



Faculty of Computer Science and Information Technology

**Recommendation System for Selecting Suitable Programmes in University
based on Student Interest**

MOHAMMAD KHAIRUL IKHWAN BIN ABDUL BASIT

Bachelor of Computer Science with Honours (Software Engineering)

2025

Recommendation System for Selecting Suitable Programmes in University based on Student Interest

MOHAMMAD KHAIRUL IKHWAN BIN ABDUL BASIT

This project is submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE: Recommendation System for Selecting Suitable Programmes in University based on Student Interes

ACADEMIC SESSION: 2024/2025

MOHAMMAD KHAIRUL IKHWAN BIN ABDUL BASIT

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

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

☐

CONFIDENTIAL

(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)

☐

RESTRICTED

(Contains restricted information as dictated by the body or organization where the research was conducted)

☒

UNRESTRICTED

Validated by



(AUTHOR'S SIGNATURE)



(SUPERVISOR'S SIGNATURE)

Permanent Address:

MOHAMMAD KHAIRUL IKHWAN BIN ABDUL
BASIT
MDLD 4336 LORONG 1 TAMAN GABUNGAN,
91100 LAHAD DATU, SABAH

Date: 23/6/2025

Date: 23/6/2025

Note * Thesis refers to PhD, Master, and Bachelor Degree

** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

TABLE OF CONTENT

ACKNOWLEDGEMENT.....	v
LIST OF FIGURES	vi
LIST OF TABLES.....	viii
LIST OF ABBREVIATIONS	ix
ABSTRACT.....	x
ABSTRAK	xi
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives.....	2
1.4 Methodology	3
1.5 Scope	4
1.6 Significance of Project	4
1.7 Project Schedule	5
1.8 Expected Outcome	7
1.9 Project Outline.....	7
1.10 Summary	8
CHAPTER 2: LITERATURE REVIEW	9
2.1 Introduction	9
2.2 Recommendation Systems Overview.....	9
2.2.1 Content-Based Filtering.....	10
2.2.2 Collaborative Filtering.....	11
2.2.3 Hybrid Approach.....	12

2.3 Related Work	12
2.3.1 Recommender Systems for University Elective Course Recommendation.....	12
2.3.2 Fuzzy-Based Recommendation System for University Major Selection	14
2.3.3 A Recommender System for Private University Admission in Bangladesh.....	14
2.4 Comparison of Related Works.....	15
2.5 Brief Overview of Proposed System	16
2.6 Summary	17
CHAPTER 3 REQUIREMENTS ANALYSIS AND DESIGN.....	18
3.1 Introduction	18
3.2 Methodology Overview.....	18
3.3 Requirements Analysis	19
3.3.1 Method for gathering requirements	19
3.3.2 Survey Analysis	20
3.3.3 List of required requirements for the proposed system	25
3.4 System Design.....	25
3.4.1 Overview.....	25
3.4.2 Recommendation Algorithm Design	26
3.4.3 Context Diagram.....	30
3.4.4 Data Flow Diagram Level-0	31
3.4.5 Data Flow Diagram Level-1	32
3.4.6 Entity Relationship Diagram	33
3.4.7 Data Dictionary.....	35
3.4.8 Interface Design.....	39
3.5 Summary	45

CHAPTER 4: IMPLEMENTATION	46
4.1 Introduction	46
4.2 Implementation Tool	46
4.2.1 Software used for development	46
4.3 Programme Data Collection	47
4.4 Recommendation Algorithm Overview.....	48
4.5 Prototype	50
4.5.1 System hierarchy.....	50
4.5.2 Landing Page	52
4.5.3 User Registration	53
4.5.4 User Login	54
4.5.5 Select Qualification.....	55
4.5.6 Diploma Form.....	56
4.5.7 Matriculation Form	57
4.5.8 STPM Form	58
4.5.9 SPM Form.....	59
4.5.10 Interest Quiz.....	60
4.5.11 Recommendation Page.....	61
4.5.12 View Programme	62
4.5.13 Search Programme.....	63
4.5.14 Favourite Programme	64
4.6 Discussion	65
4.7 Summary	65
CHAPTER 5: TESTING	66
5.1 Introduction	66

5.2 Functionality Testing	66
5.2.1 Overview.....	66
5.2.2 List of Functionality Testing Result.....	66
5.3 Usability Testing.....	68
5.3.1 Overview.....	68
5.3.2 List of Usability Testing Result	68
5.4 User Acceptance Testing	69
5.4.1 List of User Acceptance Testing Results	70
5.5 Summary	74
CHAPTER 6: CONCLUSION & FUTURE WORKS.....	75
6.1 Introduction	75
6.2 Achievement.....	75
6.3 Limitation	76
6.4 Future Work.....	76
REFERENCES.....	78
APPENDIX A – Google Form Questionnaire Structure	80
APPENDIX B – Programme Data Collection	85
APPENDIX C – Interest Quiz Questions.....	87
APPENDIX D – Functional Testing Results.....	90
APPENDIX E – Usability Testing Results	100
APPENDIX F – Usability Testing & User Acceptance Testing Form.....	108
APPENDIX G – Link Hosted.....	111
APPENDIX H – Comments during Testing.....	112

ACKNOWLEDGEMENT

I am deeply grateful to everyone who supported me throughout the journey of this project. I would like to thank my supervisor, Abdul Rahman bin Mat, for their invaluable guidance, feedback, and encouragement, which helped shape the project into its current form. I extend my gratitude to my family and friends for their unwavering support and motivation during challenging times.

LIST OF FIGURES

Figure 1.1: The different phases of the Waterfall Model (Tutorials Point, 2024).....	3
Figure 1.2: FYP1 Timeline Gantt Chart.....	6
Figure 1.3: FYP2 Timeline Gantt Chart.....	6
Figure 2.1: Example of Content-Based Filtering (Ng, 2019).	10
Figure 2.2: Example of Collaborative-Based Filtering (Ng, 2019).	11
Figure 3.1: Distribution of responses to the difficulty of selecting a university programme.	21
Figure 3.2: Challenges faced when selecting a university programme	21
Figure 3.3: Sources currently used to gather information about university programmes.....	22
Figure 3.4: Usefulness of the university program recommendation system.....	23
Figure 3.5: Preferred method for providing user preferences.....	23
Figure 3.6: Additional features expected from the recommendation system.....	24
Figure 3.7: Likelihood of using a recommendation system to choose a university.....	24
Figure 3.8: Context Diagram	30
Figure 3.9: Data Flow Diagram Level 0	31
Figure 3.10: DFD Level 1 For Process 5.0 (Recommend Suitable Programmes).....	32
Figure 3.11: Entity Relationship Diagram	34
Figure 3.12: Landing Page.....	40
Figure 3.13: Login Page.....	41
Figure 3.14: Registration Page.....	41
Figure 3.15: Academic Qualification.....	42
Figure 3.16: Interest Page	42
Figure 3.17: Show Recommendation Page.....	43
Figure 3.18: View Programme Detail Page	43
Figure 3.19: Favourite List page.....	44
Figure 3.20: View All Programmes Page.....	44
Figure 4.1: Sample of Entry Requirements used in the database	48
Figure 4.2: Recommendation Algorithm Overview.....	49
Figure 4.3: Landing Page.....	52
Figure 4.4: User Registration Page	53

Figure 4.5: Login Page.....	54
Figure 4.6: Select Qualification Page	55
Figure 4.7: Diploma Form Page.....	56
Figure 4.8: Matriculation Form.....	57
Figure 4.9: STPM Form Page	58
Figure 4.10: SPM Form Page.....	59
Figure 4.11: Interest Quiz Page.....	60
Figure 4.12: Recommendation Page	61
Figure 4.13: View Programme Page	62
Figure 4.14: Search Programme Page.....	63
Figure 4.15: View Favorited Programme Page	64
Figure 5.1: User Registration & Login UAT Results.....	70
Figure 5.2: Academic Qualification UAT Results	71
Figure 5.3: Interest Quiz UAT Results.....	71
Figure 5.4: Recommendation of Programme UAT Results	72
Figure 5.5: Search & Filtering Function UAT Results	72
Figure 5.6: Favourite Function UAT Results	73
Figure 5.7: Overall User Acceptance	74

LIST OF TABLES

Table 1.1: Final Year Project 1 (FYP 1) Timeline.....	5
Table 1.2: Final Year Project 2 (FYP 2) Timeline.....	5
Table 2.1: Comparison of Related Works with Proposed System	16
Table 3.1: Explanation of RIASEC Traits.....	28
Table 3.2: Example of RIASEC Profile.....	28
Table 3.3: List of Entities.....	33
Table 3.4: User Entity	35
Table 3.5: UserData Entity.....	35
Table 3.6: UserPreference Entity	36
Table 3.7: University Entity	36
Table 3.8: Programme Entity	36
Table 3.9: EntryRequirement Entity	37
Table 3.10: FavouriteProgramme Entity	37
Table 3.11: Tag Entity	38
Table 3.12: DegreeField Entity	38
Table 3.13: TagDegreeField Entity	39
Table 3.14: Question Entity	39
Table 4.1: Software Used for Developmnet.....	46
Table 5.1: Recommendation Module Functional Testing Results	67
Table 5.2: Recommendation Module Usability Testing.....	68
Table 6.1: Project Objective and Achievement.....	75

LIST OF ABBREVIATIONS

Abbreviation	Meaning
ALS	Alternating Least Squares
ERD	Entity-Relationship Diagram
FES	Fuzzy Expert System
FYP	Final Year Project
GPA	Grade Point Average
HSC	Higher Secondary Certificate
NEC	National Education Code
RIASEC	Realistic, Investigative, Artistic, Social, Enterprising, and Conventional
SPM	Sijil Pelajaran Malaysia
SSC	Secondary School Certificate
STPM	Sijil Tinggi Persekolahan Malaysia
UI	User Interface

ABSTRACT

Selecting a suitable university programme is a critical decision for students, especially in specialized fields such as computer science. This thesis presents UniReco, a web-based recommendation system developed to assist students in identifying appropriate computer science programmes offered by Malaysian public universities. The system analyses users' academic qualifications, interests, and personality traits to generate personalized recommendations. Data for recommendation is collected through a structured questionnaire that captures user preferences and maps personality traits using the RIASEC model. A hybrid recommendation approach is applied, combining content-based filtering for interest alignment with rule-based filtering for academic eligibility. Programme recommendations are determined by comparing the user profile against predefined clusters of computer science subfields, such as Software Engineering and Multimedia, and filtering results based on university entry requirements. The system is implemented as a prototype and evaluated through functionality and usability testing, focusing on its ability to provide relevant suggestions based on user input.

ABSTRAK

Pemilihan program universiti yang sesuai merupakan keputusan penting bagi pelajar, terutamanya dalam bidang pengkhususan seperti sains komputer. Tesis ini membentangkan UniReco, sebuah sistem cadangan berasaskan web yang dibangunkan untuk membantu pelajar mengenal pasti program sains komputer yang bersesuaian yang ditawarkan oleh universiti awam di Malaysia. Sistem ini menganalisis kelayakan akademik, minat, dan ciri personaliti pengguna bagi menghasilkan cadangan yang diperibadikan. Data untuk cadangan dikumpulkan melalui borang soal selidik berstruktur yang merekodkan keutamaan pengguna serta memetakan ciri personaliti berdasarkan model RIASEC. Pendekatan cadangan hibrid digunakan, yang menggabungkan penapisan berasaskan kandungan untuk penjajaran minat dengan penapisan berasaskan peraturan bagi menilai kelayakan akademik. Cadangan program ditentukan dengan membandingkan profil pengguna terhadap kluster bidang pengkhususan sains komputer yang telah ditetapkan, seperti Kejuruteraan Perisian dan Multimedia, serta menapis keputusan berdasarkan syarat kemasukan universiti. Sistem ini dibangunkan sebagai prototaip dan dinilai dari segi fungsi dan kebolegunaan, dengan tumpuan terhadap keupayaannya dalam memberikan cadangan yang relevan berdasarkan input pengguna.

CHAPTER 1: INTRODUCTION

1.1 Introduction

Pursuing higher education is a pivotal milestone for many individuals, as it lays the foundation for their future careers and personal growth. However, choosing the right university programme is often a challenging and overwhelming decision, especially in Malaysia, where students are faced with a wide range of options. This decision is especially important because selecting a programme that does not match their interests can lead to academic struggles and dissatisfaction. A study in Spain found that students who were not enrolled in their preferred field of study were significantly more likely to drop out (Lassibille & Navarro Gómez, 2008). Similarly, in Malaysia, a recent study on Technical and Vocational Education and Training (TVET) students revealed that 38 percent of those who dropped out cited a lack of interest in their chosen programme as the main reason (Rodzi, Basri, & Nopiah, 2024). This issue is particularly evident in fields like computer science, which consist of many specialized subfields. Without proper guidance or exposure to these options, students may struggle to choose a programme that truly fits their interests and qualifications.

Furthermore, the lack of sufficient information about the distinct characteristics of different programmes can lead to confusion, dissatisfaction, and regret later in their academic journey. According to a global survey conducted in 2017, 87% of students choose the wrong course or college due to a lack of information (IndustryLeadersMagazine, 2017). This statistic underscores the widespread issue that many students find themselves in unsuitable academic programmes because they do not have access to the necessary guidance to make informed decisions.

To address this challenge, this project proposes a web-based recommendation system that will recommend suitable programmes from multiple universities. While the system is designed to recommend university programmes, this project will focus specifically on computer science programmes to simplify its scope and ensure detailed recommendations. By analyzing students' interests and academic qualifications, the system aims to offer personalized suggestions, bridging the gap between student aspirations and the diverse subfields within computer science.

1.2 Problem Statement

Many individuals struggle to choose the right degree programme in the right university due to the overwhelming number of options available, particularly among Malaysians seeking to further their studies. This decision-making process is often complicated by a lack of awareness about the various degree programmes, their content, and how they align with not just their qualifications, but also their individual interests, strengths, and career goals. For example, fields like computer science offer a wide range of specializations, such as software engineering, artificial intelligence, cybersecurity, and more. This diversity can leave students feeling confused and uncertain about which path best suits their aspirations. Additionally, students often face challenges in determining which programmes they are eligible for based on their academic qualifications. Entry requirements vary across universities and programmes, and many students struggle to navigate these requirements effectively. Without proper guidance, they may waste time considering programmes they do not qualify for or overlook suitable alternatives that match both their academic background and interests. The limited access to accurate and detailed information about programmes and universities further aggravates this problem, leaving many students feeling ill-equipped to make informed decisions. As a result, students may enroll in programmes that do not match their interests or long-term goals. This mismatch can lead to dissatisfaction, a decline in academic performance, and in some cases, students discontinuing their studies altogether.

1.3 Objectives

The project aims to develop a recommendation system that effectively matches students with suitable computer science programmes based on their interests and qualifications. Key objectives to achieve this aim include:

- To analyze the range of Computer Science domains offered by Public Universities in Malaysia to define the scope of the recommendation content.
- To design a recommendation algorithm that can predict a list of suitable programmes for the user based on their interests and qualifications.
- To build a functional web application featuring core components such as user sign-in, structured forms or questionnaires, programme recommendations utilizing the recommendation algorithm, and a favorites list for saved options.
- To evaluate the functionality of the system through testing and feedback.

1.4 Methodology

This project will use the Waterfall model as its development methodology. The Waterfall model is a step-by-step process that follows specific stages: requirements gathering, design, implementation, testing, and maintenance (Lutkevich & Lewis, 2024). Each stage must be completed before moving to the next.

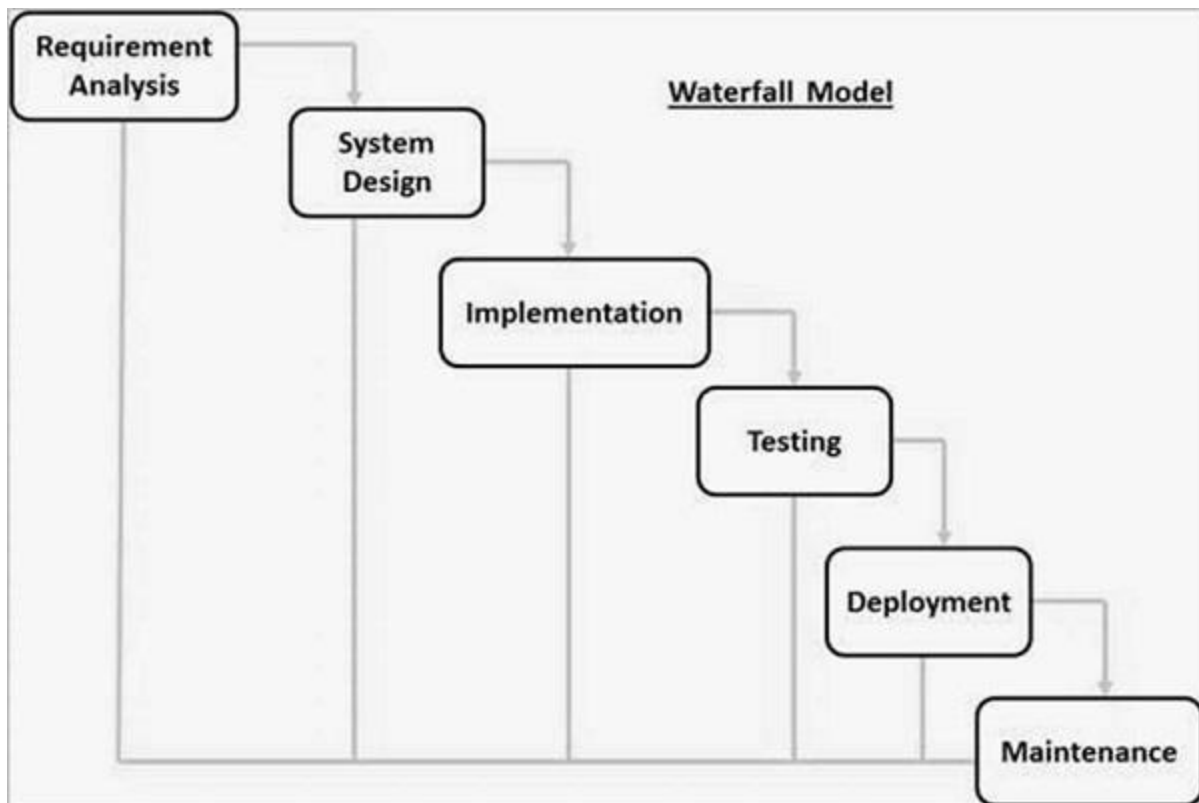


Figure 1.1: The different phases of the Waterfall Model (Tutorials Point, 2024).

Figure 1.1 shows the different phases of the Waterfall Model (Tutorials Point, 2024). Following this model, the development will be divided into several phases. In the Requirements Analysis phase, data will be collected from two sources. The first will involve gathering information about computer science programmes and their attributes to support the design of the recommendation algorithm. The second will involve conducting surveys with users to understand their needs and preferences for the system. This data will be used to define the system's main features. During the System Design phase, the system structure will be planned, the user interface

prototype will be created, the database will be designed, and the recommendation algorithm will be drafted based on the collected data. The Implementation phase will involve coding the system, connecting it with the database, and deploying it to a hosting environment. The final stage is Testing and Deployment, where the system will go through functional and user testing, along with debugging, to ensure it is working properly before it is presented.

1.5 Scope

The project will prioritize developing a web-based application with the core features that enable students to do the following:

- Register and login to their account,
- Answer a questionnaire to collect user interests and qualifications and attach them to their account,
- Get recommended programmes from multiple universities based on their interests
- Able to search and browse the recommendations and outside of it.
- Allow users to save the programmes for later viewing

To maintain a manageable scope and ensure effective development, the following factors will guide the project's design:

- Only bachelor's degree programmes in computer science will be recommended.
- The system will include programmes from Malaysian public universities only.
- The recommendation algorithm will utilize a limited set of attributes (approximately 5-10) to categorize programmes.

1.6 Significance of Project

This project aims to simplify the process of selecting a university programme, particularly in specialized fields like computer science. Many students struggle to make informed decisions due to limited exposure to programme details and career outcomes, often leading to regret and misalignment with personal goals. By developing a recommendation system focused on computer science programmes at Malaysian public universities, the project seeks to bridge this gap and help students explore options aligned with their interests and aspirations.

1.7 Project Schedule

Tables 1.1 and 1.2 present the project activities and durations based on the Waterfall model, while Figures 1.2 and 1.3 illustrate the overall project timeline and associated tasks throughout the final year project duration.

Table 1.1: Final Year Project 1 (FYP 1) Timeline

Phase	Activity	Time Duration
Literature Review	• Research into Existing systems • Research on Recommendations systems • FYP1 Report Introduction and Literature Review Submission	2 weeks
Data Collection	• Research and compile computer science programmes from Malaysian Public Universities • Research computer science programmes • Questionnaire distribution for requirements analysis	2 weeks
Requirements Analysis	• Questionnaire Analysis • FYP1 Requirements Analysis and Design Submission	2 weeks
System Design	• Prototyping User Interface through Figma • Web-Based System Design and Modeling • Database Design • Recommendation Algorithm Design • Submission of FYP 1 Report	2 weeks

Table 1.2: Final Year Project 2 (FYP 2) Timeline

Phase	Activity	Time Duration
Implementation	• Full coding implementation in local environment • Database integration and setup • FYP2 Chapter 4 Submission	7 weeks
System Testing and Deployment	• System Deployment to Hosting • Functional Testing • Usability and User Acceptance Testing • FYP2 Chapter 5 and 6 Submission	3 weeks

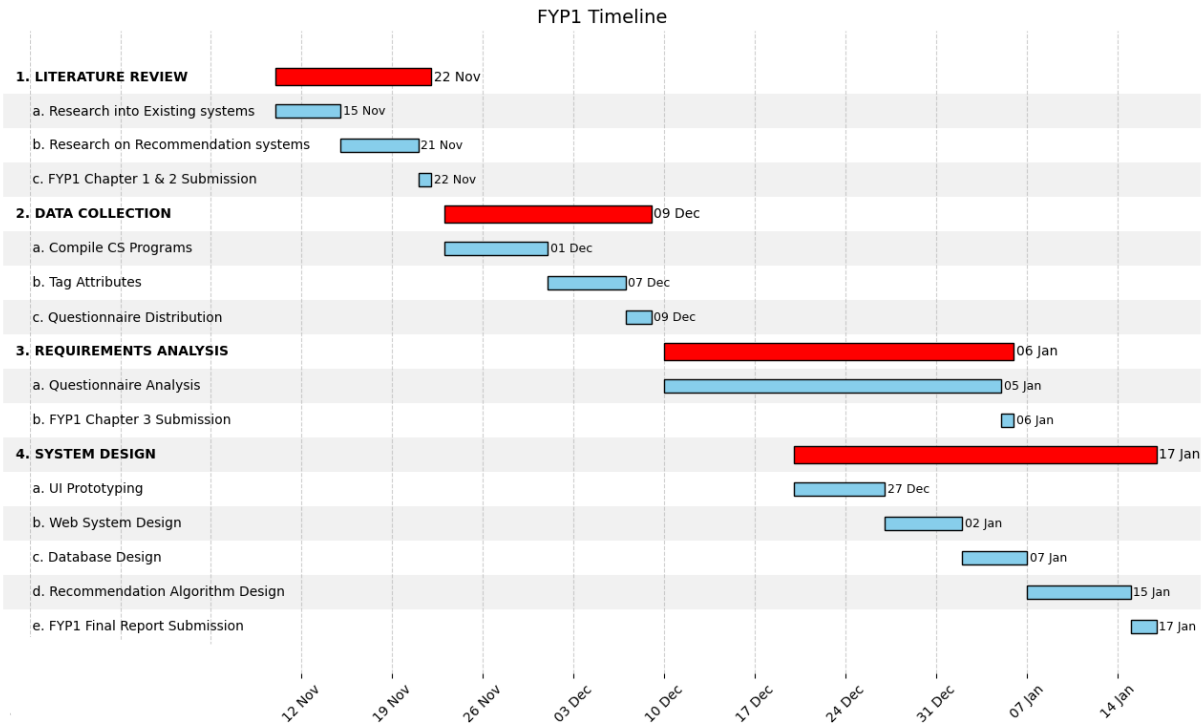


Figure 1.2: FYP1 Timeline Gantt Chart.

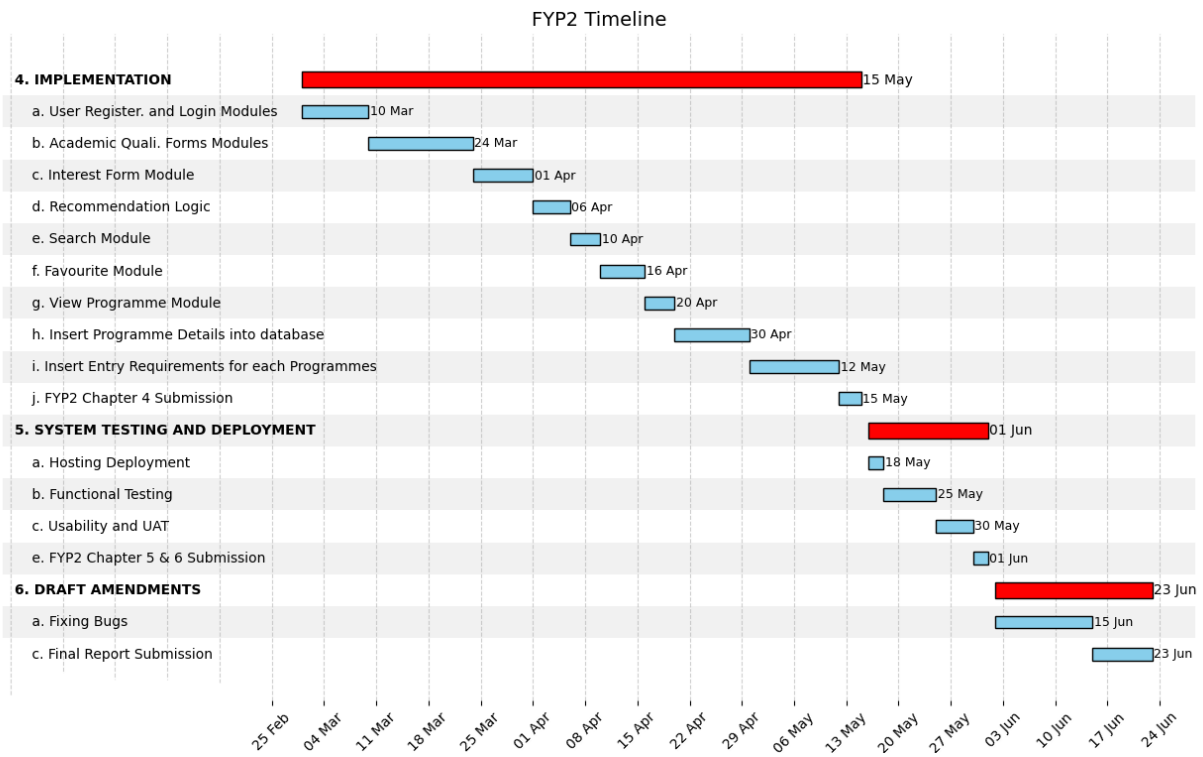


Figure 1.3: FYP2 Timeline Gantt Chart.

1.8 Expected Outcome

This project aims to deliver a web-based application that recommends suitable computer science programmes across various universities, tailored to students' interests. The recommendation system will employ an algorithm that analyzes user interests based on responses to a structured questionnaire completed after account registration. Users will be able to search and browse recommended programmes, view programme details, and save programmes to a favorites list for their future reference.

1.9 Project Outline

Chapter 1: Introduction

This chapter introduces the project by outlining the problem, objectives, scope, methodology, project timeline, project significance and outcome.

Chapter 2: Literature Review

This chapter reviews various types of recommendation systems, exploring their applications and how they relate to the proposed system. It also contrasts existing systems, highlighting their strengths and weaknesses, and discusses research findings that help frame the proposed system's design and functionality.

Chapter 3: Requirements Analysis and Design

This chapter describes the development methodology, focusing on the requirements analysis phase. It includes insights from user surveys, functional requirements, and the system's data flow design. It also presents data flow diagrams, entity relationship diagrams, and other illustrations to demonstrate how data flows in the system.

Chapter 4: Implementation

This chapter discusses the implementation process of the web-based recommendation system, covering the development of the recommendation algorithm, system coding, database integration, and user interface design.

Chapter 5: Testing

This chapter covers the software testing phase, outlining the types of testing performed to ensure the system meets its requirements. It includes functional testing, usability testing, and user acceptance and discusses the system's functionality, performance, and any issues encountered during the testing process.

Chapter 6: Conclusion & Future Works

This chapter summarizes the project's contributions and limitations, reflecting on whether the objectives were met. It also discusses potential future improvements and enhancements.

1.10 Summary

This chapter introduces the project, which focuses on developing a web-based recommendation system to help students select suitable computer science programmes in Malaysia. The project aims to address the challenges faced by students in choosing the right programme, particularly within the diverse field of computer science, by providing personalized recommendations based on their qualifications and interests. The objectives of the project include analyzing the available computer science programmes, designing a recommendation algorithm, building a functional web application, and evaluating its effectiveness. This chapter also introduces a brief methodology used in the project, the project timeline, scope of the project and expected outcome.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Recommendation systems are essential tools across many fields, helping users make informed decisions by offering personalized suggestions based on their preferences and behavior. These software-based techniques support decision-making tasks such as choosing products, music, or articles (Ricci et al., 2015). By analyzing large datasets to identify patterns, they help users navigate overwhelming choices in areas like e-commerce, entertainment, healthcare, and education. Techniques such as content-based filtering, collaborative filtering, and hybrid approaches are commonly employed to enhance user experience and decision-making efficiency.

Recently, their application in education has gained significant attention, particularly in guiding students toward suitable fields of study, majors, or courses. This chapter reviews existing educational recommendation systems, evaluating their design, methodologies, and effectiveness.

The scope includes systems utilizing content-based, collaborative, and hybrid methods. The review explores their underlying principles, strengths, and limitations to provide a comprehensive understanding of their educational applicability. By examining current research and comparing different systems, this chapter identifies best practices and gaps, laying the foundation for the proposed system aimed at recommending computer science programmes in Malaysian universities.

2.2 Recommendation Systems Overview

Recommendation systems are software applications designed to provide users with personalized suggestions based on their preferences, behaviors, or interactions (Ricci et al., 2015). These systems are widely applied across industries such as e-commerce, entertainment, and education, where they assist users in making decisions such as selecting products, listening to music, or reading content. In the context of education, recommendation systems help students make informed choices about courses, fields of study, or educational institutions by analyzing relevant data. This section introduces the main types of recommendation systems, based on the classification presented in the “Recommender Systems Handbook” by Francesco Ricci, Lior

Rokach, and Bracha Shapira (2015), which provides a comprehensive overview of the methodologies and approaches used in this field.

2.2.1 Content-Based Filtering

Content-based recommendation systems use the features of items previously interacted with by a user to build a profile of their preferences (Ricci, Rokach, & Shapira, 2021). Recommendations are generated by matching the attributes of new items to this profile, producing relevance scores that indicate how well the items align with the user's interests. For example, in a movie recommendation system, this approach might suggest films of the same genre, director, or actors as movies the user has previously enjoyed. Figure 2.1 shows an example of a movie recommendation system that uses content-based filtering.

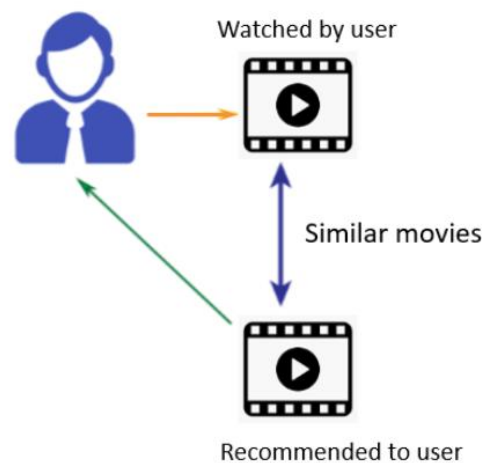


Figure 2.1: Example of Content-Based Filtering (Ng, 2019).

In education, this approach can assist students in discovering courses, programmes, or institutions by analyzing attributes such as course content, prerequisites, or skill outcomes. For instance, a system could recommend programming courses to a student with a strong interest in coding, based on their past course selections or stated preferences.

While content-based systems work effectively in well-defined domains, they have notable limitations. They often struggle with a lack of diversity, as recommendations tend to focus on items like those already liked, potentially overlooking diverse yet relevant options. Additionally, when

user interaction history is limited, such as for new users, the system may not have enough information to generate meaningful or accurate recommendations, reducing its effectiveness.

2.2.2 Collaborative Filtering

Collaborative filtering utilizes the preferences and feedback of multiple users to generate recommendations, focusing on patterns in user-item interactions (Ricci, Rokach, & Shapira, 2021). Unlike content-based methods, it relies on the assumption that users who have rated items similarly in the past are likely to have similar preferences in the future. This approach can be implemented in two primary ways: user-based and item-based filtering.

In user-based collaborative filtering, the system identifies users with similar rating patterns and recommends items highly rated by these peers. Conversely, item-based collaborative filtering analyzes the relationships between items by observing how they are rated by users, recommending items with similar rating trends. For example, in a movie recommendation system, if User A and User B both rate several action movies highly, and User B also rates a specific comedy movie highly, the system may recommend that comedy movie to User A.

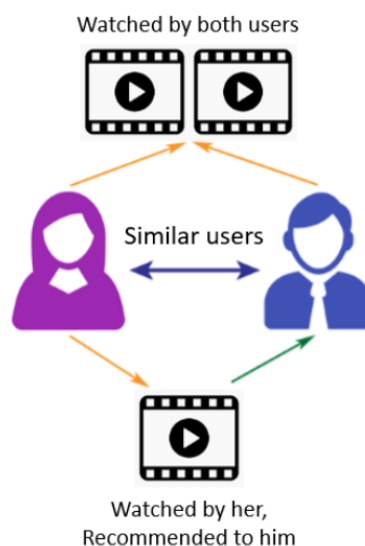


Figure 2.2: Example of Collaborative-Based Filtering (Ng, 2019).

In education, collaborative filtering can recommend courses or programmes based on peer evaluations, leveraging the collective experiences of students with similar academic interests or

goals. This approach enhances the personalization and relevance of educational recommendations, creating a more engaging user experience.

However, a key limitation of collaborative filtering is its need for large amounts of user data to produce accurate results. In educational context, this is difficult because students usually enrol in only one programme, giving little data to work with. There is not enough overlap between users to identify patterns, making the method less effective. Without sufficient data, the recommendations can be inaccurate or irrelevant.

2.2.3 Hybrid Approach

A hybrid recommender system combines collaborative filtering and content-based methods, leveraging the strengths of both to create a more comprehensive and effective recommendation solution (NVIDIA, n.d.). It addresses limitations like the cold start problem and sparsity by integrating user behavior with item attributes, improving accuracy and personalization. However, hybrid systems are computationally intensive and more complex, requiring significant processing power to analyze multiple data streams. Despite these challenges, they offer diverse and context-aware recommendations that align better with user preferences.

2.3 Related Work

Three systems were reviewed, that consists of the different types of recommendation systems reviewed previously. The review examines these systems' methodologies, evaluating the techniques used to recommend the outputs of the system respectively. By analyzing these studies, this review identifies key approaches, strengths, and limitations, which provide valuable insights for the development of a recommendation system tailored to university programme selection.

2.3.1 System 1: Recommender Systems for University Elective Course Recommendation

The study by Bhumichitr et al. (2017) developed a recommendation system to assist students in selecting suitable elective university courses by analyzing academic records and course-taking patterns. It also aimed to help registrar predict student enrollment to support course planning. The system employed Collaborative Filtering with Pearson Correlation and Alternating

Least Squares (ALS) to compare the effectiveness of different algorithms in generating course recommendations. The methodology of the study involves several key processes:

1. **Data Gathering:** Enrollment records from the university's registration database were collected, including data on 3,614 students and 52 courses over 8 years.
2. **Data Preprocessing:** The collected data was cleaned to remove unrelated information, such as courses no longer offered.
3. **Course Template Creation (User Profiling):** Each student's academic record was converted into a binary template to reflect their course history.
4. **Similarity Discovery:** The system used Pearson Correlation to calculate the similarity between students based on their templates.
5. **Recommendation Model Creation:** A model was built using similarity scores and course templates to predict suitable courses.
6. **Optimization:** The ALS algorithm was applied to optimize the interaction matrix and enhance prediction accuracy.
7. **Course Recommendation:** Recommendations were generated by identifying courses taken by similar students and using ALS to suggest additional electives.

The study evaluated the performance of the two algorithms using academic records from 3,614 students over an 8-year period. Recommendations were assessed by comparing them with actual elective course enrollments, using a threshold-based accuracy measure. Results showed that the ALS algorithm performed better, achieving 86% accuracy compared to 67% for Collaborative Filtering.

While the findings were promising, the system had limitations, primarily due to its reliance on enrollment records alone. The study suggests that incorporating additional data such as grades and personal attributes could enhance accuracy. Future improvements may include testing other algorithms and developing more sophisticated models to better reflect student preferences and behaviors.

2.3.2 System 2: Fuzzy-Based Recommendation System for University Major Selection

The study by Alghamdi et al. (2019) proposed a fuzzy-based recommendation system designed to assist students in selecting a suitable university major by integrating academic performance and personal preferences. The system aimed to support students in making informed decisions while addressing the complexity and uncertainty involved in academic major selection. The methodology involved a multi-stage approach combining fuzzy logic and a cluster-based preference model:

1. **Fuzzy Expert System (FES) Development:** Academic inputs such as High School Average, General Ability Test, and Achievement Test were converted into fuzzy variables, and inference rules were applied to suggest suitable academic tracks (Science, Engineering, Medicine, Art).
2. **Preference Clustering:** Majors were grouped by content similarity. An online questionnaire was used to capture students' interests, and a distance measurement algorithm matched preferences to potential majors.
3. **Recommendation Filtering:** Final recommendations were generated by intersecting eligible academic programmes with those aligned to student interests.

The system was evaluated in a pilot involving fifteen prospective students. Results showed that 66% were satisfied with the system and 54% agreed with the recommended majors, suggesting early effectiveness of combining fuzzy logic with personalized input. The study concluded that integrating fuzzy logic with student preference profiling can improve decision support in major selection. Future work should focus on refining preference measurement, adapting the system to broader educational contexts, and conducting comparative studies to benchmark its effectiveness.

2.3.3 System 3: A Recommender System for Private University Admission in Bangladesh

The study by Mokarrama et al. (2020) introduced a content-based recommendation system to assist students in selecting suitable private universities in Bangladesh. The system aimed to simplify the university selection process by considering academic qualifications, financial constraints, and institutional rankings. The methodology followed a structured process combining data collection, preprocessing, and similarity-based recommendation:

1. **Data Collection:** Student data such as SSC GPA, HSC GPA, preferred divisions, subject interests, and tuition budget, was gathered via an input interface. University data such as subjects offered, GPA requirements, fees, and rankings was scraped from official websites.
2. **Data Storage:** Both datasets were stored in a real-time cloud database to ensure synchronization and continuous availability.
3. **Data Preprocessing:** Student and university features were normalized to a common scale (0 to 1) to ensure comparability across metrics.
4. **Similarity Calculation:** Cosine similarity was used to calculate matching scores between each student and the universities based on six features: SSC GPA, HSC GPA, total GPA, world ranking, Google rating, and tuition fees.
5. **Top-K Selection:** Universities were ranked based on similarity scores, and the top-K most relevant institutions were selected for each student.
6. **Recommendation Generation:** Each selected university was assigned a system rating by multiplying its similarity score by 5 (range: 0 to 5), and the final list was presented to the student.
7. **Result Evaluation:** Recommendations were evaluated using precision, recall, F1 score, specificity, and balanced accuracy based on feedback from 947 students.

2.4 Comparison of Related Works

After reviewing related works and existing systems, several gaps were identified. One notable gap is the lack of available systems specifically designed for recommending university degree programmes, as this area remains underexplored. Additionally, there is a significant lack of systems tailored to recommending university programmes in Malaysia. Furthermore, many existing systems do not utilize comprehensive metrics that incorporate both students' interests and academic qualifications to provide personalized recommendations. These gaps present a valuable opportunity for the development of this project. Table 2.1 shows the comparison of the related works to analyze the features needed for the proposed system.

Table 2.1: Comparison of Related Works with Proposed System

Recommendation System	System 1	System 2	System 3	Proposed System
System Input	Student academic records	High school average, test scores, online questionnaire on student preference	Student GPA, tuition budget, subject preferences, university rankings and ratings	Academic Qualifications, Personality and Interest,
Recommendation Objective	Elective courses for students	University major selection	Private university selection	University programme recommendation (Malaysia)
Methodology	Collaborative Filtering and ALS	Fuzzy Expert System (FES) and Cluster-Based Preferences Technique	Content-Based Filtering with Cosine Similarity	Content-Based Filtering with Rule-Based Filtering
Strength	Effective for identifying course patterns without needing domain-specific knowledge	Handles uncertainty and vagueness in student preferences	Matches students with universities based on multiple practical factors	Handles preferences, together with qualifications.
Weakness	Limited by reliance on course enrollment data only	Requires predefined fuzzy rules and limited dataset	Depends heavily on data availability and requires manual domain knowledge	Limited data and manual domain knowledge.

2.5 Brief Overview of Proposed System

The proposed system, “UniReco”, is a web-based recommendation platform designed to assist students in selecting suitable university programmes. It adopts a hybrid approach by combining content-based filtering with rule-based filtering to ensure that recommendations align with both user preferences and academic eligibility. Content-based filtering is used to match users with programme clusters based on their interests and personality traits, while rule-based filtering ensures that recommended programmes meet the user's academic qualifications through logical entry requirement checks.

This method is particularly effective given the limited availability of detailed programme data and the variability of qualifications in Malaysian universities. Instead of recommending individual programmes, the system groups options under broader Computer Science subfields, such as Multimedia or Software Engineering, to enhance clarity and relevance.

The system will include essential features such as user registration and login, a multi-step questionnaire for capturing academic background and preferences, a recommendation module that displays matched programme clusters with percentage fit, a catalogue to explore available programmes, and a bookmarking function for saving preferred options.

2.6 Summary

This chapter provided an overview of recommendation systems, focusing on their application in educational contexts such as course and programme recommendations. It discussed various techniques, including content-based filtering, collaborative filtering, and hybrid models, and examined their strengths and limitations. The chapter also reviewed existing systems in the field, highlighting gaps in providing personalized recommendations for university degree programmes, particularly within Malaysia. These insights set the foundation for the proposed system, which aims to enhance decision-making for upcoming students by offering tailored recommendations based on their academic qualifications and interests.

CHAPTER 3 REQUIREMENTS ANALYSIS AND DESIGN

3.1 Introduction

This chapter discusses the methodology employed in carrying out the development of the project. To achieve the primary objective of the project, a web application was designed to incorporate all the core features necessary for a university degree programme recommendation system. The chosen methodology for this project is the Waterfall Model, which provides a structured and systematic approach to software development.

The focus of this chapter is on the first two phases of the Waterfall Model: Requirements Analysis and System Design, as identified in Chapter 1. These phases are critical in laying the foundation for the subsequent development stages, ensuring that all necessary specifications are identified and properly planned.

3.2 Methodology Overview

This project adopts the Waterfall Model as the primary methodology to achieve the objectives of the university programme recommendation system. The Waterfall Model is a linear and sequential approach that divides the development process into distinct phases, each building upon the completion of the previous one. Each phase is crucial in ensuring the project's success, as it lays a strong foundation for the subsequent stages.

a. Requirements Analysis

Requirements Analysis marks the initial phase where extensive data collection is performed. This includes gathering information about computer science programmes and their attributes, which will inform the design of the recommendation algorithm. Additionally, user surveys are conducted to understand the preferences and requirements of potential users regarding the system's features and design. The collected data is thoroughly analyzed to define the system's functional and non-functional requirements, ensuring that the final product meets user needs effectively.

b. System Design

In the System Design phase, the overall structure of the system is developed based on the requirements identified earlier. This involves creating a user interface prototype to visualize how users will interact with the system and outlining the recommendation algorithm's logic. These design elements are carefully documented to serve as blueprints for the implementation phase.

c. Implementation

The Implementation phase will focus on converting the system's design into a functional prototype web application. During this phase, the application will be developed in accordance with the identified requirements, aiming to produce a working prototype that effectively demonstrates the core features and functionalities necessary to achieve the project objectives.

d. Testing and Deployment

Finally, the Testing and Deployment phase focuses on ensuring the system is reliable, user-friendly, and ready for presentation. This phase begins with Functional Testing, where each feature of the system is rigorously tested to confirm it operates according to the specified requirements. Following this, Usability testing and User acceptance testing is also conducted to evaluate the user experience, targeting potential users such as diploma holders and university students. After testing, the system undergoes final debugging and optimization to ensure it is fully operational. The system is then prepared for presentation, showcasing its functionality, design, and overall effectiveness in achieving the project's objectives.

3.3 Requirements Analysis

This section outlines the process of gathering and analyzing the requirements for the University Programme Recommendation System. It details the methods employed for requirements gathering, including surveys and research, and discusses how the collected data is analyzed to define the necessary features and functionalities of the system.

3.3.1 Method for gathering requirements

To design a recommendation system that aligns with user expectations, a survey form was used as the primary method for gathering requirements. Created with Google Forms, this survey

is an accessible tool to collect feedback from potential users, helping shape the system's features and functionality.

The survey aims to gather insights into what potential users believe the recommendation system should include. By understanding their needs and preferences, the system can be tailored to provide meaningful and relevant recommendations. The survey is distributed to two main groups, each offering unique perspectives:

- 1. Students Aspiring to Attend University:**

This group includes individuals preparing for higher education, such as Sijil Pelajaran Malaysia (SPM) graduates, Pre-University students (Matriculation, Foundation programmes, STPM graduates), and diploma holders. They provide valuable insights into the features and guidance they expect from a recommendation system to assist in their decision-making process.

- 2. Individuals Who Have Attended University:**

This group comprises degree holders and graduate students who have gone through the process of selecting a university programme. Their retrospective opinions highlight areas where the system could offer support, based on what they wish they had known or used during their decision-making journey.

3.3.2 Survey Analysis

A total of 26 respondents participated in the survey, providing valuable insights into their expectations and preferences for the university programme recommendation system. The following section analyzes key statistics from the survey results, which are visualized in the graphs. Refer to Appendix A for the contents of the google form.

Figure 3.1 shows that the majority of respondents (42.3%) rated the difficulty as 3, with 23.1% selecting 4 and 19.2% selecting 5. This suggests that most users find the process moderately to highly challenging when aligning their goals and interests with a university programme.

How difficult do you find the process of selecting a university program that aligns with your goals and interest?

26 responses

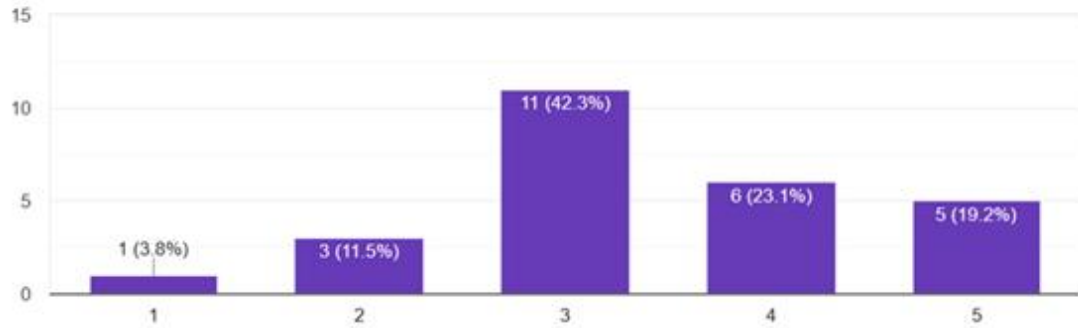


Figure 3.1: Distribution of responses to the difficulty of selecting a university programme.

Figure 3.2 shows that most respondents experienced multiple challenges, with the top three being uncertainty about which programme to choose, feeling overwhelmed, and struggling to understand which university offers the best programme for their needs. These issues highlight the complexity and confusion often faced during the programme selection process. The proposed system will aim to address these challenges by providing a structured and data-driven approach to programme selection, helping students make informed decisions based on their interests and academic preferences.

What challenges do you face when selecting a university program? (Select all that apply)

26 responses

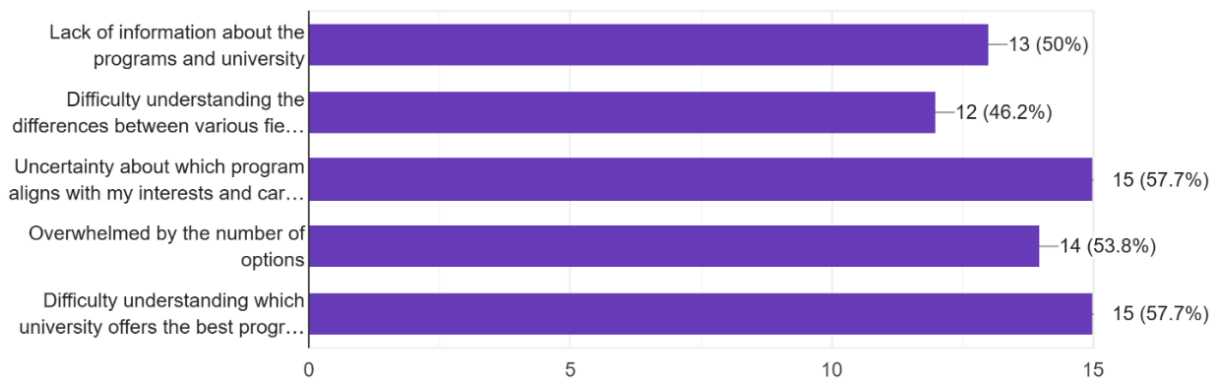


Figure 3.2: Challenges faced when selecting a university programme

Figure 3.3 shows that university websites and social media were the most used sources of information. This suggests a tendency to rely on familiar platforms, potentially leading to a limited perspective. Users may be restricted to the programmes they already know about, missing out on a broader range of options that could better align with their interests and goals. The proposed system aims to provide comprehensive and well-structured information, ensuring that users have access to a wider range of programme options, allowing them to make more informed and confident decisions.

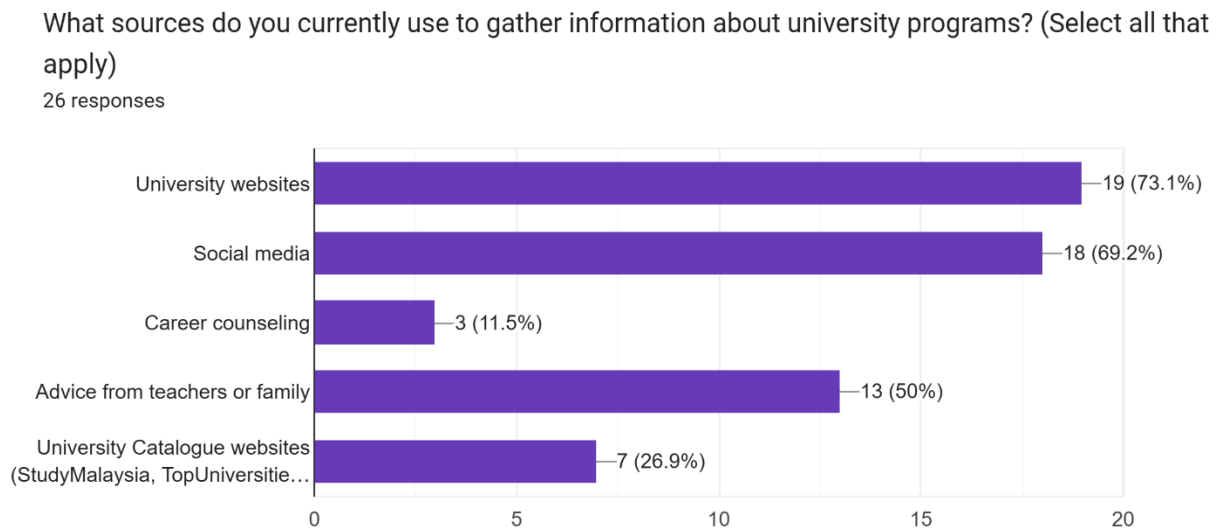


Figure 3.3: Sources currently used to gather information about university programmes

Figure 3.4 shows most respondents rated the system as highly useful, which indicates a strong interest and perceived value in having a recommendation system to assist in programme and university selection based on personal interests.

On a scale of 1 to 5, how useful would a recommendation system be for suggesting programs and universities based on your interests?

26 responses

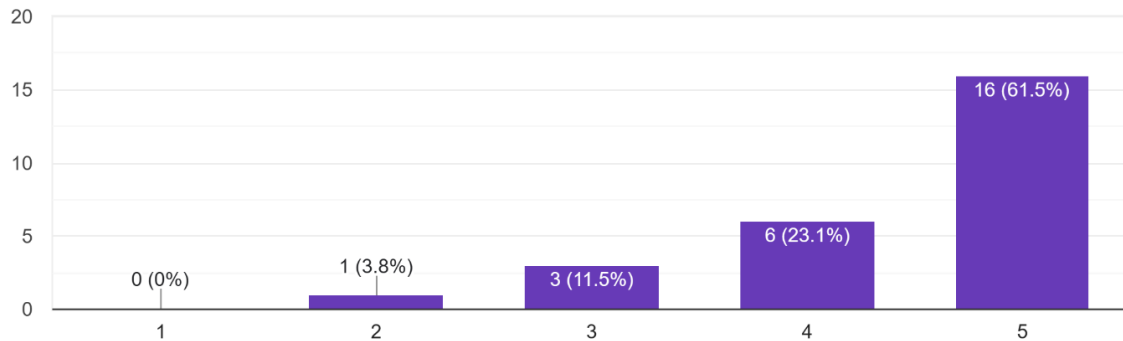


Figure 3.4: Usefulness of the university programme recommendation system

Figure 3.5 illustrates that most respondents favor a balanced approach that integrates both structured questionnaires and predefined options. This indicates that, to effectively meet user needs, the system will adopt a hybrid input method, ensuring both ease of use and seamless user experience.

How would you prefer to provide your interests and preferences to the recommendation system?

26 responses

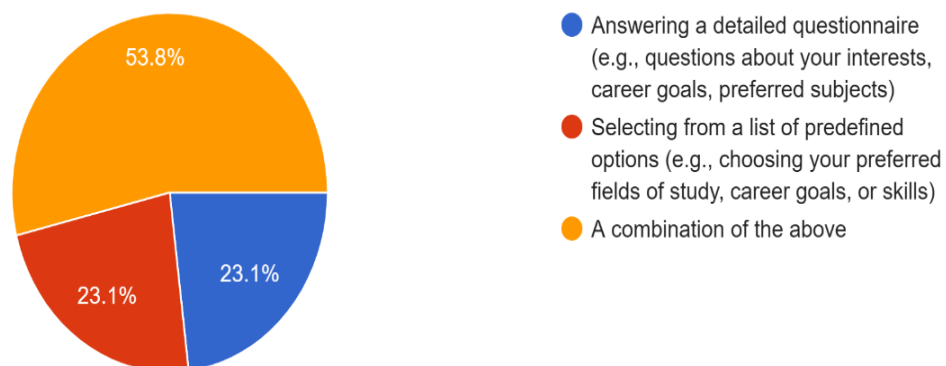


Figure 3.5: Preferred method for providing user preferences

Figure 3.6 shows that respondents prioritize features such as job-related information, an easy-to-use interface, and detailed programme descriptions. Search functionality and bookmarking options are also highlighted as important. The majority of respondents prefer almost all the options offered, indicating that most of these features will be considered in the system design. The majority of respondents prefer almost all the options offered, indicating that these features will be incorporated into the proposed system to enhance usability and assist users in making well-informed decisions.

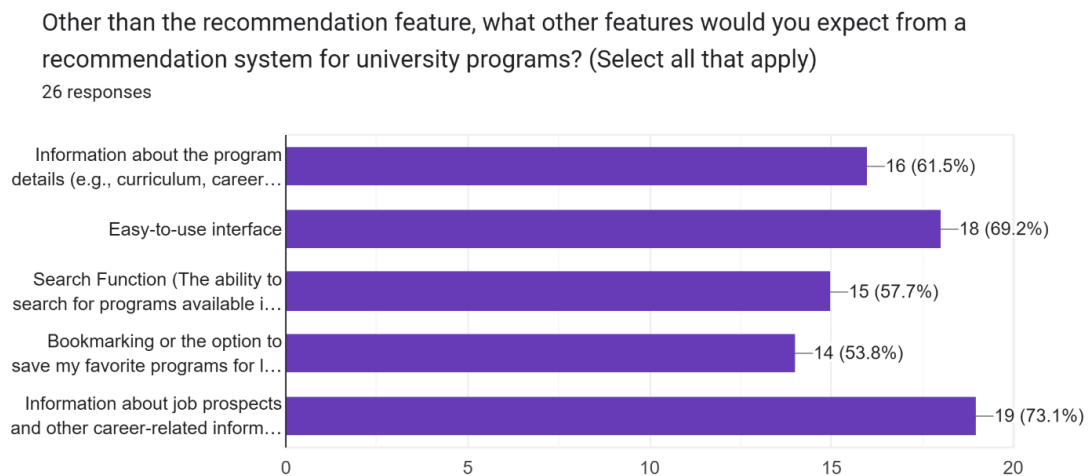


Figure 3.6: Additional features expected from the recommendation system.

Figure 3.7 shows that most respondents selected 4 and 5, indicating a strong likelihood of using the system to aid in their decision-making process.

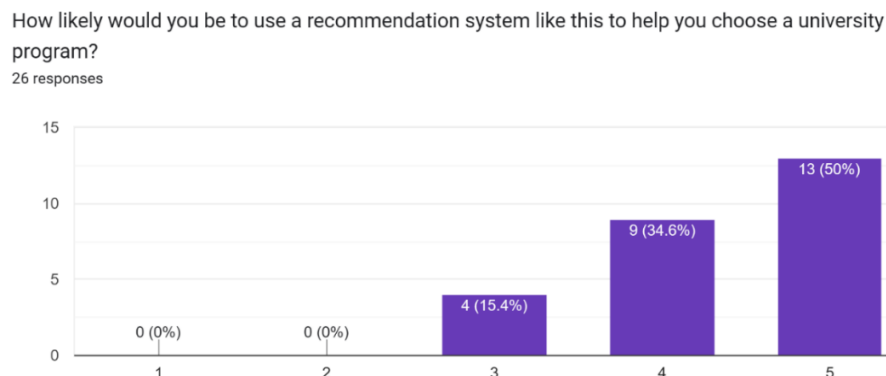


Figure 3.7: Likelihood of using a recommendation system to choose a university

3.3.3 List of required requirements for the proposed system

This section outlines the key functional requirements derived from the survey analysis to ensure the proposed system meets user expectations. The functional requirements are listed below:

1. User Registration and Login
 - Users can create an account and securely log in to access personalized features.
2. View All Universities and Programmes
 - Users can browse and explore a comprehensive list of universities and their respective programmes
3. Programme Recommendations
 - Questionnaire Interface: The system presents a questionnaire to users to collect information about their academic qualifications and interests.
 - Personalized Suggestions: Users receive tailored programme recommendations that align with their inputted preferences and interests.
4. Programme Information
 - Users can view detailed information about each programme, including descriptions, prerequisites, career opportunities, and other relevant details.
5. Favoriting Programmes
 - Users can bookmark programmes as favorites for easy access and comparison.

3.4 System Design

3.4.1 Overview

The Data Flow Diagram (DFD) is used to represent the flow of data within the system, providing a visual overview of how information moves between external entities and the system. The diagram includes three levels: the Context Level, which shows the system's interaction with external entities, Level 0 which outlines the main processes within the system and Level 1 which further breaks down these processes into more detailed sub-processes.

An Entity-Relationship Diagram (ERD) is used to model the relationships between key data entities within the system. This diagram is crucial for visualizing the database structure and understanding how different entities, such as users, programmes, and preferences, are interrelated.

A Data Dictionary is created to define and describe the data elements used within the system. It provides a detailed reference to ensure consistency in data naming conventions, types, and usage across the system, which is essential for proper database design and management.

The User Interface (UI) design focuses on creating an intuitive and user-friendly experience for the system. It considers the layout, navigation, and interaction flow to ensure users can easily engage with the system and efficiently complete their tasks.

Lastly, the recommendation logic defines the core algorithm that matches users to suitable university programmes based on their academic qualification and preferences. This logic is central to the system's functionality, ensuring accurate and personalized recommendations are generated for each user.

3.4.2 Recommendation Algorithm Design

3.4.2.1 Programme Data Collection

To construct a reliable and meaningful recommendation system, data will be systematically collected from three selected Malaysian public universities. For each of these institutions, all computer science-related programmes will be identified and analyzed. The information collected for each programme will include the following:

- Programme name
- Programme description
- Course outline
- Entry requirements, including qualifications through Sijil Pelajaran Malaysia (SPM), Matriculation, Sijil Tinggi Persekolahan Malaysia (STPM), and Diploma
- Career options associated with the programme
- Mode of study (full-time, part-time, online)
- University-specific details (location, facilities)

This data will form the basis of a structured knowledge database, which serves as the foundation of the recommendation engine. The programmes will be manually classified into distinct clusters, and each cluster will be associated with a predefined set of features. These

features will subsequently be used to match user preferences and filter out qualifications through the recommendation logic.

3.4.2.2 Recommendation Logic

The recommendation system will be designed to incorporate two primary components, which are academic qualifications and user interests and personality. These components will be combined to produce a weighted recommendation score, indicating the suitability of a given subgroup to an individual user.

A. Academic Qualification

The academic qualifications component evaluates whether the user meets the entry requirements for a given programme. If the user's academic background aligns with the stated requirements of a programme, the system will flag the programme as suitable. Only programmes for which the user is academically qualified will proceed to the next stage of recommendation.

B. Interest and Personality

This component is further divided into two subcomponents, which are personality assessment and interest matching:

a. Personality Assessment

The personality model utilized in this system is based on John Holland's RIASEC theory, which classifies individuals into six personality types, which are Realistic (R), Investigative (I), Artistic (A), Social (S), Enterprising (E), and Conventional (C). According to Holland's theory of career choice, individuals experience greater satisfaction and success when their work or educational environment aligns with their dominant personality traits (CareerKey, 2025). Table 3.1 presents an overview of each personality type along with its general characteristics and how it is applied within the context of this recommendation system to suggest suitable academic programmes.

Table 3.1: Explanation of RIASEC Traits

Type	General Trait	Programme Relevance
Realistic	Practical, hands-on, prefers structured tasks.	Likely to enjoy technical and practical fields that involve hardware, systems, or physical tools.
Investigative	Analytical, curious, enjoys problem-solving.	Attracted to fields that involve research, data analysis, or coding due to their logical and inquisitive nature.
Artistic	Creative, expressive, values design and innovation.	Prefers programmes that allow for creative media, visual design, and innovative thinking.
Social	Cooperative, supportive, good at communication.	Drawn to tech roles that involve collaboration, user interaction, or educational support.
Enterprising	Persuasive, leadership-driven, goal-oriented.	Suited for business-tech programmes involving project leadership, startups, or tech marketing.
Conventional	Organized, detail-focused, prefers routines.	Enjoys structured work in data, documentation, and system management.

Each subgroup of programmes will be manually profiled with an assumed RIASEC distribution, represented as a vector of weights corresponding to the six personality traits. These weights range from 1 to 5, where 1 indicates the weakest relevance and 5 indicates the strongest relevance to the programme cluster. Table 3.2 shows an example of such a vector for selected clusters:

Table 3.2: Example of RIASEC Profile

Programme	R	I	A	S	E	C
Multimedia	5	3	5	3	3	3
Networking	5	2	5	4	3	2
Software Engineering	3	1	4	4	2	2

Users will complete a personality assessment that generates their personal RIASEC profile in vector form. The system will calculate the similarity between the user's personality profile and each programme cluster's RIASEC vector. The higher the similarity score, the more aligned the user is with that cluster, and consequently, programmes within that cluster will be ranked higher in the recommendation output.

b. Interest Matching

Interest preferences will be collected through a series of questions designed to evaluate the user's inclination toward specific programme features. Each question will correspond to a programme feature that has been pre-tagged within the knowledge database. For example, if a user indicates a strong preference for hands-on lab work and a programme cluster emphasizes practical training, the subgroup will receive a higher score. The scoring system will accumulate points for each subgroup based on the user's answers. Subgroups that accumulate the highest scores will be ranked as more relevant to the user's interests.

The final recommendation output is derived by combining both the academic qualification status and the results from the interest and personality components. The combined score is expressed as a percentage that reflects the overall suitability of a programme subgroup to the user. Programmes clusters with higher combined scores will be recommended with higher priority.

3.4.3 Context Diagram

Figure 3.8 shows the context diagram that illustrates the scope and interactions of the University Programme Recommendation System. The system aims to assist students in identifying suitable university programmes based on their individual preferences and interests. The system interacts with two external entities: Users and an Administrator.

- **Users** provide input to the system, including registration/login details and their qualifications and interests based on the forms provided. Users also interact with the system to browse available programmes, view personalized recommendations generated by the system, and manage a list of their preferred programmes. The system, in turn, provides Users with lists of available programmes, tailored recommendations based on their input, and the functionality to save preferred programmes.
- **Administrator** or admin is responsible for maintaining the system's database of university programme information. This involves adding, removing, and modifying programme data.

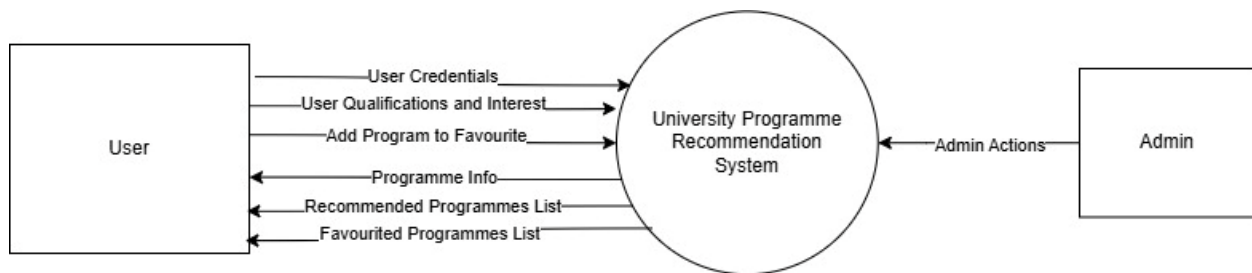


Figure 3.8: Context Diagram

3.4.4 Data Flow Diagram Level-0

Extending from the Context Diagram, the Data Flow Diagram (DFD) Level 0 details the primary processes and data flows within the system. It breaks down the system into its core functions, illustrating interactions and key data stores, Figure 3.9 shows the Data Flow Diagram Level 0 for the system.

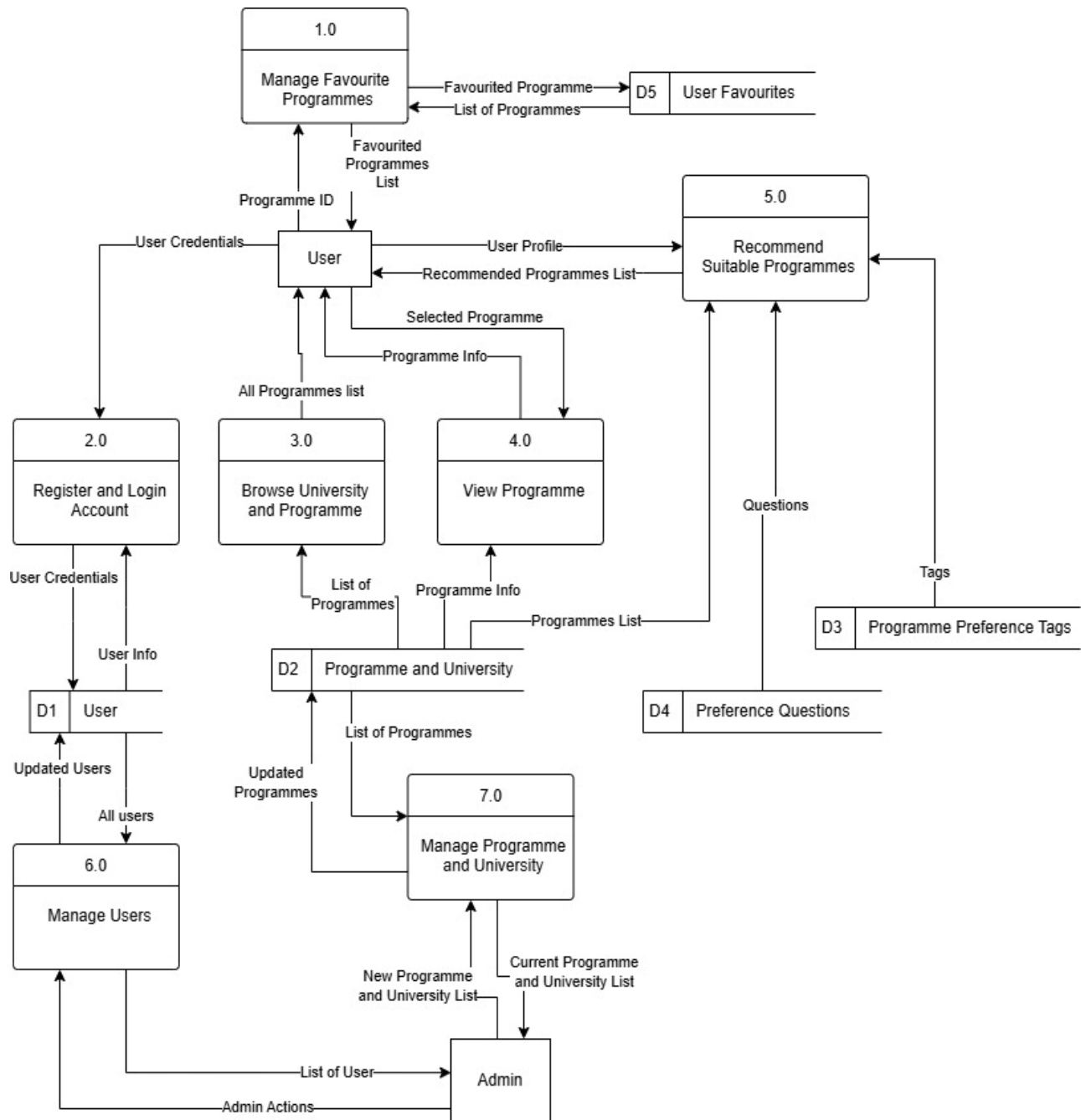


Figure 3.9: Data Flow Diagram Level 0

3.4.5 Data Flow Diagram Level-1

Expanding on the Data Flow Diagram Level 0, the Data Flow Diagram Level 1 details Process 5.0. It illustrates the internal data flow, showing how user inputs are processed, how programmes are computed based on preferences and qualifications, and how recommendations are presented. Figure 3.10 shows the Data Flow Diagram Level 1.

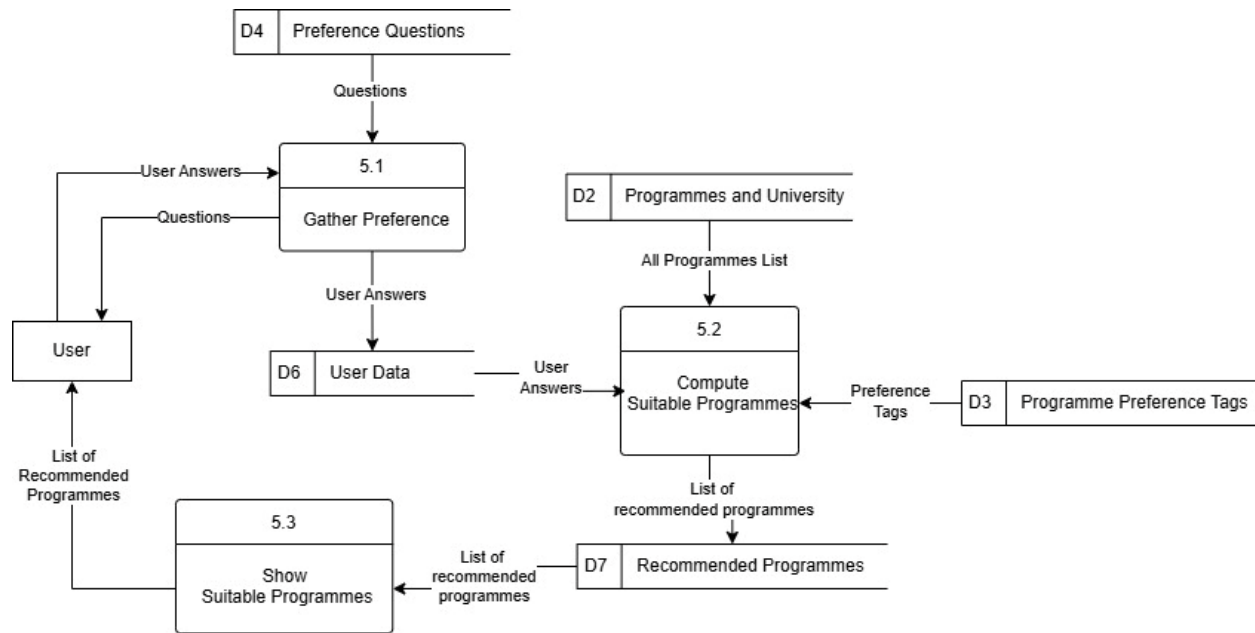


Figure 3.10: DFD Level 1 For Process 5.0 (Recommend Suitable Programmes)

3.4.6 Entity Relationship Diagram

Figure 3.11 presents the Entity Relationship Diagram (ERD) of the University Programme Recommendation System, which serves as the blueprint for modeling the system's database. Table 3.3 explains each of the entities.

Table 3.3: List of Entities

Entity Name	Description
User	The User entity represents the account of an individual using the system. It stores the account credentials required for registration and login.
UserData	The UserData entity holds additional user information that is not related to login or registration, such as personal details, academic background, and qualifications.
UserPreference	The UserPreference entity stores the preference data provided by the user, which will be used to generate personalized recommendations.
University	The University entity stores information about the universities offering the programmes. This includes details such as the university's name, website, and location.
Programme	The Programme entity represents the university programmes themselves. It stores information such as the programme's name, a descriptive text outlining the programme's content, and a foreign key referencing the entry requirements for that programme.
EntryRequirement	The EntryRequirement entity stores the specific academic qualifications required for admission to a particular programme.
FavouriteProgramme	The FavouriteProgramme entity represents the programmes that a user has marked as favorites. It establishes a many-to-many relationship between Users and Programmes, using foreign keys to both entities.
Tag	The Tag entity represents specific feature keywords or concepts that can describe a programme i.e. "software development", "artificial intelligence", "Realistic", "Investigative" and others. These tags are crucial for content-based recommendations, allowing the system to match user interests and personality profiles (from UserPreference) with relevant programme characteristics.
DegreeField	The DegreeField entity represents clusters or categories of programmes that share similar content, features, or align with specific academic or career areas. This allows for grouping programmes, even from different universities, that teach comparable subjects, such as Software Engineering or CyberSecurity.
TagDegreeField	The TagDegreeField entity links Tags to DegreeFields, forming a many-to-many relationship. It includes a relevancy score that indicates how strongly a Tag is associated with a specific programme subgroup. This score helps define RIASEC

	distributions and supports weighted interest matching in the recommendation process.
Question	The Question entity stores quiz questions used to assess user interests and personality. Each question is linked to a tag and includes its text, type, and order in the questionnaire.

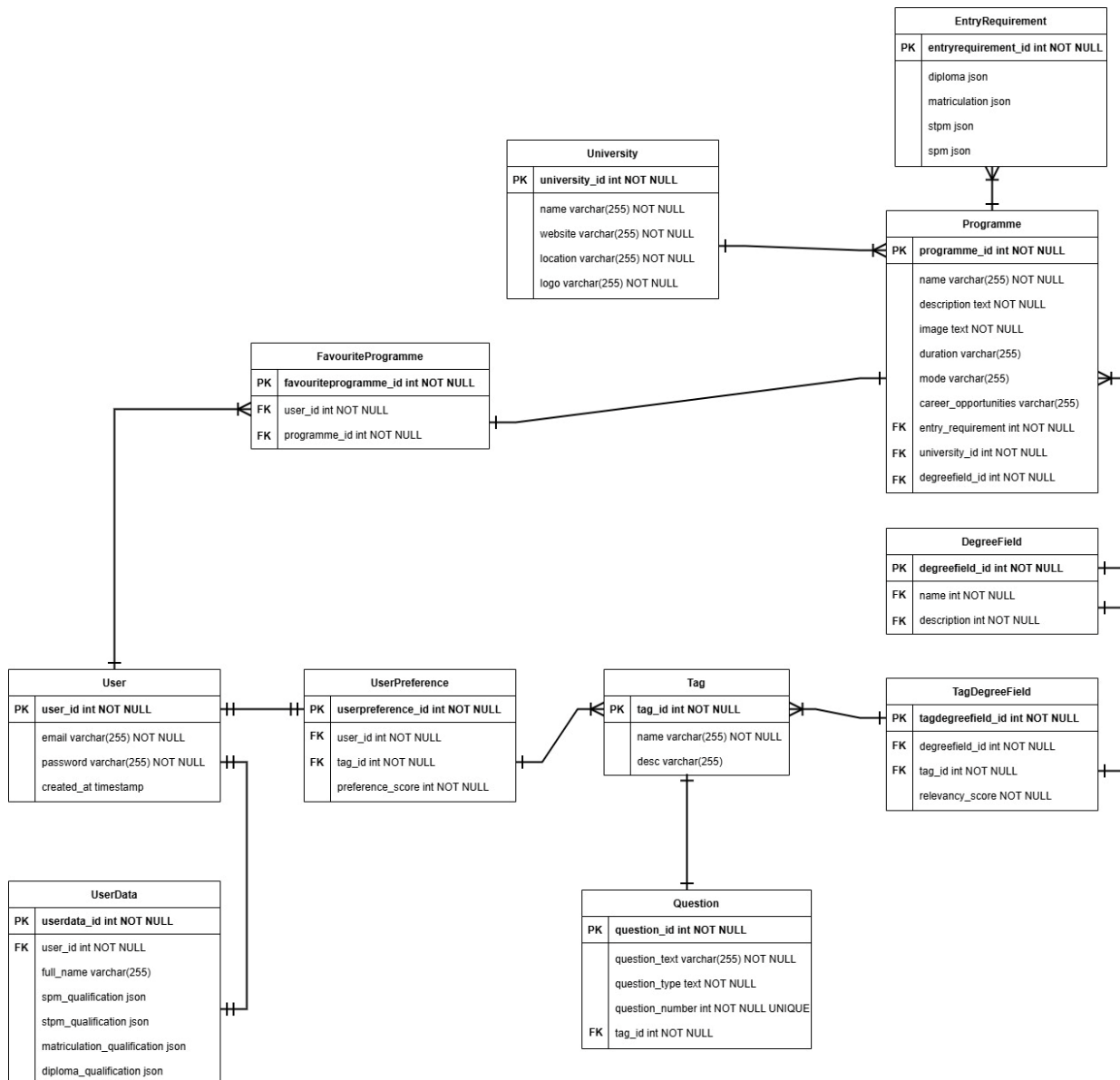


Figure 3.11: Entity Relationship Diagram

3.4.7 Data Dictionary

User

Table 3.4: User Entity

Field Name	Data Type	Constraints	Description
user_id	INT	PK, NOT NULL	Unique identifier for the user.
email	VARCHAR(255)	NOT NULL	Email address of the user.
password	VARCHAR(255)	NOT NULL	Hashed password of the user.
created_at	TIMESTAMP	-	Timestamp when the user was created.

UserData

Table 3.5: UserData Entity

Field Name	Data Type	Constraints	Description
userdata_id	INT	PK, NOT NULL	Unique identifier for user data.
user_id	INT	NOT NULL, FK (references User(user_id))	Foreign key linking to the user.
full_name	VARCHAR(255)	-	Full name of the user.
spm_qualification	JSON	-	JSON storing details of SPM qualifications.
matriculation_qualification	JSON	-	JSON storing details of matriculation qualifications.
stpm_qualification	JSON	-	JSON storing details of STPM qualifications.
diploma_qualification	JSON	-	JSON storing details of diploma qualifications.

UserPreference

Table 3.6: UserPreference Entity

Field Name	Data Type	Constraints	Description
userpreference_id	INT	PK, NOT NULL	Unique identifier for user preferences.
user_id	INT	NOT NULL, FK (references User(user_id))	Foreign key linking to the user.
tag_id	INT	NOT NULL, FK (references Tag(tag_id))	Foreign key linking to the tag.
preference_score	INT	NOT NULL	The preference score (1-10) for the specific tag.

University

Table 3.7: University Entity

Field Name	Data Type	Constraints	Description
university_id	INT	PK, NOT NULL	Unique identifier for the university.
name	VARCHAR(255)	NOT NULL	Name of the university.
website	VARCHAR(255)	NOT NULL	Website URL of the university.
location	VARCHAR(255)	NOT NULL	Location of the university.
logo	VARCHAR(255)	NOT NULL	Logo of the University

Programme

Table 3.8: Programme Entity

Field Name	Data Type	Constraints	Description
programme_id	INT	PK, NOT NULL	Unique identifier for the programme.
name	VARCHAR(255)	NOT NULL	Name of the programme.
description	TEXT	NOT NULL	Description of the programme.
image	TEXT	NOT NULL	Image of the programme

duration	VARCHAR(255)	-	Duration of the programme.
mode	TEXT	-	The mode of study for the programme offered.
university_id	INT	NOT NULL	Foreign key linking to the university the programme belongs to.
degreefield_id	INT	NOT NULL	Foreign key linking to a group where the programme fits in to.
entryrequirement_id	INT	NOT NULL, FK (references EntryRequirement (entryrequirement_id))	Foreign key linking to the entry requirements.

EntryRequirement

Table 3.9: EntryRequirement Entity

Field Name	Data Type	Constraints	Description
entryrequirement_id	INT	PK, NOT NULL	Unique identifier for entry requirements.
spm	JSON	-	JSON storing SPM entry requirements.
matriculation	JSON	-	JSON storing matriculation entry requirements.
stpm	JSON	-	JSON storing STPM entry requirements.
diploma	JSON	-	CGPA requirement for diploma entry.

FavouriteProgram

Table 3.10: FavouriteProgramme Entity

Field Name	Data Type	Constraints	Description
------------	-----------	-------------	-------------

favouriteprogramme_id	INT	PK, NOT NULL	Unique identifier for the favourite programme entry.
user_id	INT	NOT NULL, FK (references User(user_id))	Foreign key linking to the user.
programme_id	INT	NOT NULL, FK (references Programme(programme_id))	Foreign key linking to the programme.

Tag

Table 3.11: Tag Entity

Field Name	Data Type	Constraints	Description
tag_id	INT	PK, NOT NULL	Unique identifier for the tag.
name	VARCHAR(255)	NOT NULL	Name of the tag (e.g., "Artificial Intelligence", "Cybersecurity").

DegreeField

Table 3.12: DegreeField Entity

Field Name	Data Type	Constraints	Description
Degreefield_id	INT	PK, NOT NULL	Unique identifier for the programme-tag association.
programme_id	INT	NOT NULL, FK (references Programme(programme_id))	Foreign key linking to the programme.
tag_id	INT	NOT NULL, FK (references Tag(tag_id))	Foreign key linking to the tag.

TagDegreeField

Table 3.13: TagDegreeField Entity

Field Name	Data Type	Constraints	Description
programmetag_id	INT	PK, NOT NULL	Unique identifier for the programme-tag association.
programme_id	INT	NOT NULL, FK (references Programme(programme_id))	Foreign key linking to the programme.
tag_id	INT	NOT NULL, FK (references Tag(tag_id))	Foreign key linking to the tag.

Question

Table 3.14: Question Entity

Field Name	Data Type	Constraints	Description
question_id	INT	PK, NOT NULL	Unique identifier for each question.
question_text	VARCHAR(255)	NOT NULL	The text content of the question.
question_type	TEXT	NOT NULL	Type of question (e.g., slider, multiple-choice).
question_number	INT	UNIQUE, NOT NULL	Sequential number to indicate the order of the question.
tag_id	INT	NOT NULL, FK (references Tag(tag_id))	Foreign key linking the question to a specific tag category.

3.4.8 Interface Design

This section will model the user flow of the application, demonstrating how users will interact with the interface and navigate through the various stages of the recommendation process. Additionally, this section includes a low-fidelity prototype that visually represents the application's

design, offering insight into how the project is intended to appear and function from the user's perspective. It is important to note that the design presented is preliminary and may undergo modifications during the implementation phase.

Upon accessing the system, the user encounters the Landing Page (Figure 3.12), which provides an overview and options to log in or create an account. An existing user may proceed to the Login Page (Figure 3.13) to enter their credentials, while a new user can navigate to the Registration Page (Figure 3.14) to complete the sign-up process. Following successful authentication, the user is directed to pages where they input their academic qualifications (Figure 3.15) and programme interests (Figure 3.16). The system uses this information to generate tailored suggestions on the Show Recommendation Page (Figure 3.17). The user can view detailed information for a specific programme on the View Programme Detail Page (Figure 3.18) and can save selections for future reference on the Favourite List page (Figure 3.19). The system also includes a comprehensive catalogue of all programmes, which is accessible on the View All Programmes Page (Figure 3.20).

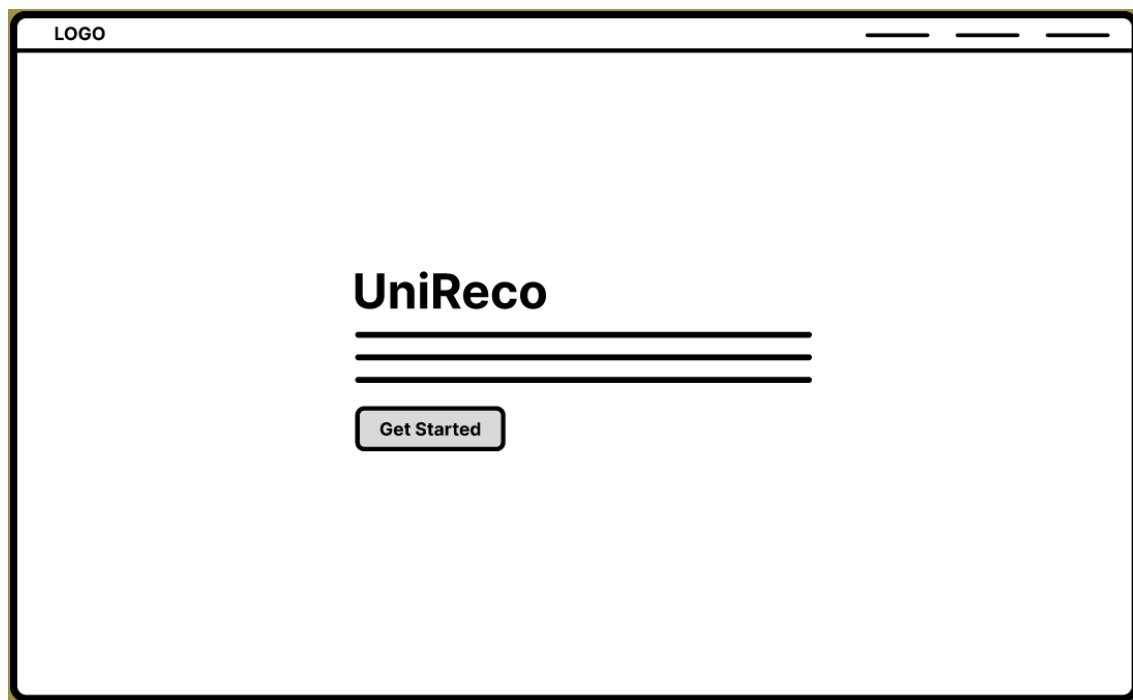
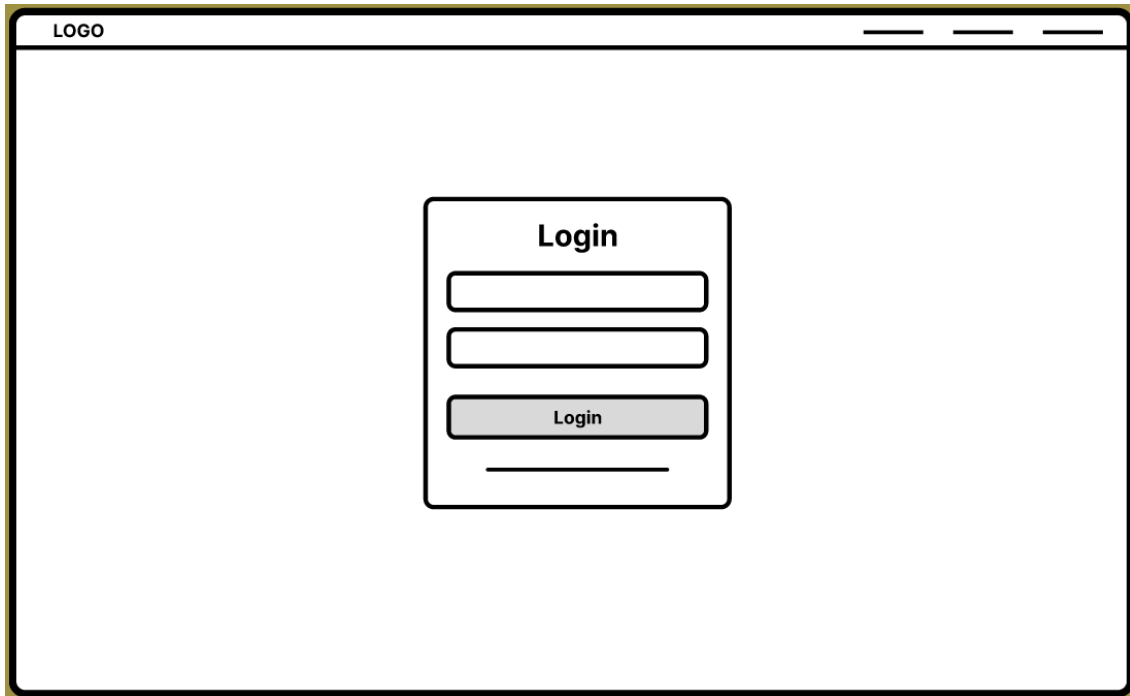
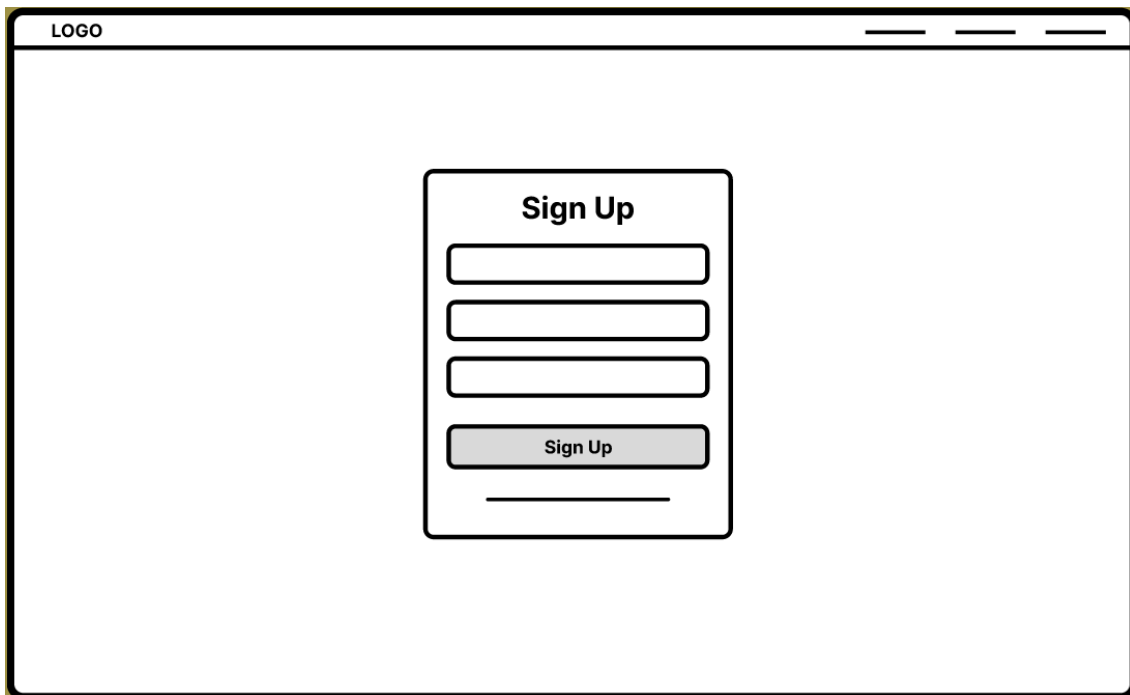


Figure 3.12: Landing Page



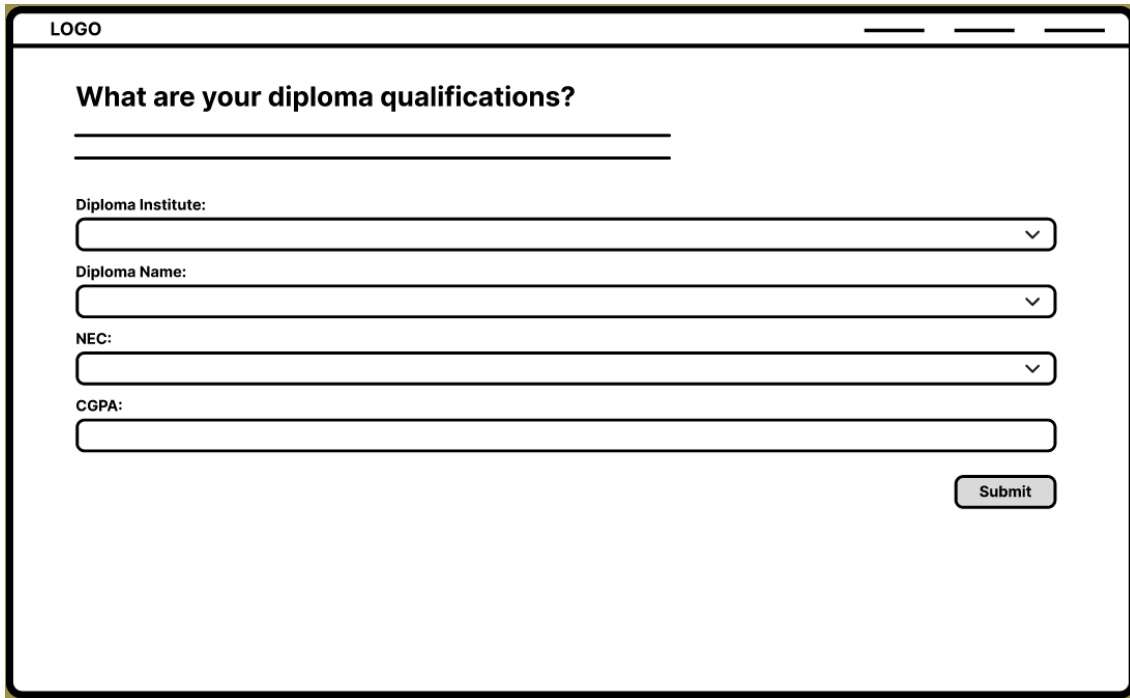
A wireframe of a login page. At the top left is a "LOGO" label. At the top right are three horizontal lines representing browser window controls. In the center is a rounded rectangle containing the title "Login", two input fields, a "Login" button, and a horizontal line for a password or link.

Figure 3.13: Login Page



A wireframe of a registration page. At the top left is a "LOGO" label. At the top right are three horizontal lines representing browser window controls. In the center is a rounded rectangle containing the title "Sign Up", four input fields, a "Sign Up" button, and a horizontal line for a password or link.

Figure 3.14: Registration Page



LOGO

What are your diploma qualifications?

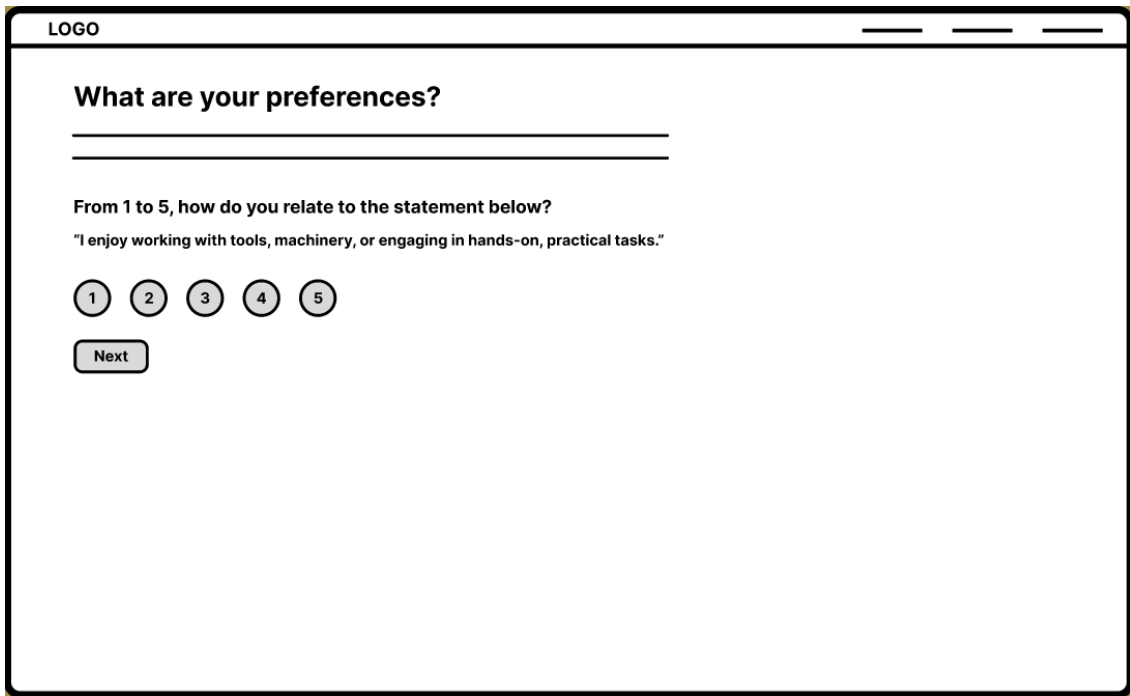
Diploma Institute:

Diploma Name:

NEC:

CGPA:

Figure 3.15: Academic Qualification



LOGO

What are your preferences?

From 1 to 5, how do you relate to the statement below?

"I enjoy working with tools, machinery, or engaging in hands-on, practical tasks."

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Figure 3.16: Interest Page

LOGO

Here's what we recommend for you.

1 - Multimedia

Interested? Here are programs in this field:

Program 1

Program 2

Program 1

2 - Software Engineering

Interested? Here are programs in this field:

Program 1

Program 2

Program 1

3 - Computing Science

Interested? Here are programs in this field:

Program 1

Program 2

Program 1

Figure 3.17: Show Recommendation Page

LOGO

Bachelor in Computer Science (Multimedia Computing) with Honours

Description:

Duration:

Mode:

Entry Requirements:

> Diploma:

> Matriculation:

> STPM:

Career Opportunities (Not limited to):

Institute Details

Institute:

Location:

Go To University Page

Add To Favourites

Related Programs

Bachelor of Computer Science (Multimedia Technology)

Bachelor of Computer Science (Multimedia Computing)

Figure 3.18: View Programme Detail Page

43

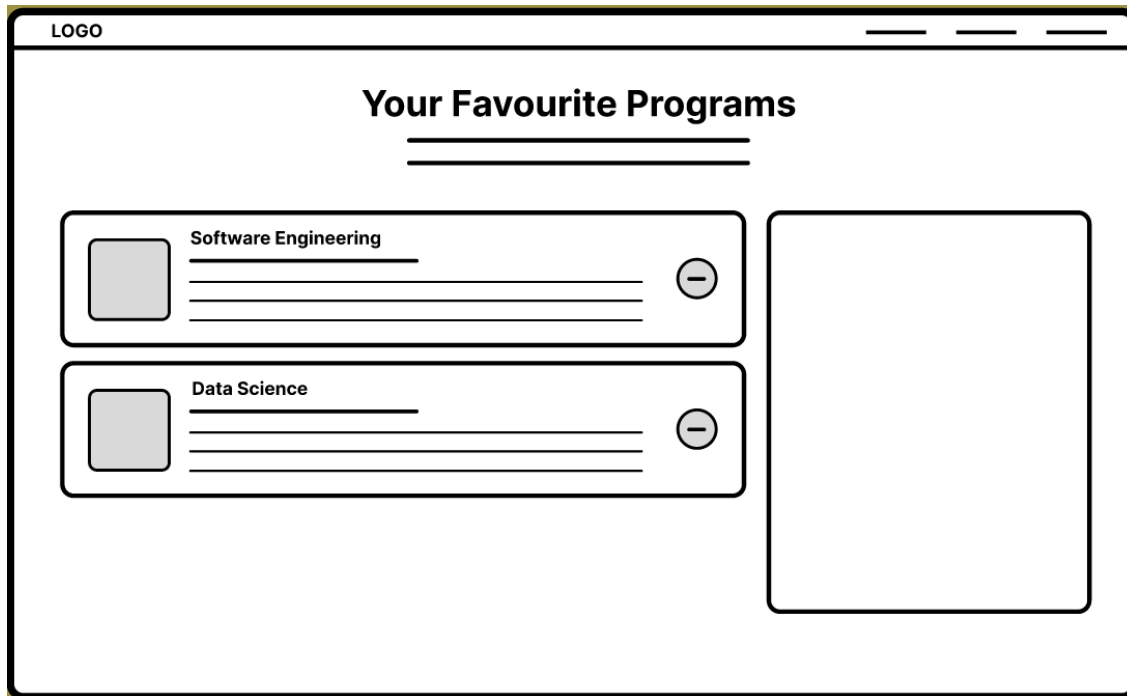


Figure 3.19: Favourite List page

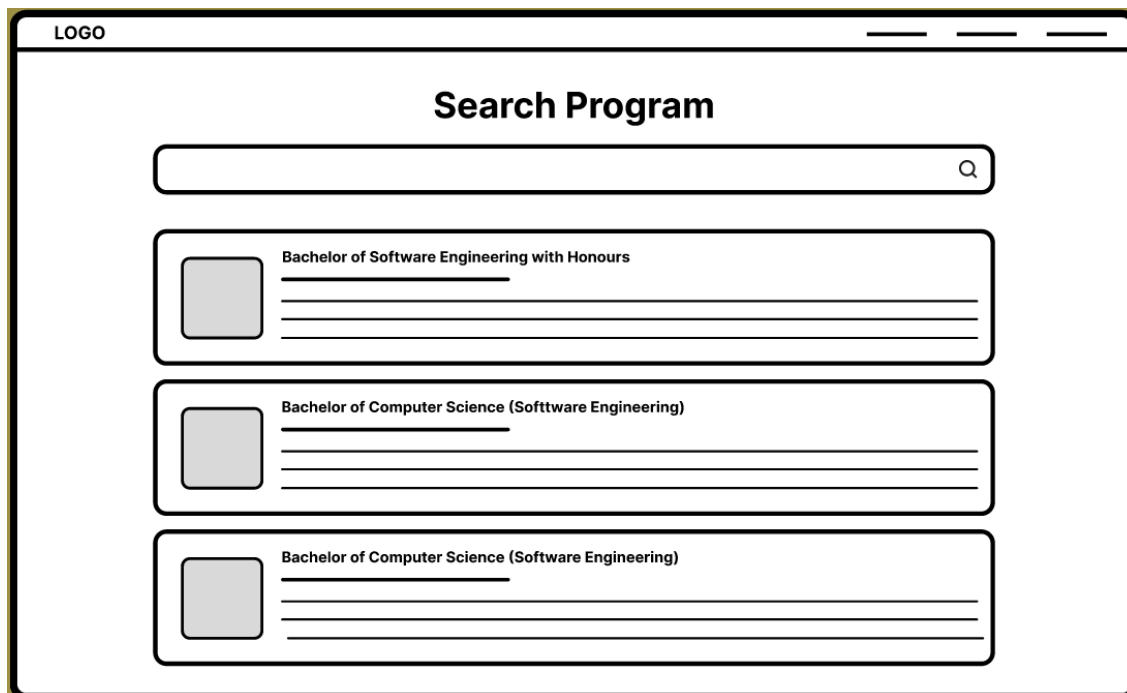


Figure 3.20: View All Programmes Page

3.5 Summary

This chapter presents the methodology employed to guide the development of the recommendation system for university programme selection based on user interests and qualifications. The chapter begins by discussing the requirements gathering process, which identifies the key features and specifications necessary to build the system. The system design is then explored, utilizing various modeling techniques to structure the components and user flow. Additionally, the recommendation logic is proposed, outlining the approach to generating personalized programme recommendations based on user preferences and responses. This chapter provides the necessary groundwork for the next phase of the project, where the implementation of the system will be carried out.

CHAPTER 4: IMPLEMENTATION

4.1 Introduction

This chapter describes the implementation of the project in alignment with the stated objectives. It details how the methodology outlined in the previous chapter was applied to develop a functional prototype. The chapter outlines the overview of the recommendation algorithm, development process, the structure and functionality of each module, the overall user flow, and key considerations encountered during the system's development. The content presented here provides a comprehensive overview of how the system was built based on the proposed design and implementation strategy.

4.2 Implementation Tool

4.2.1 Software used for development

The development of the recommendation system involved the use of various technologies across the backend, frontend, and database layers. Each tool was selected based on its suitability for the project's requirements, ease of integration, and ability to support rapid development. Table 4.1 summarizes the key technologies used, along with their type and specific role in the system's

Table 4.1: Software Used for Developmnet

Technology Used	Type	Explanation
Python	Programming Language	Used for implementing backend logic and the recommendation algorithm. Python was chosen due to its simplicity and availability of mathematical libraries like NumPy that can support similarity calculations and score processing.
Django	Web Framework (Backend)	A high-level Python web framework that provides built-in support for routing, form handling, authentication, and ORM. It simplifies backend development and ensures a secure, scalable architecture.

SQLite3	Database	A lightweight, file-based relational database used during the development phase. It is easy to set up and suitable for rapid prototyping and testing.
HTML, CSS, JavaScript	Web Technologies (Frontend)	Core technologies used to build the user interface. This combination enables basic interactivity and content rendering without introducing complex frontend frameworks.
Bootstrap	CSS Framework	A responsive CSS framework used to streamline UI design. It provides pre-built components and a grid system that ensure consistency and adaptability across screen sizes.

4.3 Programme Data Collection

Three public universities have been analysed, and a total of 17 computer science-related programmes were selected and studied. Each programme is grouped into a specific cluster based on its focus area. Each cluster is associated with its own set of tags, RIASEC profiles, and academic interest categories. These tags and categories are used to match user preferences from the Interest Quiz. The full classification of programmes into clusters, including their related tags and RIASEC profiles, is available in Appendix B.

Each programme includes specific academic entry requirements. These requirements are used to filter users based on their academic qualifications. This helps indicate which programmes a user is likely to be eligible for. The requirements are implemented as a rule-based system, where each rule is stored in the database using JavaScript Object Notation (JSON). These rules are then evaluated by the system during the recommendation process. A sample of the programme entry requirements stored in the database can be seen in Figure 4.1.

All the collected programme data is stored in the system's database and is used during the recommendation process. The recommendation logic refers to this data when determining the most

suitable programmes for a user. This ensures that both user preferences and qualification constraints are taken into account when generating results.

Start typing to filter...	
APP	
Degree fields	+ Add
Entry requirements	+ Add
Favourite programmes	+ Add
Programmes	+ Add
Questions	+ Add
Tag degree fields	+ Add
Tags	+ Add
Universities	+ Add
User datas	+ Add
User preferences	+ Add
Users	+ Add
AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add

Change entry requirement	
Entry Requirement 16	
Stpm:	<pre>{ "AND": [{ "OR": [{ "AND": [{ "atleast": { "count": 1, "grade": "C", "subjects": ["1", "2"], "qualification": "stpm" } }, { "atleast": { "count": 1, "grade": "C", "subjects": ["3", "4", "5", "6"], "qualification": "stpm" } }] }, { "atleast": { "count": 2, "grade": "C", "subjects": ["1", "2", "3", "4", "5", "6"], "qualification": "stpm" } }] }, { "OR": [{ "atleast": { "count": 1, "grade": "C", "subjects": ["7"], "qualification": "spm" } }, { "AND": [{ "atleast": { "count": 1, "grade": "C", "subjects": ["1"], "qualification": "spm" } }, { "atleast": { "count": 1, "grade": "C", "subjects": ["2", "8", "9", "10", "12", "13"], "qualification": "spm" } }] }] }] }</pre>
Matriculation:	<pre>{ "AND": [{ "abovecgpa": { "value": 2, "qualification": "matriculation" } }, { "atleast": { "count": 1, "grade": "C", "subjects": ["7"], "qualification": "spm" } }] }</pre>
Diploma:	<pre>{ "OR": [{ "AND": [{ "abovecgpa": { "value": 2.5, "qualification": "diploma" } }, { "nec_in": { "nec_codes": ["06"], "qualification": "diploma" } }] }, { "AND": [{ "abovecgpa": { "value": 2.75, "qualification": "diploma" } }, { "nec_in": { "nec_codes": ["05", "07"], "qualification": "diploma" } }] }, { "OR": [{ "atleast": { "count": 1, "grade": "C", "subjects": ["7"], "qualification": "spm" } }, { "AND": [{ "atleast": { "count": 1, "grade": "C", "subjects": ["1"], "qualification": "spm" } }, { "atleast": { "count": 1, "grade": "C", "subjects": ["2", "8", "9", "10", "12", "13"], "qualification": "spm" } }] }] }] }</pre>

Figure 4.1: Sample of Entry Requirements used in the database

4.4 Recommendation Algorithm Overview

Figure 4.2 shows the recommendation algorithm used in the system. The recommendation algorithm operates through a multi-stage ranking process designed to provide personalized and relevant programme suggestions. The process begins after a user registers and inputs their information, which is divided into two main categories: academic qualifications and personal interests.

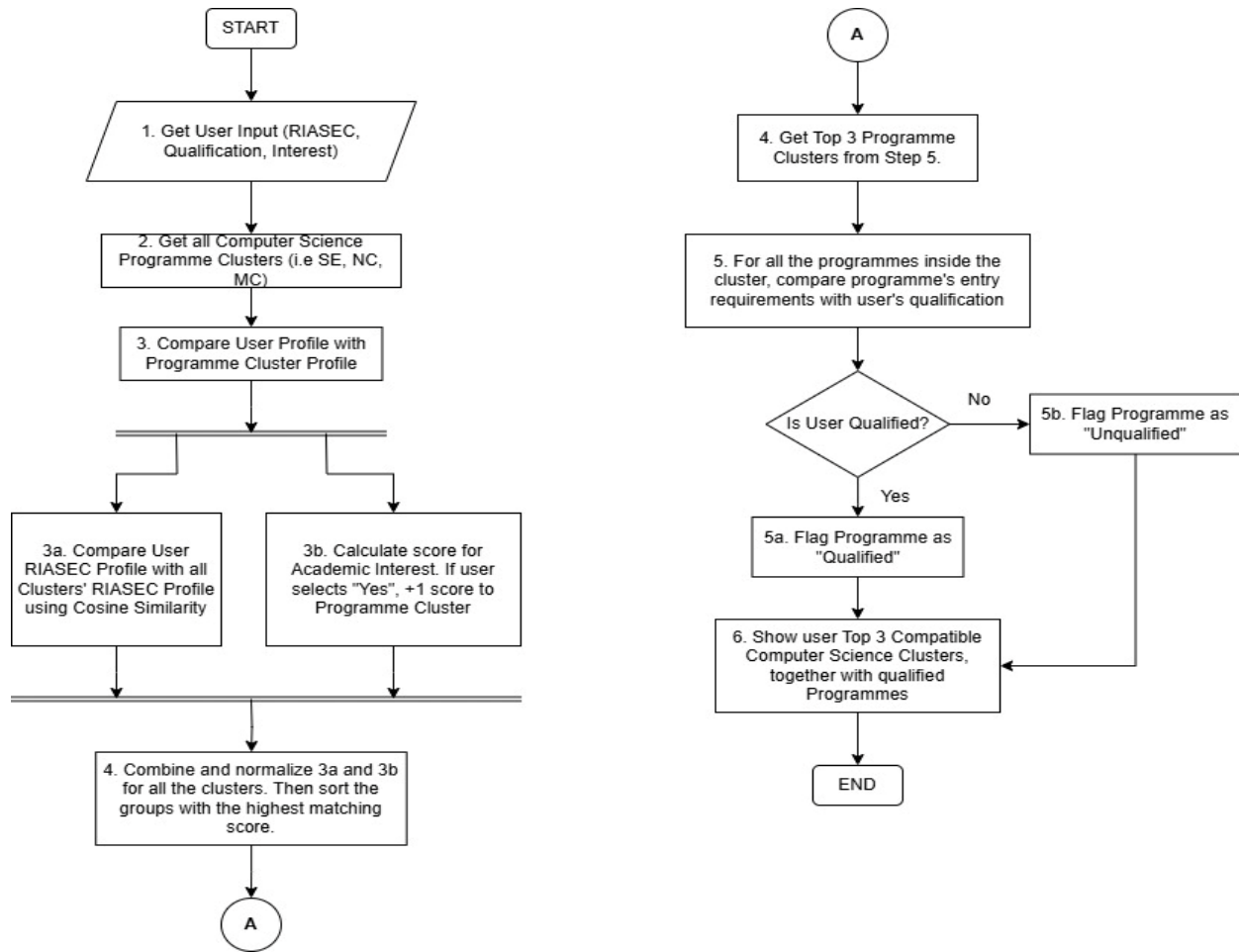


Figure 4.2: Recommendation Algorithm Overview

The process begins with user profile input (Step 1). The system collects essential data such as the user's RIASEC results, academic background, and selected areas of interest. The following phase is programme group identification (Step 2), where the system retrieves a list of Computer Science programme clusters. These include various specializations such as Software Engineering, Network Computing, and Multimedia Computing.

The core of the algorithm involves matching the user's profile with each programme cluster (Step 3). This matching process consists of two parts, namely personality profile comparison (Step 3a) and academic interest scoring (Step 3b). In the first part, the system calculates a personality compatibility score by comparing the user's RIASEC profile to each programme cluster's predefined profile using the cosine similarity metric. In the second part, the system assigns interest scores based on how closely the user's academic interests align with each group.

The next stage is the ranking and selection of the most compatible programme clusters (Step 4). The system combines and normalizes the personality and interest scores for each cluster, then ranks them based on the final score. It proceeds to select the top three clusters that best match the user's profile.

After identifying the top programme groups, the algorithm performs an academic qualification check (Step 5). For each degree programme within the selected groups, the system compares the user's academic qualifications with the specific entry requirements. Each programme is then marked as either "Qualified" (Step 5a) or "Unqualified" (Step 5b), depending on the result of the comparison.

The final step involves displaying the recommendation results to the user (Step 6). The system presents the top three programme clusters in order of compatibility, along with the university programmes that fall under each group. Each listing clearly shows whether the user meets the academic requirements for that particular programme.

4.5 Prototype

A web application named "UniReco" was developed as a functional prototype to demonstrate the core features of the recommendation system. The application is designed to collect user information, including academic qualifications and personal interests, in order to generate suitable programme recommendations. The following subsections describe the key modules and interfaces developed as part of the prototype.

4.5.1 System hierarchy

The system begins with a landing page, which serves as the initial interface for all users. From there, users can choose to register for an account, where they are required to provide basic credentials such as email and password. Upon successful registration or login, users are directed to a qualification selection page, where they must specify their academic background by choosing from Matriculation, Diploma, or STPM. Based on the selected qualification, a corresponding form will be displayed to collect detailed academic information. This data is used to assess the user's eligibility for university programs. After submitting the qualification form, all users are required

to complete the SPM form, which captures essential secondary-level academic results necessary for programme filtering.

Once academic information is submitted, the user proceeds to the Interest Quiz, which captures individual preferences and personality traits using a series of Likert scale and binary questions. The system then combines the academic data and quiz results to generate a personalized list of recommended programmes, which is displayed on the Recommendation Page. Users can explore these programs further by clicking on any listed item, which redirects them to a detailed View Programme page containing in-depth descriptions and requirements. Additional features include the ability to search for specific programmes or universities, filter search results, and bookmark programmes to a personalized favourites list for future reference. This structured flow ensures a guided and tailored experience, from user registration to personalized academic recommendations.

4.5.2 Landing Page

Figure 4.3 shows the landing page of the system. This is the first screen displayed to users upon accessing the platform. It contains a "Get Started" button, which redirects users to the registration form. From there, new users can create an account to proceed with using the system's features.

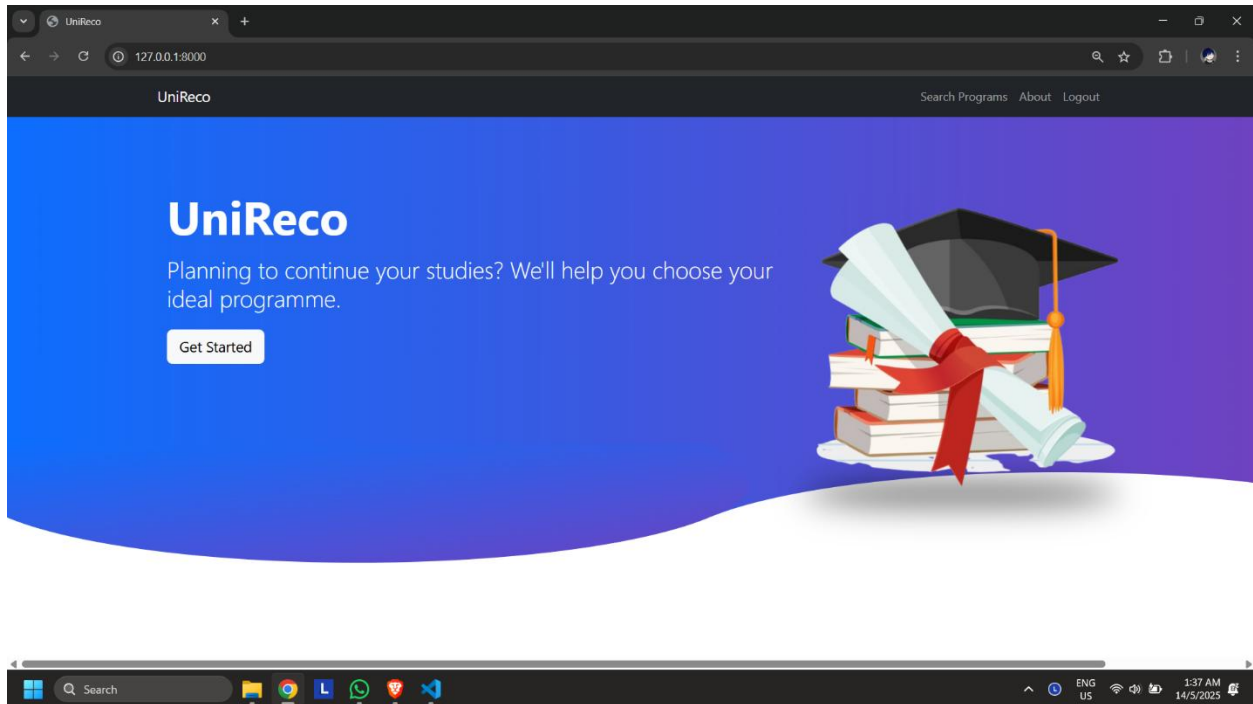


Figure 4.3: Landing Page

4.5.3 User Registration

Figure 4.4 shows the user registration page of the system. This page allows new users to create an account by providing essential information such as email address, and password. Upon successful registration, users are granted access to the main features of the platform and will be redirected back to the landing page for the next step. Input validation is implemented to ensure data integrity and security during the registration process.

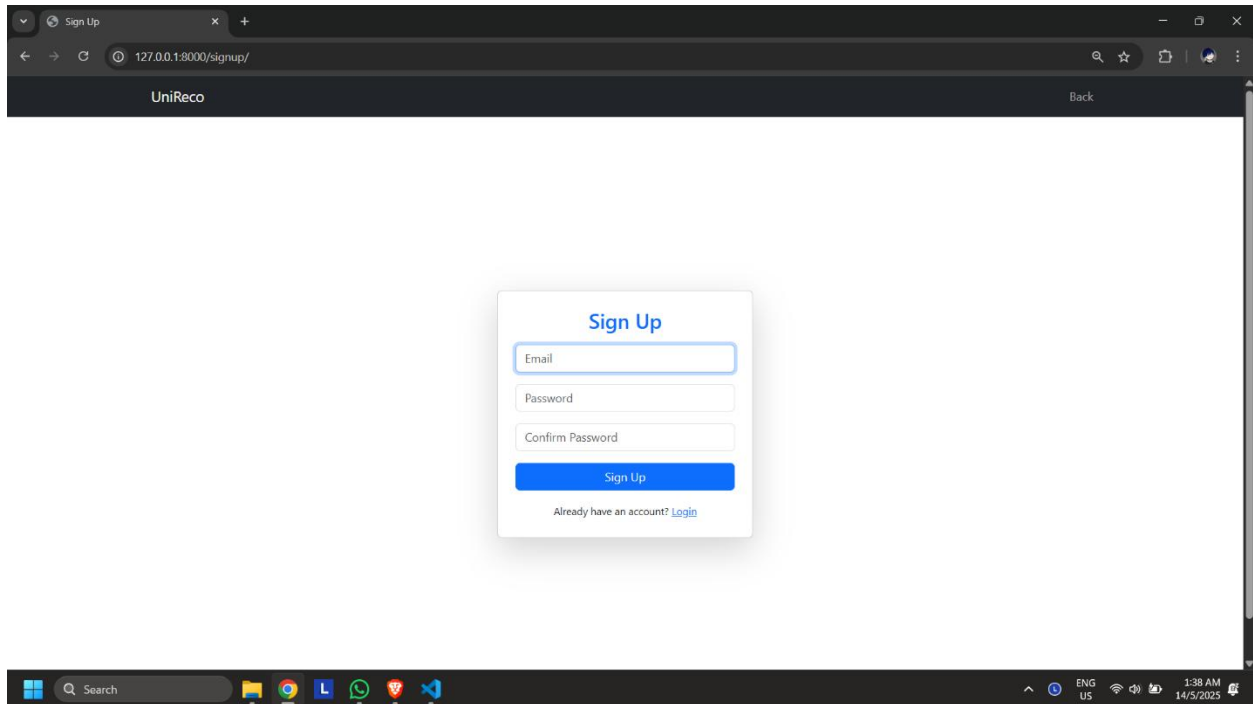


Figure 4.4: User Registration Page

4.5.4 User Login

Figure 4.5 presents the login page of the system. This interface enables registered users to access their accounts by entering their email address and password. The system includes authentication mechanisms to verify user credentials and prevent unauthorized access. Upon successful login, users are redirected back to the Landing Page for the next step.

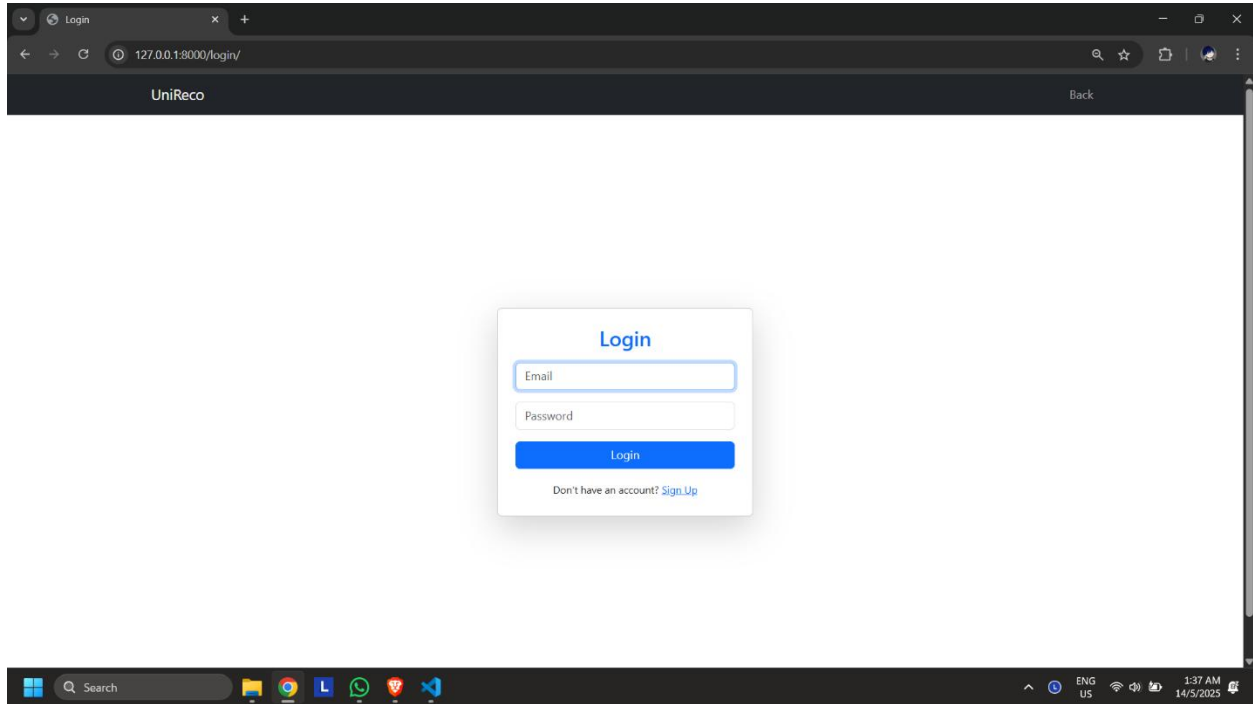


Figure 4.5: Login Page

4.5.5 Select Qualification

Figure 4.6 displays the qualification selection page, where users are required to specify their academic background before proceeding. The page presents three options: Matriculation, Diploma, and STPM, each represented by a distinct button. Upon selection, users are redirected to a corresponding form tailored to the chosen qualification, allowing them to input relevant academic details.

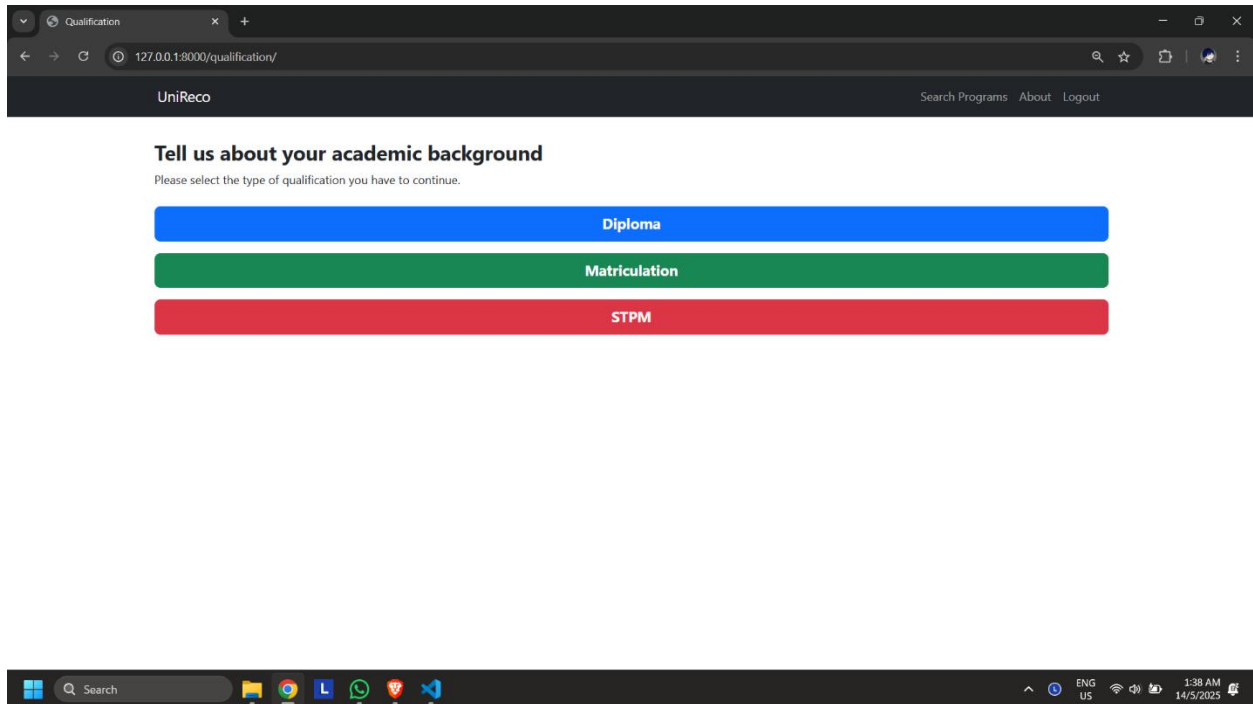


Figure 4.6: Select Qualification Page

4.5.6 Diploma Form

Figure 4.7 shows the form page displayed when the user selects "Diploma" on the qualification selection page. This form collects specific academic information related to the user's diploma qualification. The required fields include the name of the institute where the diploma was obtained, the diploma programme name, the National Education Code corresponding to the diploma classification, and the user's CGPA.

Qualifications - Diploma

127.0.0.1:8000/qualification/diploma/

UniReco Search Programs About Logout

What are your diploma qualifications?
Please input your diploma results.

Diploma Institute:
Politeknik Kuching

Diploma Name:
Diploma in Computer Science

NEC:
05: Natural sciences, mathematics and statistics

CGPA:
0.00

Submit

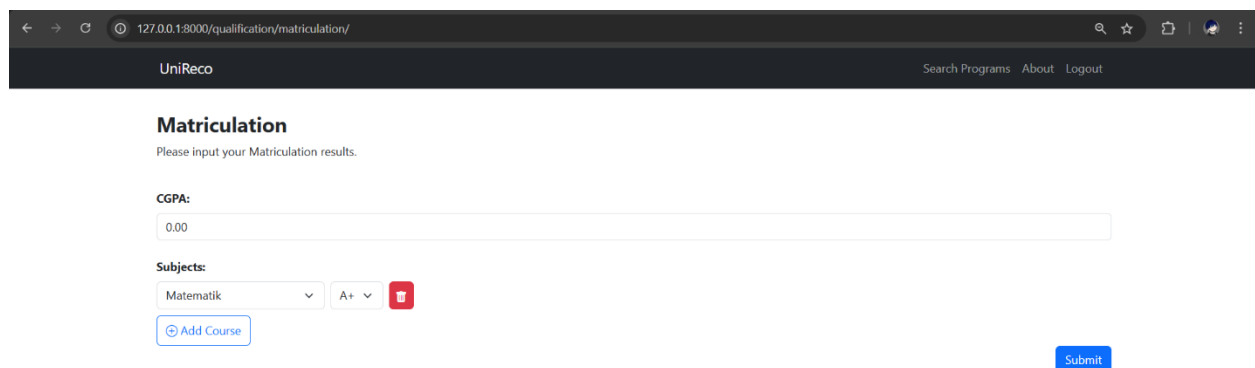
Figure 4.7: Diploma Form Page

The National Education Code (NEC) is a standard system used in Malaysia to classify and code fields of education and training (Kementerian Pengajian Tinggi, 2021). Its main purpose is to create a uniform platform for tracking, evaluating, and reporting on educational programmes, which helps to streamline data integration and align with international reporting standards. The NEC classifies programmes using a four-digit hierarchical structure, starting with a two-digit code for the broad field of study, a three-digit code for a more specialized area, and a four-digit code for a detailed field.

This recommendation system uses the National Education Code (NEC) to identify compatible diploma qualifications for degree programmes. The primary filtering method matches qualifications within the same broad field of study, represented by a two-digit NEC code. For instance, a diploma in Engineering ('07') will be matched with a degree requiring an engineering background. Additionally, the system's logic accounts for cross-field compatibility, where a programme may accept applicants from different but related disciplines. A common example is a diploma holder from Engineering ('07') being considered eligible for a degree in the Information and communication technologies field ('06'). Using the two-digit classification is ideal for this model, as it effectively groups major disciplines for both same-field and cross-field matching, avoiding the needless complexity of more granular codes.

4.5.7 Matriculation Form

Figure 4.8 presents the form displayed when the user selects "Matriculation" on the qualification selection page. In this form, the user is required to enter their CGPA along with the list of subjects taken and the corresponding grades. This information is used to assess the user's academic performance and to filter out programmes for the recommendations.



← → ↻ 127.0.0.1:8000/qualification/matriculation/ 🔍 ☆ 📄 🌐 ⋮

UniReco Search Programs About Logout

Matriculation

Please input your Matriculation results.

CGPA:

Subjects:

Matematik ▾ A+ ▾ 🗑️

+ Add Course

Submit

Figure 4.8: Matriculation Form

4.5.8 STPM Form

Figure 4.9 shows the form presented when the user selects "STPM" on the qualification selection page. This form functions similarly to the Matriculation form, requiring the user to input their CGPA along with the subjects taken and the respective grades.

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/qualification/stpm/". The page header includes the UniReco logo and navigation links for "Search Programs", "About", and "Logout". The main heading is "Sijil Pelajaran Tinggi Malaysia (STPM)" with a subtext "Please input your STPM results." Below this, there is a "CGPA:" label and a text input field containing "0.00". Under the "Subjects:" label, there are two rows of subject selection. Each row consists of a dropdown menu (showing "Matematik M" and "Matematik T" respectively), a grade dropdown (showing "A+" and "A+" respectively), and a red trash icon. At the bottom left, there is a blue "Add Course" button with a plus icon. At the bottom right, there is a blue "Submit" button.

Figure 4.9: STPM Form Page

4.5.9 SPM Form

Figure 4.10 displays the SPM qualification form page, which appears after the user completes any of the selected qualification forms (Matriculation, Diploma, or STPM). On this page, the user is required to enter their SPM results, including subjects and corresponding grades. This information is essential for filtering programme recommendations, as universities often impose specific subject requirements at the SPM level for admission eligibility.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/qualification/spm/'. The page header includes the 'UniReco' logo and navigation links for 'Search Programs', 'About', and 'Logout'. The main heading is 'What are your SPM qualifications?' followed by the instruction 'Please input your Sijil Pelajaran Malaysia (SPM) results.' Below this, there is a table with four rows for subjects: 'Matematik', 'Sains', 'Bahasa Melayu', and 'Sejarah'. Each row has a dropdown menu for the subject name, a dropdown menu for the grade, and a red trash icon. The grades entered are 'A+', 'B-', 'A-', and 'B+' respectively. At the bottom left of the table is a button labeled 'Add Course' with a plus icon. At the bottom right of the form area is a blue 'Submit' button.

Subject	Grade	Action
Matematik	A+	Remove
Sains	B-	Remove
Bahasa Melayu	A-	Remove
Sejarah	B+	Remove

[+ Add Course](#) [Submit](#)

Figure 4.10: SPM Form Page

4.5.10 Interest Quiz

Figure 4.11 presents the Interest Quiz Page, which users are directed to after completing the SPM qualification form. This page contains a series of questions designed to assess the user's personality traits using RIASEC and academic interests. The quiz includes a combination of Likert scale items and Yes/No questions, aimed at capturing user preferences and tendencies. The responses collected from this quiz serve as an input for the system's recommendation engine to generate more personalized and relevant programme suggestions. The full question list can be referred to in Appendix C.

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/interest/". The website header includes the "UniReco" logo and navigation links for "Search Programs", "About", and "Logout". The main content area is titled "What are your preferences?" and instructs users to "Answer the questions below to help us understand your preferences." The first question asks, "From 1 to 5, how do you relate to the statement below?" with the statement: "I enjoy working with tools, machinery, or engaging in hands-on, practical tasks." Below the statement are five circular radio buttons labeled 1, 2, 3, 4, and 5. The button labeled "3" is selected, indicated by a blue highlight. A "Next" button is located below the radio buttons. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons (File Explorer, Chrome, L, WhatsApp, V, and a blue icon). The system tray on the right shows the language set to "ENG US", signal icons, and the date and time: "1:41 AM 14/5/2025".

Figure 4.11: Interest Quiz Page

4.5.11 Recommendation Page

Figure 4.12 shows the Recommendation Page, where users are presented with a list of academic programmes categorized under various clusters or subfields of computer science. For each cluster, the system displays the relevant programmes along with an indication of whether the user meets the minimum qualification requirements. Users can click on any listed programme to view its detailed information. This page serves as the final output of the recommendation process, providing users with personalized suggestions based on their academic background and interests.

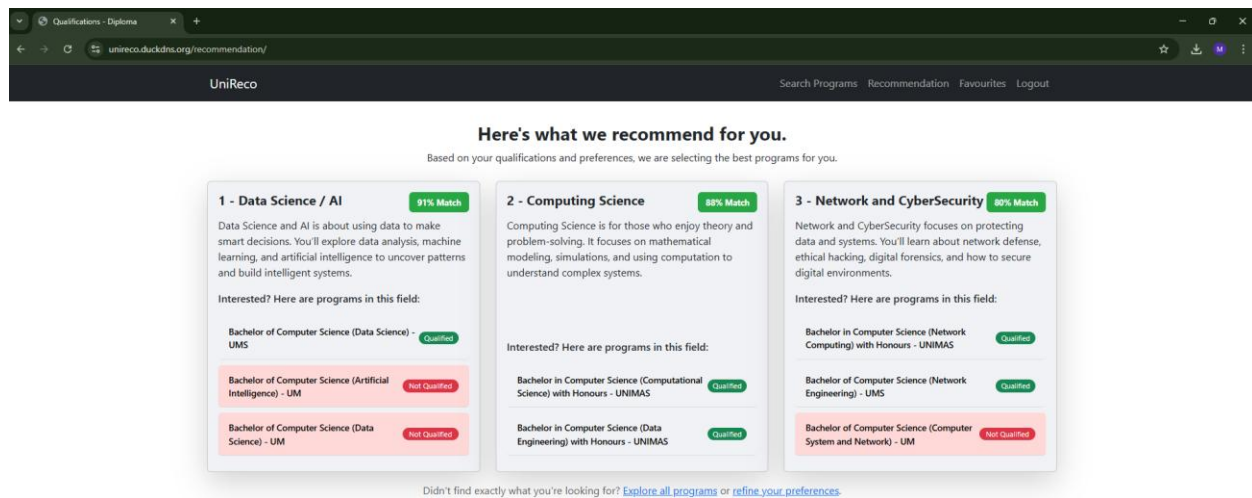


Figure 4.12: Recommendation Page

4.5.12 View Programme

Figure 4.13 displays the View Programme Page, where users can access detailed information about a specific programme. This includes the programme's description, full admission requirements which can be expanded when clicked, career opportunities, and other relevant details. Users can also click on the “Go to University Page” button to be redirected to the university’s official website for further information. Additionally, the page offers an “Add to Favorite” button, allowing users to bookmark the programme for later viewing and easy access. At the bottom of the page, users can view a list of related similar programmes, helping them explore alternative options within the same field.

The screenshot shows a web browser at the URL `unireco.duckdns.org/program/2/`. The page header includes the UniReco logo and navigation links: Search Programs, Recommendation, Favourites, and Logout. The main content area features the programme title **Bachelor in Computer Science (Data Engineering) with Honours** and a large image of a hand pointing at a digital interface with gears and the text **DATA SCIENCE**. To the right, the **Institute Details** section lists the Institute as Universiti Malaysia Sarawak (UNIMAS) and the Location as Kota Samarahan, Sarawak, Malaysia. Below this are two buttons: **Go To University Page** and **Add to Favourites**. The **Related Programs** section displays a card for the **Bachelor in Computer Science (Computational Science) with Honours** at UNIMAS. The main description text states: "Nowadays, enterprises are moving to a higher level of automation and are highly dependent on information provided by computer-based system. These complex systems will become an exceedingly important strategic asset of the organisation. The study of Information System, therefore, is concerned with the development of systems that will get the right information to the right people at the right time. This programme's four (4) year structure is designed for students who wish to be competent in deciding how technology are utilise to support organisational/business objectives or to create new opportunities, determine the supporting process and data, implementing computer-based systems and also developing new and innovative products. All courses consist of extensive theoretical and practical studies will require the use of a wide variety of system development tools. With organisations, people, processes and technical aspects in system development as its focal point, students will be equipped with basic organisational theory, decision-making, project management and system development skills, which are parallel with the latest information technology system." The **Duration** is 4 Years (8 Semesters), the **Mode** is Full time, and the **Entry Requirements** section is expanded to show the **Diploma** requirement: Possess a recognised Diploma in related field with a CGPA of at least 2.50. The **Matriculation** and **STPM** sections are also visible. The **Career Opportunities (Not limited to):** section lists various roles including Information Systems Officer, System Analyst, Database Administrator, Database Analyst, Project Manager, Application Developer, Application Consultant, Business Analyst, Business Process Analyst, User Interface Designer, Web Content Manager, e-Business Manager, IT Consultant, IT Architect, Research Officer and Educator.

Bachelor in Computer Science (Data Engineering) with Honours

Institute Details

Institute: Universiti Malaysia Sarawak (UNIMAS)
Location: Kota Samarahan, Sarawak, Malaysia

[Go To University Page](#)

[Add to Favourites](#)

Related Programs

Bachelor in Computer Science (Computational Science) with Honours
Universiti Malaysia Sarawak (UNIMAS)
Computational Science is the field of st...

Description:
Nowadays, enterprises are moving to a higher level of automation and are highly dependent on information provided by computer-based system. These complex systems will become an exceedingly important strategic asset of the organisation. The study of Information System, therefore, is concerned with the development of systems that will get the right information to the right people at the right time. This programme's four (4) year structure is designed for students who wish to be competent in deciding how technology are utilise to support organisational/business objectives or to create new opportunities, determine the supporting process and data, implementing computer-based systems and also developing new and innovative products. All courses consist of extensive theoretical and practical studies will require the use of a wide variety of system development tools. With organisations, people, processes and technical aspects in system development as its focal point, students will be equipped with basic organisational theory, decision-making, project management and system development skills, which are parallel with the latest information technology system.

Duration: 4 Years (8 Semesters)
Mode: Full time

Entry Requirements: [\(Tap to Expand\)](#)

▼ **Diploma:**

- Possess a recognised Diploma in related field with a CGPA of at least 2.50

› **Matriculation:**

› **STPM:**

Career Opportunities (Not limited to):
Information Systems Officer, System Analyst, Database Administrator, Database Analyst, Project Manager, Application Developer, Application Consultant, Business Analyst, Business Process Analyst, User Interface Designer, Web Content Manager, e-Business Manager, IT Consultant, IT Architect, Research Officer and Educator.

Figure 4.13: View Programme Page

4.5.13 Search Programme

Figure 4.14 presents the Search Programme Page, where users can search for programmes by university name or programme name. The page also provides filtering options, allowing users to narrow down search results based on specific criteria. Once the user enters their search query and applies any filters, they can click on the relevant search results to be redirected to the View Programme Page for detailed information about the selected programme.

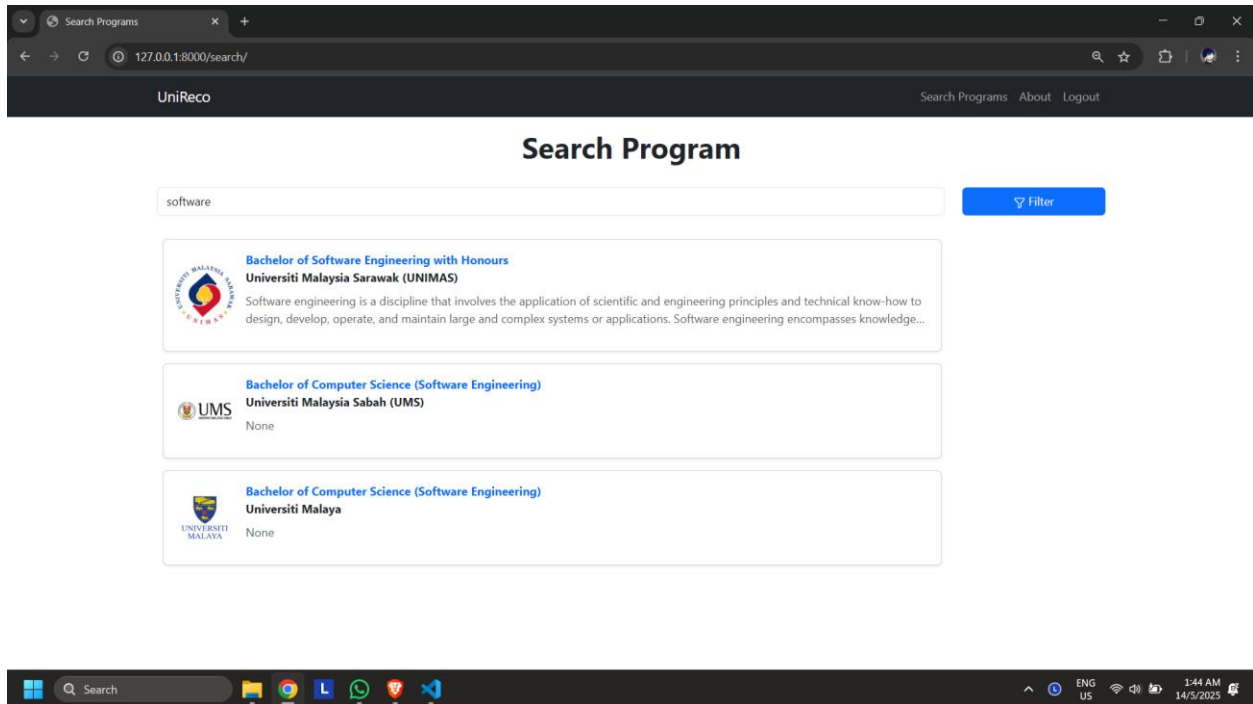


Figure 4.14: Search Programme Page

4.5.14 Favourite Programme

Figure 4.15 displays the View Favorite Programme Page, where users can access a list of their bookmarked programmes. From this page, users can click on any programme to be redirected to the View Programme Page for detailed information. Additionally, users have the option to remove programmes from their favorites by clicking the trash icon next to the programme, allowing them to manage their saved programmes efficiently.

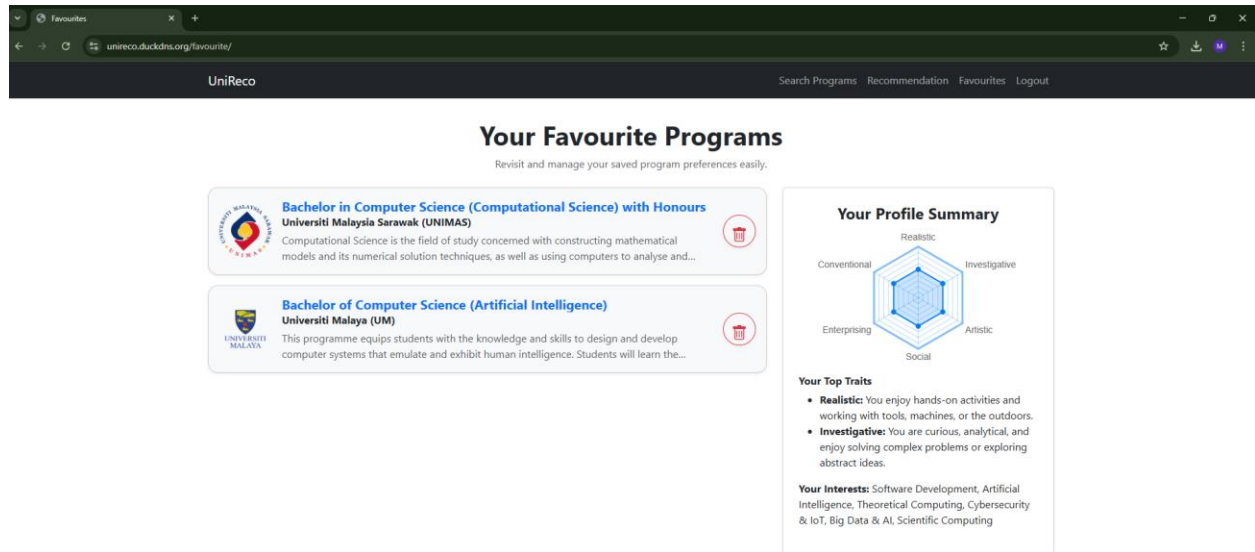


Figure 4.15: View Favorited Programme Page

4.6 Discussion

A notable adjustment during the implementation phase was the shift from recommending individual programmes to recommending clusters. This change was based on the observation that many computer science programmes appeared highly similar when using publicly available data, making it difficult to distinguish nuanced differences across universities. Grouping them into defined subfields, such as Software Engineering or Multimedia, offered a more practical and meaningful level of recommendation. To further improve accuracy, the RIASEC model was also integrated to address situations where users showed equal interest in multiple areas, making preference detection unclear. By incorporating personality dimensions, the system could better capture user tendencies and align them with suitable programme clusters, resulting in more accurate and personalized recommendations.

One of the most significant technical challenges encountered was the implementation of the academic filtering system. University entry requirements are inherently complex, with distinct criteria for different qualifications such as SPM, STPM, and Matriculation. Accommodating these diverse rules required a manually built system tailored to interpret and apply various combinations of conditions for each programme. This process was highly labor-intensive, involving detailed analysis, rule configuration, and extensive testing to ensure accuracy in filtering programmes based on academic qualifications.

4.7 Summary

This chapter presents the implementation of the “UniReco” system, detailing the development process, key modules, and user interaction flow based on the methodology outlined earlier. Each component of the system, from data input forms to the recommendation interface, was developed to support the core functionality of providing tailored programme suggestions based on user input. The prototype serves as a step toward completing the project objectives. The next chapter will focus on the testing phase, where the functionality and performance of the prototype will be assessed to ensure it meets the software quality standard.

CHAPTER 5: TESTING

5.1 Introduction

The purpose of this chapter is to evaluate and validate the implementation of the Recommendation System for University Programmes. Three types of testing were conducted to ensure the quality and reliability of the system, which are functionality testing, usability testing, and user acceptance testing.

Functionality testing was performed to verify that all features of the system operate correctly and according to the specified requirements. It involves checking each modules to confirm that the system behaves as expected without errors. Usability testing, on the other hand, focuses on assessing the user experience by observing how easily and efficiently users can interact with the system. This includes evaluating the interface design, navigation flow, and overall satisfaction of the users. User acceptance testing was carried out to confirm that the system meets user needs and expectations. Together, these tests provide comprehensive insights into the effectiveness and readiness of the system for deployment.

5.2 Functionality Testing

5.2.1 Overview

The purpose of the functionality testing is to evaluate the performance and correctness of the recommendation system. This test involves examining each core feature of the system to ensure that it performs as intended under normal and edge-case scenarios. All modules are tested individually and as a complete system. The objective is to identify any defects or inconsistencies in the system's behavior and resolve them prior to deployment. By thoroughly validating the system's operations, functionality testing helps ensure that the application is reliable, stable, and capable of meeting user expectations.

5.2.2 List of Functionality Testing Result

A total of 12 modules were tested to ensure full functional coverage of the system. The testing process adopted the black-box testing technique, focusing on both valid and invalid inputs according to the nature of each module. Each function was tested for expected behavior, ensuring the system meets its functional requirements. Table 5.1 presents a sample of the testing conducted

on the recommendation module, detailing the specific aspects tested, testing steps, expected outcomes, and actual results. The remaining test results for other modules are documented in Appendix D. Overall, the unit testing of all modules indicates that the system is approximately 99 percent defect-free, demonstrating its ability to operate reliably without disrupting the user experience.

Table 5.1: Recommendation Module Functional Testing Results

Unit Testing 9: Recommendation Module						
Objective: To verify login module with correct email and password						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify form submission with all valid inputs	Fill in the previous form	Should be showing a top 3 groups of computer science fields	Recommendation shown.	Pass	-
2	Verify redirect to view program details	Click on any of the programme listed on the recommendation	Should be redirected to the appropriate page with programme detail	Redirected to programme detail page with correct information	Pass	-
3	Verify behaviour when user tries to access recommendation page without completing previous steps	Register a new account and modify link to "Recommendation" page	Should prompt error or redirect back to qualification page	Redirected to "Get-Started" section.	Pass	-

5.3 Usability Testing

5.3.1 Overview

The purpose of usability testing is to evaluate the overall user experience of the recommendation system. This type of testing focuses on how real users interact with the system and whether the system is intuitive, accessible, and satisfactory from a user perspective. Potential users, consisting of multiple demographics, are tested. They go through the system while observers provide instructions and record their interactions. The results are then tabulated to identify usability issues and gather feedback for system improvement.

5.3.2 List of Usability Testing Result

Usability testing for the recommendation module was conducted through live one-on-one sessions with potential users. Each user was guided through specific test steps while their interactions were observed to assess whether the interface was intuitive, understandable, and easy to navigate. The goal was to identify any usability issues that could affect user satisfaction or accessibility. The testing was done together with User Acceptance Testing, as covered in the next section. Users performed predefined tasks based on test scenarios for the module, and observers recorded whether users could complete the tasks without assistance. Any issues or confusion were noted in the actual results, along with user comments and suggestions.

Table 5.2 presents a sample of the usability test cases, specifically for the recommendation module. The remaining test results for other modules are documented in Appendix E. Overall, users successfully navigated the system and understood the content as intended. Minor feedback, such as adding hover effects for clearer interaction, was noted for improvement. The results confirm that the system provides user-friendly experience and meets essential usability standards.

Table 5.2: Recommendation Module Usability Testing

Module: Recommendation Module					
Objective: To determine if the user can understand the recommendation results page.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail

1	User views recommended programs	1. Direct user to open the Recommendations page 2. Instruct user to explore the page and locate a program	User understands layout, terminology, and clickable elements	Users interacted successfully; Some users suggested adding hover effects for clearer interface.	Pass
2	User returns to "Recommendations" page.	1. Direct user to go to Home 2. Instruct user to navigate back to Recommendations	User navigates without help	User returned without assistance	Pass

5.4 User Acceptance Testing

User Acceptance Testing (UAT) was conducted in conjunction with usability testing to validate that the system achieves its primary objective as a university programme recommendation tool. The core purpose of this testing was to evaluate the system's function in terms of academic advising by assessing its overall usefulness for students planning their higher education. To gather a comprehensive understanding, the testing targeted two key demographics. The primary group consisted of students who have not yet attended university, to determine if the system is a practical tool for their study planning. The second group comprised university graduates or currently attending one. These people provided an experienced perspective on whether the system is useful for study planning. The questions used to guide this evaluation can be found in Appendix F.

To simulate real-world usage, the system was hosted online allowing participants to interact with it independently in a realistic environment. This hosted version of the system was made available for the duration of the project until the end of the FYP2 timeline. After completing the full user flow, participants were asked to submit their feedback through a Google Form. The link to access the web application can be referred to in Appendix G.

5.4.1 List of User Acceptance Testing Results

A total of 13 users participated and gave feedback. The feedbacks were analyzed to evaluate the system's overall effectiveness. A detailed breakdown of the results is presented in the following sections, supported by charts.

Figure 5.1 shows the user's feedback of the User Registration and Login of the system. Initial user experience with account management was highly positive. The registration process was considered clear and easy to understand, with 76.9% of users rating it 5/5 and 23.1% rating it 4/5. Similarly, the login process was rated 5/5 by 84.6% of users. The system's feedback for incorrect login details also received favorable responses, with 61.5% rating it 5/5 and 38.5% rating it 4/5.

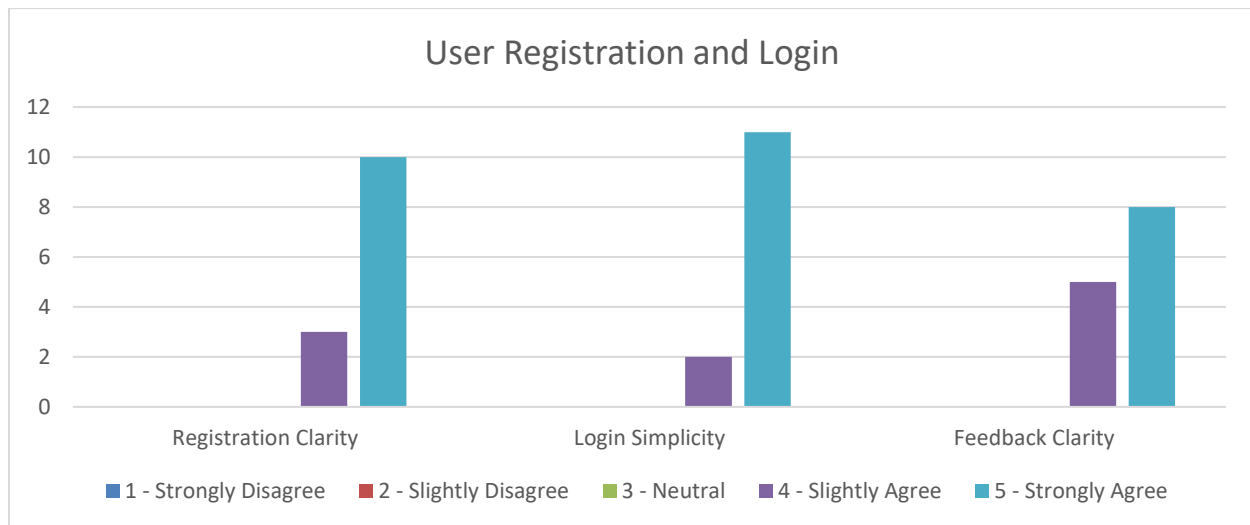


Figure 5.1: User Registration & Login UAT Results

Figure 5.2 presents feedback on the Academic Qualification module. The process of inputting academic details such as CGPA, subjects, and grades was met with high satisfaction. It was rated 5/5 by 92.3% of users for ease of input, while the layout of the qualification form was rated 5/5 by 69.2% and 4/5 by 30.8%. Error messages for incorrect inputs also received positive feedback with 69.2%. Overall, 84.6% of participants found this part of the system user-friendly. Mixed ratings indicate that the user interface for this module is lacking but still have room for improvement.

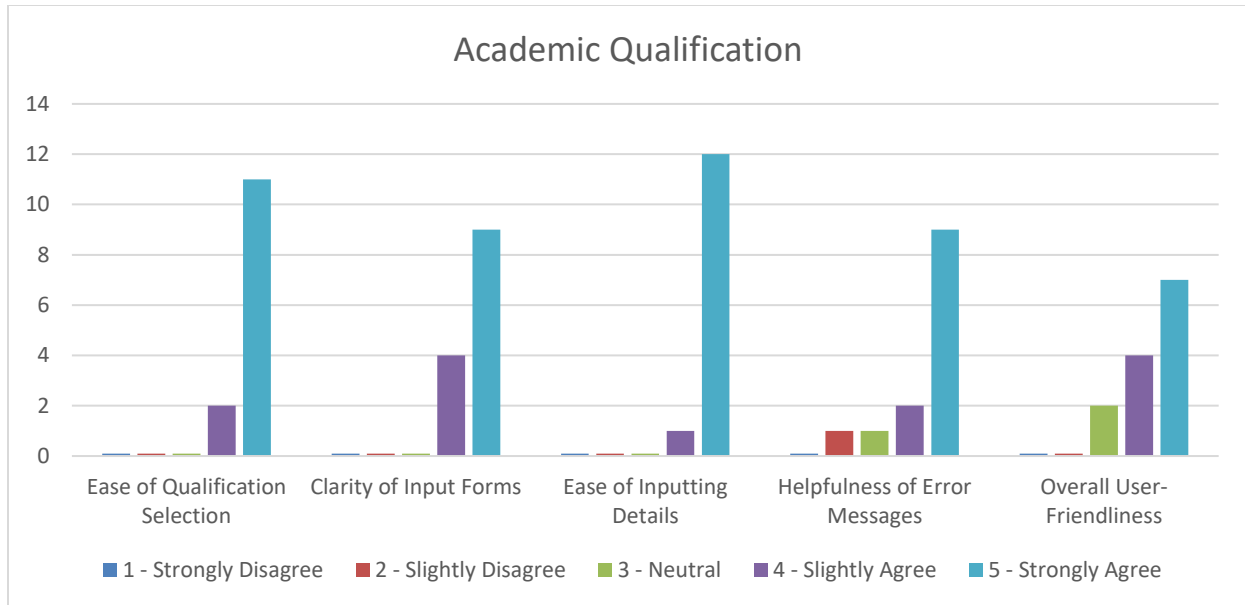


Figure 5.2: Academic Qualification UAT Results

Figure 5.3 covers the Personality and Interest Quiz section, which is used to generate personalized recommendations. For the clarity of questions, 61.5% of users rated it 5/5, and 30.8% gave it 4/5. The ease of selecting answers was rated 5/5 by 76.9%, while the navigation within the quiz form received a 5/5 rating from 69.2% of participants.

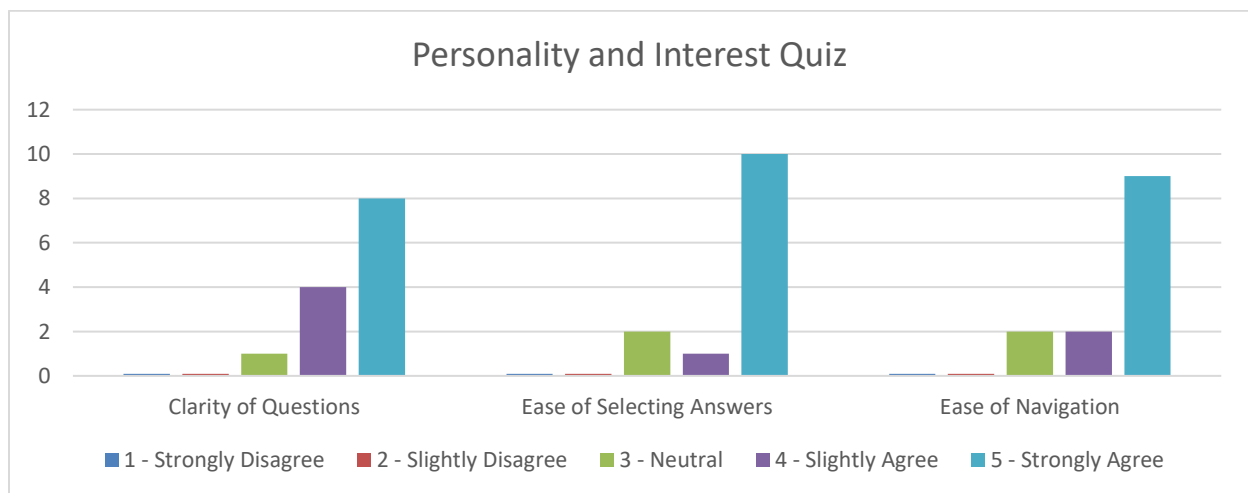


Figure 5.3: Interest Quiz UAT Results

Figure 5.4 shows the evaluation of the Recommendation of Programme section. The presentation of recommendation results was rated 5/5 by 69.2% of users and 4/5 by 23.1%. The viewing of programme details and the sufficiency of information provided were also positively received, with 69.2% and 61.5% respectively rating these features 5/5. However, even when the recommendation output was clear, informative, and well received by users, mixed ratings of this module indicate that there is lacking of information provided.

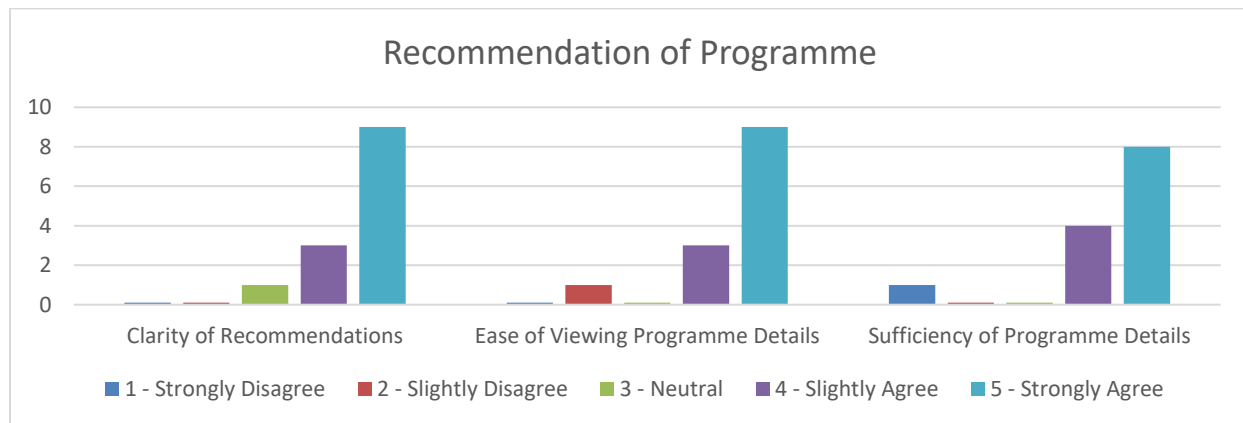


Figure 5.4: Recommendation of Programme UAT Results

Figure 5.5 highlights the feedback on the Search and Filtering Function. While the clarity of the search function was rated 4/5 or 5/5 by 84.7% of users, the ease of finding specific programmes showed more varied results. Only 38.5% rated it 5/5, and 23.1% gave it 3/5. User comments reflected this, with several suggested improvements such as adding filter options by university and location. This indicates that this module needs refining.

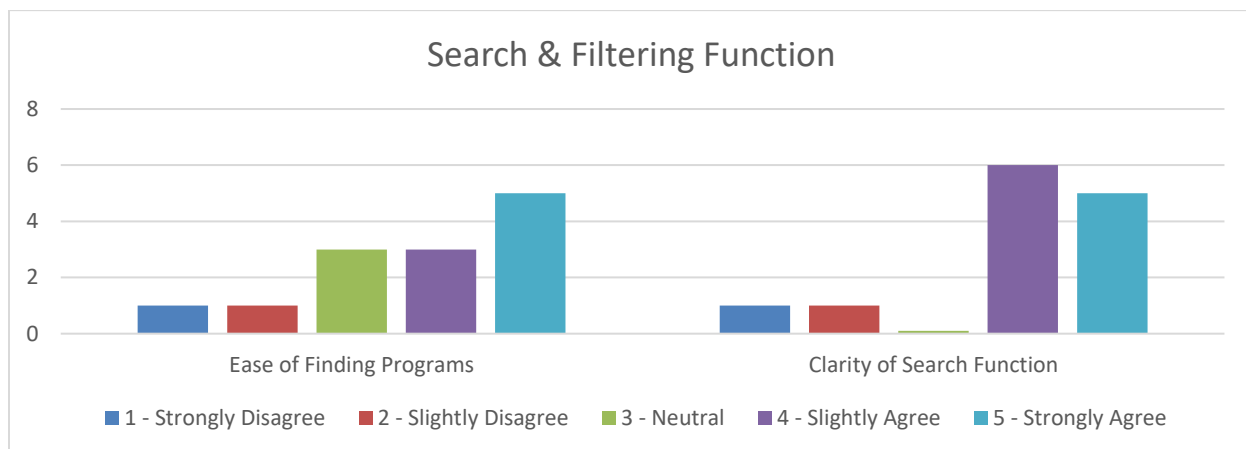


Figure 5.5: Search & Filtering Function UAT Results

Figure 5.6 shows user responses to the Favourite Programmes feature. It was considered highly successful, with 76.9% rating the ease of adding and removing programmes as 5/5. Additionally, 84.6% of users gave a 5/5 rating for viewing their favourited list. This indicates that users found the feature intuitive and helpful for planning purposes.

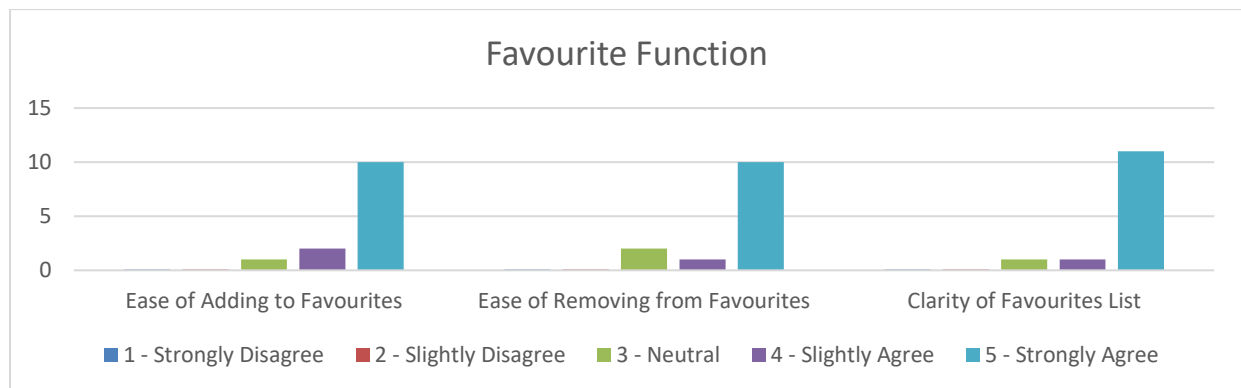


Figure 5.6: Favourite Function UAT Results

Figure 5.7 presents the Overall User Acceptance of the system, showing a generally positive reception with areas for improvement. The relevance of recommended programmes to users' interests and academic background received mixed ratings, with the majority giving a score of 4/5 (92.3% and 92.4% respectively). This suggests that the recommendations are somewhat accurate but not fully aligned with users' personality and interests. Likewise, 92.3% found the 'View Programme' page informative, and 92.4% agreed the recommendations were helpful for planning their studies. However, feedback on overall utility was mixed. Ratings for overall feature usefulness ranged from 'Slightly Disagree' to 'Strongly Agree', indicating that while many found the features helpful, the system is not yet universally useful. Similarly, responses on the system's effectiveness in solving study planning problems varied, highlighting that the system shows potential but needs further refinement to ensure consistent reliability in real-world use.

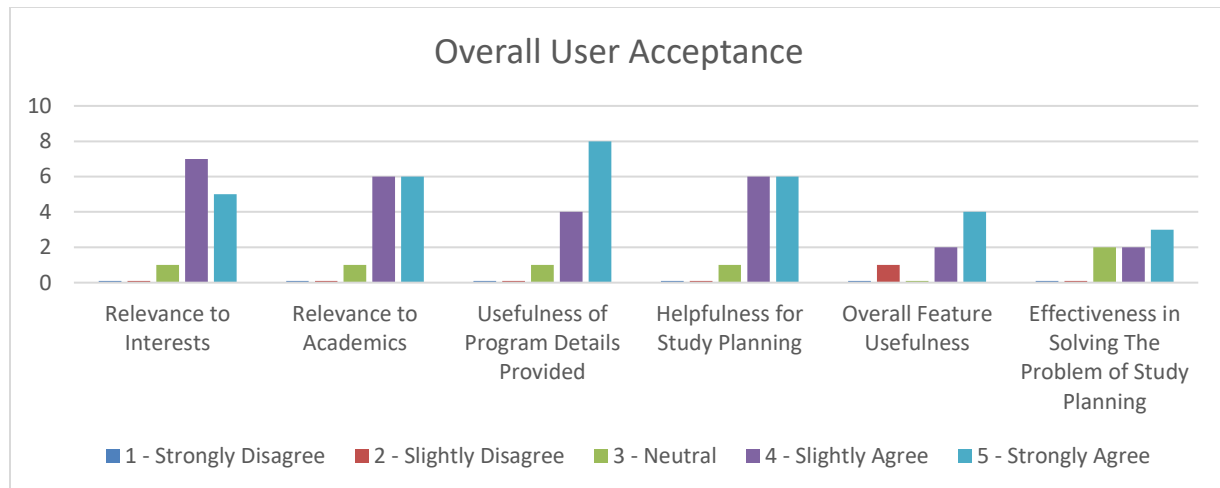


Figure 5.7: Overall User Acceptance

Based on direct user feedback comments, several key areas for enhancement were identified. Participants suggested improving interface clarity with features like a user guide and hover effects to better direct users. The search functionality was highlighted for improvement, with specific requests to add filters for both university and location. Input forms could also be refined by using a table layout for SPM results and adding descriptive labels to the interest quiz's Likert scale for better comprehension. A final common suggestion was to expand the system's database with a broader selection of programmes. The full, detailed user comments and suggestions can be referred to in Appendix H.

5.5 Summary

This chapter presented functionality, usability, and user acceptance testing on the implemented prototype of the University Programme Recommendation System. Functionality testing confirmed that all features work correctly as designed. Usability testing, involving real users from the target group, provided useful insights into the user experience. Results from both tests were positive, showing the system effectively meets its goals. This chapter demonstrates that the system has been successfully evaluated in line with the project's objectives.

CHAPTER 6: CONCLUSION & FUTURE WORKS

6.1 Introduction

This chapter concludes the project by providing a comprehensive summary and analysis of its outcomes. It begins by revisiting the project's achievements, directly aligning them with the objectives established at the outset. The chapter then discusses the primary contributions and significance of the developed system. Subsequently, it acknowledges the limitations inherent in the current study and scope of the work. Finally, this chapter proposes several recommendations for future work, outlining potential enhancements and expansions that could build upon the foundation established by this project.

6.2 Achievement

This project has contributed significantly to supporting students in making informed decisions regarding their higher education pathways. By providing a web-based recommendation system that suggests suitable computer science programmes based on user interests and preferences, it enables users to explore academic options that align with their strengths and inclinations. The project successfully met all its primary objectives. The achievements corresponding to each objective are detailed in table 6.1.

Table 6.1: Project Objective and Achievement

Objective	Achievement
1. To analyse Computer Science programmes	An analysis of Computer Science programmes from Malaysian public universities was conducted to extract key features relevant to questionnaire development and classification, as presented in “Chapter 4, Section 4.3: Programme Data Collection”.
2. To design a recommendation algorithm	A recommendation algorithm was successfully designed, as outlined in “Chapter 3, Section 3.4.3: Recommendation Algorithm Design”, and later implemented to filter and suggest suitable programmes based on user input, as discussed in “Chapter 4, Section 4.4: Recommendation Algorithm Overview”.

3. To build a functional web application	A web-based system was successfully developed with all core functionalities, as presented in “Chapter 4, Section 4.5: Prototype”.
4. To evaluate the system	The system was evaluated for its functionality, usability, and user acceptance through various testing methods, as reported in “Chapter 5: Testing”.

6.3 Limitation

The project has several limitations that should be acknowledged. Firstly, the system's scope is currently narrow. It focuses only on computer science programmes and does not include other academic disciplines which confines its utility to a specific group of students.

Secondly, a significant limitation lies within the recommendation algorithm and its underlying data. The information used was gathered from publicly available curriculum outlines on university websites, without any direct collaboration with the institutions themselves. Consequently, the algorithm is designed to match user interests with these surface-level details and is not equipped to process or weigh deeper factors like tuition fees, university reputation, or specific faculty strengths. This results in a recommendation that may not fully distinguish between programs that differ greatly in overall value and cost.

Lastly, the system was developed and tested for a small number of users. Its current architecture may therefore face performance and scalability challenges if deployed in a real-world environment with a large, concurrent user base.

6.4 Future Work

There are several avenues for improving and expanding the system in future iterations. One of the key enhancements involves the recommendation algorithm. Future versions of the system could integrate data-driven technologies such as machine learning to better understand user preferences and improve the accuracy of recommendations. Implementing such technologies would require the collection of more comprehensive user data over time to train adaptive and responsive models.

In addition to algorithmic improvements, the database itself should be expanded. The current scope is limited to computer science programmes, which restricts the system's applicability. By including academic fields beyond computer science and incorporating a wider range of universities and programmes, the system could become more inclusive and valuable to a broader audience of users with diverse academic interests.

Furthermore, a crucial direction for future work is to establish direct collaborations with academic institutions. Such partnerships would allow the system to move beyond publicly available data, which is often limited to surface-level curriculum outlines. By working directly with universities, the system could access more detailed and verified information, including specific module contents, faculty expertise, and graduate employment statistics. This deeper and more reliable data would significantly enhance the accuracy of the recommendation engine, enabling it to provide more holistic and helpful guidance for prospective students.

REFERENCES

- Alghamdi, S., Alzhrani, N., & Algethami, H. (2019). Fuzzy-Based Recommendation System for University Major Selection. In *Proceedings of the 11th International Joint Conference on Computational Intelligence* (pp. 317-324). <https://doi.org/10.5220/0008071803170324>
- Bhumichitr, K., Channarukul, S., Saejiem, N., Jiamthapthaksin, R., & Nongpong, K. (2017). Recommender Systems for university elective course recommendation. In *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1-6). <https://doi.org/10.1109/JCSSE.2017.8025933>
- Career Key. (2025). *Holland code assessment and RIASEC*. Career Key.
<https://www.careerkey.org/fit/personality/holland-code-assessment-riasec>
- Industry Leaders Magazine. (2017, June 29). 13 million students pick the wrong college due to lack of information. Industry Leaders Magazine.
<https://www.industryleadersmagazine.com/13-million-students-pick-wrong-college-due-lack-information/>.
- Kementerian Pengajian Tinggi. (2021). *Kod Pendidikan Nasional 2020 (NEC-2020)*. Bahagian Perancangan dan Penyelidikan Dasar.
- Lassibille, G., & Navarro Gómez, M. L. (2008). Why do higher education students drop out? Evidence from Spain. *Education Economics*, 16(1), 89–105.
<https://doi.org/10.1080/09645290701523267>
- Lutkevich, B., & Lewis, S. (2022, November). What is the waterfall model? TechTarget.
<https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>
- Mokarrama, M. J., Khatun, S., & Arefin, M. S. (2020). A content-based recommender system for

- choosing universities. *Turkish Journal of Electrical Engineering and Computer Sciences*, 28(4), 2128-2142. <https://doi.org/10.3906/elk-1911-37>
- Ng, J. (2019, November 15). Content-based recommender using natural language processing (NLP). *KDnuggets*. <https://www.kdnuggets.com/2019/11/content-based-recommender-using-natural-language-processing-nlp.html>
- NVIDIA. (n.d.). *Recommendation system*. NVIDIA. Retrieved December 9, 2024, from <https://www.nvidia.com/en-us/glossary/recommendation-system/>
- Rodzi, R. M., Basri, N. E. A., & Nopiah, Z. M. (2024). The cause of student dropout from TVET at Malaysia Industrial Training Institute. *International Journal of Academic Research in Business & Social Sciences*, 14(9), 4718–4753. <http://dx.doi.org/10.6007/IJARBS/v14-i9/22740>
- Ricci, F., Rokach, L., & Shapira, B. (2021). *Recommender systems: Techniques, applications, and challenges*. In B. Shapira, L. Rokach, & R. Francesco (Eds.), *Recommender Systems Handbook: Third Edition* (pp. 1-35). Springer US. [https://doi.org/10.1007/978-1-0716-2197-4_1​;contentReference\[oaicite:0\]{index=0}](https://doi.org/10.1007/978-1-0716-2197-4_1​;contentReference[oaicite:0]{index=0}).
- Tutorials Point. (2024). SDLC - Waterfall Model. Tutorials Point. https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.

APPENDIX A – Google Form Questionnaire Structure

Recommendation System For University Programme Based On User Interest

Introduction

Welcome!

I'm Mohammad Khairul Ikhwan bin Abdul Basit, a final-year Bachelor of Software Engineering student at UNIMAS.

This questionnaire is part of my final-year project research. The purpose of this survey is to gather insights from respective respondents for developing a recommendation system that able to recommend suitable university programs, majors, or degrees based on student's interests before applying to enrol in the university program.

This survey consists of three sections:

- **Section A:** Demographic Information
- **Section B:** Current Challenges in Choosing a Suitable University Program
- **Section C:** Preferences for the Recommendation System

Completing this form will take approximately **10–15 minutes**. Your responses are invaluable and greatly appreciated. All feedbacks received are **confidential** and totally for the final year project purposes!

If you have any questions or inquiries, feel free to contact me at **78178@siswa.unimas.my (Ikhwan)**.

Thank you for your time and support!

Section A: Demographic Information

This section collects basic details about you. Your responses will remain confidential and are solely for research purposes.

1. **What is your age?**
 - a) 15 - 20 years old
 - b) 20 - 25 years old
 - c) 30 years old and up
2. **What is your current level of education?**
 - a) Sijil Pelajaran Malaysia (Secondary school)

- b) Pre-University (Form 6, Matriculation, Foundations)
 - c) Diploma
 - d) Degree
 - e) Masters / PhD Level
3. **If your answer in Question 2 is up to Diploma, have you ever considered pursuing a degree program at a university?**
- a) Yes
 - b) No
 - c) Maybe
4. **Where you are originally come from?**
- a) Sarawak
 - b) Sabah
 - c) Perlis
 - d) Kedah
 - e) Pulau Pinang
 - f) Perak
 - g) Selangor
 - h) Wilayah Persekutuan (Kuala Lumpur/Putrajaya/Labuan)
 - i) Negeri Sembilan
 - j) Melaka
 - k) Johor
 - l) Pahang
 - m) Terengganu
 - n) Kelantan
 - o) Not from Malaysia
5. **Please identify your study of interest (including which programme that you want to focus on and what you are currently study):**
- a) Computer Science

- b) Engineering
 - c) Medical
 - d) Social Art
 - e) Business Administration
 - f) Other: _____
-

Section B: Current Challenges in Choosing a Suitable University Program

For clarification, a **program** refers to a broader field of study offered by a university, usually the name of the degree, such as **Bachelor of Computer Science, Bachelor of Mechanical Engineering, or Bachelor of Business Administration**. It typically spans several years and leads to a degree or certification.

A **university course**, on the other hand, refers to individual subjects or classes within a program, such as **"Introduction to Programming"** or **"Data Structures"** in a Computer Science program. Courses are the building blocks of a program, providing specific knowledge or skills related to the program's overall focus.

This section will gather information about the challenges people face when choosing a university program.

6. **How difficult do you find the process of selecting a university program that aligns with your goals and interest?**

- a) Very Easy [1]
- b) 2
- c) 3
- d) 4
- e) Very Difficult [5]

7. **What challenges do you face when selecting a university program? (Select all that apply)**

- a) Lack of information about the programs and university
- b) Difficulty understanding the differences between various fields of study
- c) Uncertainty about which program aligns with my interests and career goals
- d) Overwhelmed by the number of options
- e) Difficulty understanding which university offers the best program for my needs

- f) Other: _____
8. **What sources do you currently use to gather information about university programs? (Select all that apply)**
- a) University websites
 - b) Social media
 - c) Career counseling
 - d) Advice from teachers or family
 - e) University Catalogue websites (StudyMalaysia, TopUniversities, etc.)
 - f) Other: _____
-

Section C: Preferences for the Recommendation System

This section aims to gather input on preferences and expectations for the recommendation system, helping to ensure it meets users' needs and provides relevant and personalized program suggestions.

9. **On a scale of 1 to 5, how useful would a recommendation system be for suggesting programs and universities based on your interests?**
- a) Not Useful [1]
 - b) 2
 - c) 3
 - d) 4
 - e) Very Useful [5]
10. **How would you prefer to provide your interests and preferences to the recommendation system?**
- a) Answering a detailed questionnaire (e.g., questions about your interests, career goals, preferred subjects)
 - b) Selecting from a list of predefined options (e.g., choosing your preferred fields of study, career goals, or skills)
 - c) A combination of the above
11. **Other than the recommendation feature, what other features would you expect from a recommendation system for university programs? (Select all that apply)**
- a) Information about the program details (e.g., curriculum, career paths)

- b) Easy-to-use interface
- c) Search function (The ability to search for programs available in other universities)
- d) Bookmarking or the option to save my favorite programs for later reference
- e) Information about job prospects and other career-related information in the field
- f) Other: _____

12. How likely would you be to use a recommendation system like this to help you choose a university program?

- a) Very unlikely [1]
- b) 2
- c) 3
- d) 4
- e) Very likely [5]

13. What do you think would make this recommendation system more helpful to you?

14. Do you have any other concerns/suggestions regarding this recommendation system?

APPENDIX B – Programme Data Collection

Programme Cluster Tags

Cluster	Programme	Tags
Business IT	- Bachelor of Computer Science (Business Computing) – UMS - Bachelor of Computer Science (Information Systems) – UM	"Fintech Ecommerce" "Tech Business"
Computing Science	Bachelor's in computer science (Computational Science) with Honours – UNIMAS	"Scientific Computing" "Theoretical Computing"
Data Science / AI	- Bachelor of Computer Science (Artificial Intelligence) – UM - Bachelor of Computer Science (Data Science) – UMS - Bachelor of Computer Science (Data Science) – UM - Bachelor's in computer science (Data Engineering) with Honours – UNIMAS	"AI Big Data" "AI Interest"
Multimedia	- Bachelor's in computer science (Multimedia Computing) with Honours – UNIMAS - Bachelor of Computer Science (Multimedia Computing) – UM - Bachelor of Computer Science (Multimedia Technology) – UMS	"Game Design" "UI UX Design"
Network and CyberSecurity	- Bachelors in computer science (Network Computing) with Honours – UNIMAS - Bachelor of Computer Science (Computer System and Network) – UM - Bachelor of Computer Science (Network Engineering) – UMS	"Cyber IoT" "Network Security"
Software Engineering	- Bachelor of Computer Science (Software Engineering) – UMS - Bachelor of Computer Science (Software Engineering) – UM - Bachelor of Software Engineering with Honours – UNIMAS	"Software Development" "Software Engineering Process"

Programme Cluster RIASEC Profiles used in the Recommendation System

Cluster	Realistic (R)	Investigative (I)	Artistic (A)	Social (S)	Enterprising (E)	Conventional (C)
Business IT	3	3	1	5	3	3
Computing Science	5	5	1	1	1	5
Data Science / AI	5	5	1	1	3	3
Multimedia	3	3	5	1	3	3
Network and CyberSecurity	5	5	5	1	3	5
Software Engineering	3	3	1	3	5	5

APPENDIX C – Interest Quiz Questions

Academic Interest Questions Used:

1. From 1 to 5, how do you relate to the statement below?

“I enjoy working with tools, machinery, or engaging in hands-on, practical tasks.”

Likert Scale (1–5)

2. From 1 to 5, how do you relate to the statement below?

“I enjoy building and configuring computer systems, networks, or other technical infrastructure.”

Likert Scale (1–5)

3. From 1 to 5, how do you relate to the statement below?

“I enjoy solving complex problems, conducting research, or exploring abstract ideas.”

Likert Scale (1–5)

4. From 1 to 5, how do you relate to the statement below?

“I enjoy analyzing data, debugging code, or systematically diagnosing and resolving technical issues.”

Likert Scale (1–5)

5. From 1 to 5, how do you relate to the statement below?

“I enjoy expressing myself through creative activities like art, music, writing, or design.”

Likert Scale (1–5)

6. From 1 to 5, how do you relate to the statement below?

“I enjoy designing intuitive user interfaces, creating engaging digital experiences, or finding elegant solutions.”

Likert Scale (1–5)

7. From 1 to 5, how do you relate to the statement below?

“I enjoy helping others, teaching, or working in teams to achieve common goals.”

Likert Scale (1–5)

8. From 1 to 5, how do you relate to the statement below?

“I enjoy communicating technical concepts to non-technical people or gathering requirements from users.”

Likert Scale (1–5)

9. From 1 to 5, how do you relate to the statement below?

“I enjoy leading projects, persuading others, or making decisions that impact the outcome of tasks.”

Likert Scale (1–5)

10. From 1 to 5, how do you relate to the statement below?

“I enjoy identifying market needs, developing product strategies, or influencing business decisions with technology.”

Likert Scale (1–5)

11. From 1 to 5, how do you relate to the statement below?

“I enjoy working with data, following structured procedures, or organizing information in a systematic way.”

Likert Scale (1–5)

12. From 1 to 5, how do you relate to the statement below?

“I enjoy ensuring code quality, maintaining documentation, or strictly following technical specifications and standards.”

Likert Scale (1–5)

13. Would you enjoy designing, building, or maintaining software systems?

Yes / No

14. Are you interested in how computers communicate and how to protect them from threats?

Yes / No

15. Are you excited by the idea of creating games, immersive environments, or interactive media?

Yes / No

16. Are you curious about how machines can learn, think, or analyze data?

Yes / No

17. Do you want to apply technology in a business setting or work on digital products?

Yes / No

18. Do you enjoy abstract thinking, logic, or solving theoretical problems?

Yes / No

19. Would you enjoy optimizing how software is developed — like planning, testing, or improving teamwork?

Yes / No

20. Would you like working with cybersecurity systems, ethical hacking, or connected smart devices?

Yes / No

21. Would you enjoy designing user interfaces, animations, or visual experiences like AR/VR?

Yes / No

22. Would you enjoy training AI models or working with large datasets?

Yes / No

23. Would you like to build e-commerce systems, manage business data, or innovate in financial tech?

Yes / No

24. Would you like to work on simulations, scientific computing, or mathematical modeling?

Yes / No

APPENDIX D – Functional Testing Results

Unit Testing 1: Registration Module						
Objective: To verify login module with correct email and password						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify registration with valid email and password	Email: user@example.com Password: Valid@123	Should able to register and prompt “Registration Successful”	Successfully registered and redirected to login page	Pass	—
2	Verify registration with missing email	Email: null Password: Valid@123	Should prompt error “Email is required”	Error message shown	Pass	—
3	Verify registration with missing password	Email: user@example.com Password: null	Should prompt error “Password is required”	Error message shown	Pass	—
4	Verify registration with invalid email format (missing @)	Email: userexample.com Password: Valid@123	Should prompt error “Invalid email format”	Error message shown	Pass	—
5	Verify registration with invalid email format (missing domain)	Email: user@.com Password: Valid@123	Should prompt error “Invalid email format”	Error message shown	Pass	—
6	Verify registration with weak password (too short)	Email: user@example.com Password: abc	Should prompt error “Password too weak”	Error message shown	Pass	—
7	Verify registration with password missing required characters	Email: user@example.com Password: password	Should prompt error “Password must include number and symbol”	Error message shown	Pass	—
8	Verify registration with already registered email	Email: admin@gmail.com Password: Valid@123	Should prompt error “Email already registered”	Error message shown	Pass	—

Unit Testing 2: Login Module						
Objective: To verify login module with correct email and password						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify login with valid registered email and correct password	Email: user@example.com Password: Valid@123	Should be able to login and prompt “Login Successful”	Successfully logged in and redirected to homepage	Pass	—
2	Verify login with valid registered email but incorrect password	Email: user@example.com Password: WrongPass1	Should prompt error “Incorrect password”	Error message shown	Pass	—
3	Verify login with unregistered email	Email: notregistered@example.com Password: Valid@123	Should prompt error “Account does not exist”	Error message shown	Pass	—
4	Verify login with empty email	Email: null Password: Valid@123	Should prompt error “Email is required”	Error message shown	Pass	—
5	Verify login with empty password	Email: user@example.com Password: null	Should prompt error “Password is required”	Error message shown	Pass	—

Unit Testing 3: Select Qualification Module						
Objective: To verify qualification selection redirects the user appropriately						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify redirection after user selects Diploma	Qualification: Diploma	Should be redirected to the programme recommendation form	Successfully redirected to the form	Pass	—
2	Verify redirection after user selects Matriculation	Qualification: Matriculation	Should be redirected to the programme recommendation form	Successfully redirected to the form	Pass	—
3	Verify redirection after user selects STPM	Qualification: STPM	Should be redirected to the programme recommendation form	Successfully redirected to the form	Pass	—
4	Verify behavior when user tries to bypass selection (e.g., modifying URL)	Qualification: not selected / URL manually accessed	Should prompt error or redirect back to selection page	Error message shown / redirected to selection page	Pass	—

Unit Testing 4: Diploma Form Module						
Objective: To verify diploma module						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify form submission with all valid inputs	Institute: Politeknik Diploma Name: IT NEC: NEC123 CGPA: 3.50	Should proceed to recommendation step or next form	Form submitted successfully	Pass	—
2	Verify form with non-numeric CGPA	CGPA: "abc"	Should prompt error "Invalid CGPA"	Error message shown	Pass	—
3	Verify form with CGPA out of range	CGPA: 5.0	Should prompt error "CGPA must be between 0.00 and 4.00"	Error message shown	Pass	—
4	Verify form submission with missing institute	Institute: null	Should prompt "Please select your institution"	Error message shown	Pass	—
5	Verify form submission with missing diploma name	Diploma Name: null	Should prompt "Please select your diploma"	Error message shown	Pass	—
6	Verify form submission with invalid NEC format	NEC: "123NEC"	Should prompt "Invalid NEC format"	Error message shown	Pass	—

Unit Testing 5: Matriculation Module						
Objective: To verify login module with correct email and password						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify form submission with all valid inputs	CGPA: 3.80	Should proceed to recommendation or next form	Form submitted successfully	Pass	—
		Subjects: Biology, Chemistry, Mathematics				
		Grades: A, A-, B+				
2	Verify form with non-numeric CGPA	CGPA: "abc"	Should prompt error “Invalid CGPA”	Error message shown	Pass	—
3	Verify form with CGPA out of range	CGPA: 4.50	Should prompt error “CGPA must be between 0.00 and 4.00”	Error message shown	Pass	—
4	Verify form submission with negative CGPA	CGPA: -2.00	Should prompt error “CGPA must be between 0.00 and 4.00”	Error message shown	Pass	—
5	Verify form with missing subjects or grades	Subjects: none or empty	Should prompt error “At least one subject and grade must be entered”	Error message shown	Pass	—
		Grades: —				
6	Verify form with duplicate subjects	Subjects: Chemistry, Chemistry	Should prompt error “Duplicate subjects not allowed”	Error message shown	Pass	—
		Grades: B, A				

Unit Testing 6: STPM Module						
Objective: To verify STPM Module on collecting user qualification information						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify form submission with all valid inputs	CGPA: 3.80 Subjects: Biology, Chemistry, Mathematics Grades: A, A-, B+	Should proceed to recommendation or next form	Form submitted successfully	Pass	-
2	Verify form with non-numeric CGPA	CGPA: "abc"	Should prompt error "Invalid CGPA"	Error message shown	Pass	-
3	Verify form with CGPA out of range	CGPA: 4.50	Should prompt error "CGPA must be between 0.00 and 4.00"	Error message shown	Pass	-
4	Verify form submission with negative CGPA	CGPA: -2.00	Should prompt error "CGPA must be between 0.00 and 4.00"	Error message shown	Pass	-
5	Verify form with missing subjects or grades	Subjects: none or empty Grades: —	Should prompt error "At least one subject and grade must be entered"	Error message shown	Pass	-
6	Verify form with duplicate subjects	Subjects: Chemistry, Chemistry	Should prompt error "Duplicate subjects not allowed"	Error message shown	Pass	-

Unit Testing 7: SPM Module						
Objective: To verify SPM module						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify form submission with all valid inputs	Subjects: Biology, Chemistry, Mathematics	Should proceed to recommendation or next form	Form submitted successfully	Pass	-
		Grades: A, A-, B+				
2	Verify form with missing subjects or grades	Subjects: none or empty	Should prompt error "At least one subject and grade must be entered"	Error message shown	Pass	-
		Grades: —				
3	Verify form with duplicate subjects	Subjects: Chemistry, Chemistry	Should prompt error "Duplicate subjects not allowed"	Error message shown	Pass	-

Unit Testing 8: Interest Form Module						
Objective: To verify the Interest Form module.						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify form submission with all valid inputs	Pick any answers	Should submit and redirect to Recommendations.	Form submitted successfully.	Pass	-
2	Verify behaviour when user tries to bypass interest form (e.g., modifying URL)	Change URL to "Recommendation" page link	Should prompt error or redirect back to form page.	Error message shown	Pass	-

Unit Testing 9: Recommendation Module						
Objective: To verify login module with correct email and password						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify form submission with all valid inputs	Fill in the previous form	Should be showing a top 3 groups of computer science fields	Recommendation shown.	Pass	-
2	Verify redirect to view program details	Click on any of the programme listed on the recommendation	Should be redirected to the appropriate page with programme detail	Redirected to programme detail page with correct information	Pass	-
3	Verify behaviour when user tries to access recommendation page without completing previous steps	Register a new account and modify link to "Recommendation" page	Should prompt error or redirect back to qualification page	Redirected to "Get-Started" section.	Pass	-

Unit Testing 10: View Programme Module						
Objective: To verify that users can view program details and that error handling works correctly						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify view of valid programme details	Program ID: 1 (Valid ID)	Should show full program details (name, university, description, requirements)	Program details displayed correctly	Pass	-
2	Verify behaviour with broken or invalid programme link	Program ID: 999	Should show “Program not found” or 404 error page	Error message displayed	Pass	-

Unit Testing 11: Search Programme Module						
Objective: To verify that users can search and filter programmes						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Verify user able to search programmes	Search: “Artificial Intelligence”	Should display programmes with containing the keywords	Programmes with “Artificial Intelligence” keyword is shown.	Pass	-
2	Verify user searching invalid keywords	Search: “XYZ123”	Should display “0 Search results” or “Search not found”	Error message shown.	Pass	-

Unit Testing 12: Favourite Module						
Objective: To verify user can add, delete, and view bookmarked programmes						
No	Test Case	Attribute and Values	Expected Result	Actual Outcome	Status	Reason (if fail)
1	Add program to favourites from “View Programme Details” page	Action: Click “Add to Favourite” button	Should display “Added to Favourite” confirmation and reflect in user’s favourites list	Program added and listed in favourites	Pass	-
		Program ID: 101				
2	Remove program from favourites	Action: Click “Remove from Favourite” button	Should display “Removed from Favourite” confirmation and no longer appear in favourites list	Program removed from list	Pass	-
		Program ID: 101				
3	View all favourite programmes	Action: Navigate to “My Favourites” page	Should display a list of all bookmarked programs with correct details	Favourite list displayed correctly	Pass	-
4	Try adding the same program twice	Action: Click “Add to Favourite” on same Program ID again	Should display a message like “Program already in Favourites” or disable the button	Prevented duplicate addition	Pass	-

APPENDIX E – Usability Testing Results

Module: Registration Module					
Objective: To evaluate if a new user can easily and successfully create an account without confusion or errors.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	New user registers	1. Ask user to open URL 2. Register account	User finds “Sign Up” button, completes form, and successfully create account.	Users successfully register an account	Pass
2	Incorrect input handling	1. Enter mismatched passwords	User understands the error message and knows what to do next.	Users understands and corrects credentials	Pass

Module: Login Module					
Objective: To determine if a registered user can log into their account without difficulty.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	A user with an existing account logs in	1. Ask the user to log in	User logs in without confusion	Users able to log in successfully	Pass
2	A user with an existing account logs in but with incorrect credentials	1. Ask the user to enter wrong password	User sees clear error and able to recover.	Users understood the message	Pass

Module: Select Qualification Module					
Objective: To assess if the user understands the purpose of this page and can make a selection without hesitation.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User selects academic background after login	1. Direct user to qualification page 2. Ask user to select a qualification	Users understand the page and choose quickly	Users understood and selected	Pass

Module: Diploma Form Module					
Objective: To check if a user with a Diploma qualification can fill in their academic details accurately and efficiently.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User fills out Diploma form.	1. Direct user to select "Diploma" 2. Instruct user to complete and submit the form	Users understand fields and submit without issues	Users completed the form; Some users suggested to implement tooltips	Pass
2	User fills out Diploma form incorrectly	1. Direct user to select "Diploma" 2. Instruct user to enter CGPA "35"	Users see clear error and know how to proceed	Users understood the error and corrected it	Pass

Module: Matriculation Module					
Objective: To evaluate the ease of use of the overall form for a Matriculation leaver.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	Fill out Matriculation form correctly	1. Direct user to select "Matriculation" 2. Instruct user to enter CGPA and add subjects	Users enter CGPA and manage subjects easily	Users filled in the form successfully	Pass
2	User fills out Matriculation form incorrectly	1. Direct user to select "Matriculation" 2. Instruct user to add duplicate subjects	Users see error and know how to fix it	Users understood and corrected the error	Pass

Module: STPM Module					
Objective: To ensure the STPM form is just as intuitive as the Matriculation form.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	STPM student inputs results	1. Direct user to select "STPM" 2. Instruct user to fill out and submit the form	Users input all data correctly and smoothly	Users successfully filled in the form	Pass
2	User fills out STPM form incorrectly	1. Direct user to select "STPM" 2. Instruct user to enter duplicate subjects	Users understand the error and know what to do	Users understood error and corrected it	Pass

Module: SPM Module					
Objective: To verify that the user understands the requirement to fill in their SPM results and can do so easily.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User enters SPM results after main qualification	1. Direct user to go to SPM form 2. Instruct user to enter subjects and grades	User understands the need to enter SPM Form is clear and easy to use	Users filled the form without help; Some were unsure of subject count	Yes
2	User fills out SPM form incorrectly	1. Direct user to go to SPM form 2. Instruct user to enter duplicate subjects	Users see error and know what to do next	Some users did not understand the error	Fail

Module: Interest Form Module					
Objective: To assess the clarity of the interest quiz questions and the intuitiveness of the interface.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User answers interest-based quiz	1. Direct user to open the Interest Quiz 2. Instruct user to answer all questions	User understands the questions and navigates smoothly	Users understood what to do; Some user suggested adding hints to the questions.	Pass
2	User tries to correct previous answers	1. Instruct user to retake the quiz and change question 5	User knows how to navigate and update answers	Users successfully navigated and updated their answer	Pass

Module: Recommendation Module					
Objective: To determine if the user can understand the recommendation results page.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User views recommended programs	1. Direct user to open the Recommendations page 2. Instruct user to explore the page and locate a program	User understands layout, terminology, and clickable elements	Users interacted successfully; Some users suggested adding hover effects for clearer interface.	Pass
2	User returns to “Recommendations” page.	1. Direct user to go to Home 2. Instruct user to navigate back to Recommendations	User navigates without help	User returned without assistance	Pass

Module: View Programme Module					
Objective: To evaluate if the user can find key information on the details page and use its features.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User checks a program's detail from Recommendation Page	1. Direct user to open the Recommendations page 2. Instruct user to click any programme 3. Ask if the information is understandable	User finds information easily and understands it	Users understood the information and found it sufficient	Pass
2	User checks for programme's detail from Search Page	1. Direct user to open the Search page 2. Instruct user to locate programme duration 3. Instruct user to save a programme	User understands they can view and use programme details	Users understood how to view and interact with programme details	Pass

Module: Search Programme Module					
Objective: To test if a user can successfully search for a program and understand the results.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User searches for a non-recommended program	1. Direct user to go to the Search page 2. Instruct user to search for a program using keywords	User finds search bar and performs a search	Users were able to search for programmes	Pass
2	User checks for programme's detail from Search Page	1. Direct user to go to the Search page 2. Instruct user to click on any result	User understands results are clickable	Users were able to access the "View Programme" page	Pass
3	User filters out programmes based on features they want	1. Direct user to go to the Search page 2. Instruct user to filter programmes by location	User filters results based on given criteria	Users were able to apply filters	Pass

Module: Favourite Module					
Objective: To verify that the user can access their list of saved programs and successfully manage it.					
No	Test Scenario	Test Steps	Expected Outcome	Actual Results	Pass / Fail
1	User adds a programme to their Favourites list.	1. Direct user to open the “Recommendations” page 2. Instruct user to click a programme 3. Instruct user to add it to Favourites 4. Instruct user to navigate to the “Favourites” page	User easily adds programme and finds it in Favourites	User able to add programmes and navigate to Favourite section	Pass
2	User accesses and manages saved programs	1. Direct user to open the “Favourites” page 2. Instruct user to remove a saved programme	User finds the page easily and removes a programme using the icon	Users removed programmes without issue.	Pass
3	User views programme details from the Favourites page.	1. Direct user to open the “Favourites” page 2. Instruct user to view any saved programme	User understands how to access programme details from the list	Users accessed programme details successfully	
4	User adds an already existing programme in their favourites list	1. Direct user to view a programme’s detail page 2. Instruct user to add it to Favourites twice	User understands error message and realizes the programme is already saved	Users understood the error message	Pass

APPENDIX F – Usability Testing & User Acceptance Testing Form

Title: Usability Testing Form – Recommendation System for University Programmes

Introduction:

Welcome!

I'm *Mohammad Khairul Ikhwan bin Abdul Basit*, a final-year Bachelor of Software Engineering student at UNIMAS.

This session is for the **Usability and User Acceptance Testing** of my final-year project, a **Recommendation System for University Programmes**. The goal is to evaluate the system's user-friendliness and collect feedback on its effectiveness.

The form has 9 sections and takes approximately **10–15 minutes** to complete. All responses are confidential and used solely for academic purposes. For questions, please contact: *78178@siswa.unimas.my* (Ikhwan)

Section 1: User Information

1. Please select the option that best describes your current status:
 - a. Unenrolled in University (Graduated from Diploma, Matriculation, or STPM)
 - b. Currently studying in Diploma, Matriculation, or STPM
 - c. Graduated with degree / Studying in University
2. Will you be continuing your study (to University)?
 - a. Yes
 - b. No

Section 2: User Account Management (Registration & Login)

1. The registration process was clear and easy to understand. *(Rate 1 to 5)*
2. Logging in to the system was straightforward and quick. *(Rate 1 to 5)*
3. The system's feedback when entering incorrect registration and login details was clear. *(Rate 1 to 5)*

Section 3: Qualification Selection & Input

2.1 Qualification Selection

1. It was easy to select my qualification (Diploma, Matriculation, STPM). *(Rate 1 to 5)*

2.2 Qualification Forms (Diploma, Matriculation, STPM, SPM)

2. The layout of the qualification input forms was clear and organized. *(Rate 1 to 5)*

3. It was easy to input my academic details (CGPA, subjects, grades). *(Rate 1 to 5)*
4. The error messages for incorrect or missing inputs were helpful. *(Rate 1 to 5)*
5. Overall, I found the process of entering my academic qualifications to be user-friendly. *(Rate 1 to 5)*

Section 4: Interest Quiz Form

1. The questions in the interest form were easy to understand. *(Rate 1 to 5)*
2. It was easy to select my answers on the interest form. *(Rate 1 to 5)*
3. It was easy to navigate around the interest form. *(Rate 1 to 5)*

Section 5: Recommendation System

4.1 Program Recommendation

1. The presentation of the recommended programs was clear and easy to understand. *(Rate 1 to 10)*

4.2 Program Details Viewing

2. It was easy to view the detailed information for a recommended program. *(Rate 1 to 5)*
3. The program details provided were sufficient and helpful. *(Rate 1 to 5)*

Section 6: Program Search & Filter

1. The search functionality allowed me to easily find programs. *(Rate 1 to 5)*
2. The search functionality was clear and easy to understand. *(Rate 1 to 5)*

Section 7: Favourite Programs

1. It was easy to add programs to my favourites list. *(Rate 1 to 5)*
2. It was easy to remove programs from my favourites list. *(Rate 1 to 5)*
3. Viewing my list of favourited programs was straightforward. *(Rate 1 to 5)*

Section 8: User Acceptance

1. The programmes recommended were relevant to my personal interests and preferences. *(Rate 1 to 5)*
2. The programmes recommended were relevant to my academic background. *(Rate 1 to 5)*

3. The details provided in the 'View Programme' page were informative and useful. *(Rate 1 to 5)*
4. The recommendation results were helpful for future study planning. *(Rate 1 to 5)*
5. The complete set of features was useful for university planning. *(Rate 1 to 5)*
6. The system effectively addresses the challenges of finding and choosing a university programme. *(Rate 1 to 5)*

Section 9: General Comments & Suggestions

Do you have any other comments, suggestions, or issues you encountered during your use of the system?

(Free-text response)

APPENDIX G – Link Hosted

During the testing period, the system was hosted online and accessible to participants at:

<https://www.unireco.duckdns.org>

Note: This deployment was temporary and intended solely for evaluation purposes during the Final Year Project 2 (FYP2) timeline. The system was accessible only between June 2025 and August 2025 and may no longer be available after this period.

APPENDIX H – Comments during Testing

General Comments & Suggestions:

Do you have any other comments, suggestions, or issues you encountered during your use of the system? Please elaborate.

11 responses

Consider adding a user guide to assist individuals who are not tech-savvy in navigating the system more efficiently, as some website UIs do not clearly indicate what actions users should take

Everything seems perfect, except for the search function. it has a tiny flaw which i couldnt search any uni based on the location.

For the spm certificates it will be good if it were in some sort of table for easier form fill.
for the search program part it will be great if it has a filter for me to search universiti and programme.

make user guide and like give more info guidelines for the website

Great idea!

Need to add hover effect. Quite confused whether some of the program can be select or not

Please add more programmes.

For the Section 'What are your preferences?', please put the scale numbers with words from Satisfied to

General Comments & Suggestions:

Do you have any other comments, suggestions, or issues you encountered during your use of the system? Please elaborate.

11 responses

Great idea!

Need to add hover effect. Quite confused whether some of the program can be select or not

Please add more programmes.

For the Section 'What are your preferences?', please put the scale numbers with words from Satisfied to Not Very Satisfied.

mantap

- the likert scale please include 1-5 description. Example 1- not relate all, 2- slightly relate, 3- neutral, 4- relatable, 5- very relate.
- password must include special char
- overall the flow is very good. Easy to interact without confusion.

all good