrum Collections. (2023, April 5). *Why does my makeup look cakey?* <https://shorturl.at/iVIwE>
- Sqlite Home Page*. SQLite Home Page. (n.d.). <https://www.sqlite.org/>
- Vithani, T., & Kumar, A. (2014). Presentation 5. A comprehensive mobile application development and testing lifecycle. *2014 IT Professional Conference, Gaithersburg, MD, USA, 2014, 1*, 1–27. <https://doi.org/10.1109/ITPRO.2014.7029288>

APPENDIX

Questionnaire For Final Year Project "AiBeautician: An AI-Powered Mobile App for Personalised Cosmetic Ingredient Analysis and Recommendations"

Dear Respondent,

Thank you for taking the time to participate in this questionnaire . My name is Nur Syahirah Binti Mohamad Abu Bakar, and I am a final-year Bachelor of Software Engineering (Honours) undergraduate at the Faculty of Computer Science and Information Technology (FCSIT), Universiti Malaysia Sarawak (UNIMAS). I am currently working on my Final Year Project titled: "AiBeautician: An AI-Powered Mobile App for Personalised Cosmetic Ingredient Analysis and Recommendations". By completing this questionnaire, you acknowledge your voluntary participation.

The goal of this project is to develop and provide an intelligent platform for the beauty community, enabling users to analyse and evaluate cosmetic products ingredients. Through this survey, I aim to gather insights into your experiences and preferences regarding cosmetic products, particularly in terms of ingredient analysis, formulation types, and product recommendations. Additionally, we are exploring the potential use of a mobile application for easier, on-the-go analysis of cosmetic products.

The survey consists of five sections:

- Section 1: Demographic Information
- Section 2: Current Practices
- Section 3: Challenges
- Section 4: Interest in Features
- Section 5: User Experience and Recommendations

This survey complies with Malaysia's Personal Data Protection Act 2010 (PDPA). All personal information collected will be used solely for academic purposes and kept confidential in accordance with the PDPA. The survey should take about 5-10 minutes to

Section 1: Demographics *

What is your age group?

- ☐ Under 18
- ☐ 18 - 24
- ☐ 25 - 34
- ☐ 35 - 44
- ☐ 44 and above

What is your gender? *

- ☐ Male
- ☐ Female
- ☐ Other: _____

What is your current role or occupation? *

- ☐ Student (e.g., university/college student)
- ☐ Beauty Professional (e.g., makeup artist, cosmetologist, skincare expert)
- ☐ Employed in the beauty industry (e.g., product development, marketing)
- ☐ Regular cosmetic user (e.g., not working in the beauty industry but frequently use cosmetic products)

How often do you use cosmetic products? (e.g., makeup,skincare) *

- ☐ Rarely
- ☐ Occasionally
- ☐ Weekly
- ☐ Daily

Section 2: Current Practices

How do you typically choose cosmetic products? (Select all that apply) *

- ☐ Based on brand reputation
- ☐ Recommendations from friends/family
- ☐ Online reviews
- ☐ Ingredient analysis
- ☐ Other: _____

How do you currently identify whether a product is suitable for your skin type? *

- ☐ Personal experience
- ☐ Recommendations from others
- ☐ Ingredient research
- ☐ Trial and error

Are you aware of the differences between silicone-based and water-based cosmetics? *

- ☐ Yes
- ☐ No

Do you currently check ingredient lists when purchasing cosmetic products? *

- ☐ Always
- ☐ Never
- ☐ Sometimes

Section 3: Challenges

Do you find it difficult to understand ingredient lists on cosmetic products? *

- ☐ Yes
- ☐ No
- ☐ Sometime

Do you find it hard to analyse the difference between silicone-based and water-based products? *

- ☐ Yes
- ☐ No

Have you ever experienced patchy or uneven makeup due to product incompatibility (e.g., mixing silicone-based and water-based products)? *

- ☐ Yes
- ☐ No
- ☐ Not Sure

What are your main challenges when selecting cosmetic products? (Select all that apply) *

- ☐ Understanding ingredient lists
- ☐ Finding products suitable for my skin type
- ☐ Identifying safe products (e.g., vegan, pregnancy-safe)
- ☐ Comparing similar products
- ☐ Ensuring compatibility of products
- ☐ Other: _____

Section 4: Interest in Features

Would you use a tool that analyses cosmetic products and recommends similar ones? *

- ☐ Yes
- ☐ No
- ☐ Maybe

How important is real-time analysis to you when evaluating products? *

- ☐ Very important
- ☐ Somewhat important
- ☐ Not important

Have you ever used a mobile app to scan or analyze your cosmetic products? If yes, which one? *

- ☐ INCI Beauty
- ☐ Safely: Cosmetic Ingredients
- ☐ Little Bean – Cosmetic Checker
- ☐ No
- ☐ Other: _____

Which features would you find most useful in a mobile app for cosmetic analysis? *
(Select all that apply)

- ☐ Ingredient scanner (camera-based)
- ☐ Product safety ratings
- ☐ Recommendations based on skin condition
- ☐ Comparison and recommendation of similar products
- ☐ Database search for products or ingredients
- ☐ Other: _____

Section 6: User Experience and Recommendations

How comfortable are you with using online tools for product recommendations? *

- ☐ Very comfortable
- ☐ Somewhat comfortable
- ☐ Not comfortable

Would you prefer a simple and straightforward app or a feature-rich application? *

- ☐ Simple and straightforward
- ☐ Feature-rich and detailed

If you prefer a feature-rich application, which features would you like it to include? *
(Select all that apply)

- ☐ Login/Sign-up
- ☐ Barcode/QR code scanner
- ☐ Detailed product recommendations
- ☐ Ingredient database search
- ☐ Side-by-side product comparisons
- ☐ Other: _____

How likely are you to recommend this tool to others? *

	1	2	3	4	5	
Less likely	☐	☐	☐	☐	☐	Very likely

How specific do you expect the app's recommendations to be? *

- ☐ Based on similar formulation (e.g., recommending products with the same ingredient composition)
- ☐ Broad categories (e.g., "suitable for dry skin")
- ☐ Detailed (e.g., "best for sensitive, combination skin with hydration needs")
- ☐ Other: _____

What are your main frustrations with current cosmetic apps or tools? (Select all that apply) *

- ☐ Difficult navigation
- ☐ Lack of accurate or updated ingredient information
- ☐ Limited product recommendations
- ☐ Poor performance or speed
- ☐ Premium features require a subscription
- ☐ Other: _____