
Software Design Descriptions

---

# Sistem Akreditasi Program Akademik UMT

**By: Ahmad Mu'az Nabil bin Mohamad Noor Zawawi**

**S65752**

**Supervised by: Dr Mohamad Nor bin Hasan**

---

CSF4984 FINAL YEAR PROJECT I

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS
UNIVERSITI MALAYSIA TERENGGANU

## TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

The Software Design Description (SDD) document explains how the system will be built based on the requirements. It acts as a guide for developers and designers to understand what the system needs to do and how it should be created. The SDD includes details about the system's structure, how the parts of the system interact, how data moves through the system, and what methods or processes will be used to achieve its goals. This helps everyone involved in the project to work in the same direction and make sure the design is clear and well-planned.

The document is divided into sections to make it easy to follow. One section explains the overall structure of the system, showing how the main parts connect and work together. Another section goes into more detail, describing the features, actions, and data of each part. Diagrams like sequence diagrams and class diagrams are used to show how things work step by step. This helps developers turn ideas into a working system that matches the requirements.

## 1.1 Purpose

Software Design Description (SDD) For the UMT Academic Program Accreditation System, describes how parts like program management, evaluator handling, and accreditation tracking are designed to work together. It also considers things like making the system easy to maintain, fast, and ready for future needs. The SDD is not just for building the system; it is also useful for fixing issues, upgrading, or adding new features later.

## 1.2 Scope

The software product to be produced is the **UMT Academic Programme Accreditation System**.

**What the Software Will Do:**

- The system will enable the internal staff of Pusat Pengurusan dan Penjaminan Kualiti (PPPK) at UMT to manage the accreditation process for academic programs.
- The software will facilitate tasks such as program registration, internal evaluation, tracking accreditation progress, and updating the system with feedback.
- It will support the senior administrative officers, deputy directors, and other relevant staff in efficiently executing their responsibilities regarding the accreditation and quality assurance processes.
- The system will provide reports, dashboards, and automated notifications to ensure smooth operations for stakeholders involved in accreditation activities.

**What the Software Will Not Do:**

- The system will not handle processes outside the accreditation scope of UMT.
- It will not involved with thid party only within PPPK Staff.
- The system will not be used for payment gateway to pay for the fine or certificate payment

This system will support PPPK staffs and administrator managing the process of the Academic Program Accreditation for UMT. This system will simplified the process of Registration of new Program, Managing payment records, appointing Internal Evaluators and many more.Its overall goal is to streamline the management of Accreditation.

## 1.3 Definition, Acronyms and Abbreviations

The Definition, Acronyms, and Abbreviations section provides clarity on the terminology, acronyms, and abbreviations used throughout the document.Proofreading.org (2023) It ensures that all readers, including stakeholders and developers, have a clear understanding of the specific terms and acronyms that may be unfamiliar or ambiguous. This section serves as a reference guide, enabling a consistent interpretation of key concepts, system components, and abbreviations used within the context of the UMT Academic Programme Accreditation System. By defining these

terms upfront, the document becomes more accessible and easier to follow, minimizing the risk of misunderstandings and improving communication among all parties involved.

### 1.3.1 Definitions

| **Term** | **Definition** |
|---|---|
| Academic Program | A course of study offered by UMT leading to a degree, diploma, or certificate. |
| Accreditation | The process of verifying that an academic program meets specific quality standards set by the Malaysian Qualifications Agency (MQA). |
| Evaluator | A qualified individual responsible for assessing and validating the quality of an academic program. |

Table 1.1: Definitions

### 1.3.2 Acronyms

| **Acronym** | **Definition** |
|---|---|
| MOHE | Ministry of Higher Education |
| MQA | Malaysian Qualifications Agency |
| PA | Provisional Accreditation |
| FA | Full Accreditation |
| PPPK | Pusat Pengurusan dan Penjaminan Kualiti (Center for Management and Quality Assurance) |
| UMT | Universiti Malaysia Terengganu |
| SRS | Software Requirements Specification |

Table 1.2: Acronyms

### 1.3.3 Abbreviations

| Abbreviation | Definition |
|---|---|
| ERD | Entity Relationship Diagram |
| UML | Unified Modeling Language |
| DB | Database |
| UI | User Interface |
| QA | Quality Assurance |

Table 1.3: Abbreviations

## 1.4 Overview

This Software Design Document (SDD) provides a comprehensive description of the design and architecture of the **UMT Academic Programme Accreditation System**. It outlines how the system will be structured, its components, and the design decisions made to ensure the system meets its objectives.

The document is organized into the following chapters:

- **Chapter 1: Introduction** – This chapter introduces the purpose, scope, and objectives of the **UMT Academic Programme Accreditation System**. It also outlines the structure of the document and provides an overview of the key sections covered.
- **Chapter 2: System Architecture** – This chapter details the high-level architecture of the system, including its components, interactions, and the technologies used.
- **Chapter 3: Database Design** – Provides detailed information about the database structure, including normalization, entity relational diagrams, and data dictionaries.
- **Chapter 4: UI/UX Design** – This chapter explains the design of the user interface and user experience, including the screen layout, objects, and actions.
- **Chapter 5: Conclusion** – Summarizes the key findings of the document and provides any final remarks or recommendations.

Each chapter of this document builds on the previous one, providing a detailed breakdown of the system's design. This structure ensures that the document is easy to navigate and that all relevant aspects of the system's design are covered in a clear and systematic manner.

# CHAPTER 2

# SYSTEM ARCHITECTURAL DESIGN

## 2.1 Architectural Design

The architecture of the **UMT Academic Programme Accreditation System** is designed to be modular, ensuring each subsystem is responsible for a specific functionality. The main subsystems of the system are divided into the Frontend, Backend, and Container, with each having clear roles and responsibilities.

- **Frontend:**
  The frontend is developed using React and Vite, allowing for a fast, responsive, and modern user interface. It communicates with the backend to request and display data.

- **Backend:**
  The backend is built with Node.js and Express. It handles business logic, processes user requests, and communicates with the database. The backend acts as a mediator between the frontend and the container.

- **Container:**
  The container is managed with Docker to allow the system to be containerized and run consistently across different environments. The database, MySQL, is stored in the container and used to persist data.

The following diagram shows the key components of the system and how they interact:

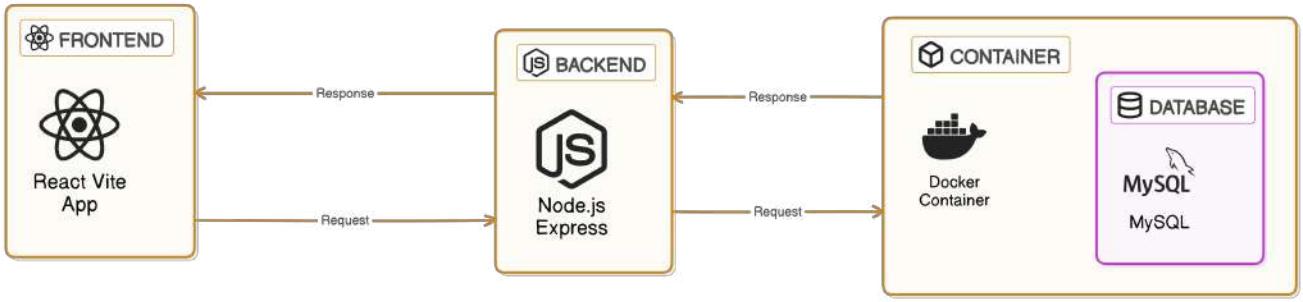


Figure 2.1: System Architecture Diagram

This diagram illustrates the modular design of the system, where each component interacts through well-defined connections. The frontend makes requests to the backend, which processes these requests and communicates with the container (and subsequently the database).

The system is designed to be scalable, maintainable, and easily extensible by following this modular structure.

## 2.2 Design Rationale

The architecture described above was chosen for several key reasons:

- **Modularity**: The system is divided into distinct subsystems, each responsible for a specific functionality. This separation of concerns makes it easier to maintain and extend individual parts of the system without affecting others.
- **Scalability**: The use of Docker containers for the database and backend ensures that the system can be easily scaled horizontally or vertically. The modular structure also supports the easy addition of new features or services in the future.
- **Flexibility**: Using React for the frontend allows for a highly interactive and dynamic user interface, while Node.js with Express provides a lightweight and efficient backend. This combination supports both performance and flexibility.
- **Ease of Deployment**: Containerizing the system with Docker simplifies deployment across different environments, ensuring consistency in how the system runs.

Other architectures, such as MVC (Model-View-Controller) or microservices-based architectures, were considered but were not chosen. The MVC approach was dismissed due to its potential difficulty in integrating with React for the frontend and Node.js for the backend. The separation of concerns in MVC would make it challenging to implement

efficiently with the technologies chosen for the system. Microservices were considered but not selected as they would introduce unnecessary complexity at this stage of the project.

The current architecture strikes a balance between simplicity, scalability, and ease of maintenance, making it the most suitable choice for the system's requirements.

# CHAPTER 3

# DATABASE DESIGN

## 3.1 Normalization

Normalization is the process of organizing data in a database. It includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.helenclu (2024)

Normalization ensures that the data model of the UMT Academic Program Accreditation System is both flexible and reliable. By organizing the database into structured tables, normalization minimizes redundancy and dependency, leading to efficient data storage and retrieval. While normalization might result in more tables, it enables seamless management and interaction with data, ensuring the system can handle complex queries and operations without requiring frequent adjustments.

In the context of the UMT system, normalization ensures that key entities such as programs, evaluators, accreditation records, payment details, and feedback are logically structured. For example, separating evaluators, programs, and accreditation details into distinct tables eliminates duplication and ensures consistency across the system. This approach allows the system to handle changes or additions, like updating an evaluator's information or adding a new program, without disrupting existing data or requiring extensive modifications.

### 3.1.1 Zero Normal Form: 0NF

Present the unnormalized form.

| userId | userName | roles | userPassword | staffId | ProgramId | programName | tahapMQF | codeNEC | modePenwaran | fakulti | jenisPengajian | tempohMinimum | tempohMaksimu | modPenyampaian | StrukturProgram | ProgramKerjasama | MQANumber | EvaluationID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | mohamadNoor | Administrator | | 1 | | | | | | | | | | | | | | |
| 2 | fauziyanIbrahim | Administrator | | 1 | | | | | | | | | | | | | | |
| 3 | robbiyahMohd | Staff | | 1 | 1 | Cloud Computing Security | 6 | 613 | Kerja Kursus | FSKM | Sepenuh Masa | 3 tahun 6 Bulan | 7 Tahun | Konvensional | Major | Tidak | MQA7223 | 1 |

Figure 3.1: Unnormalized Form(0NF)


Figure 3.2: Unnormalized Form(0NF)

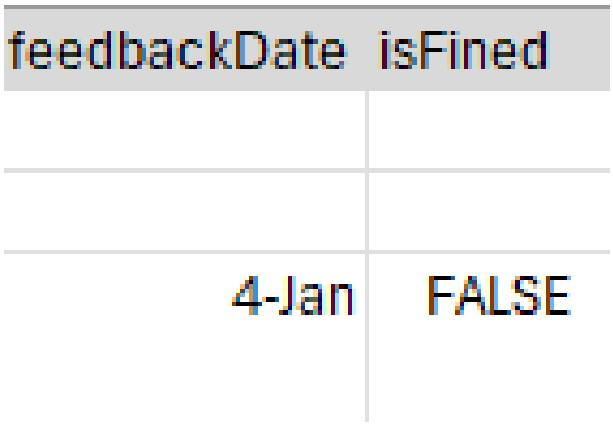


Figure 3.3: Unnormalized Form(0NF)

### 3.1.2 One Normal Form: 1NF

In the first normal form, an entity contains no repeating groups and all attributes must have a unique name.

- 1NF for User

| userId | userName | roles | userPassword |
|---|---|---|---|
| 1 | mohamadNoor | Administrator | (encrypted) |
| 2 | fauziyanilbrahim | Administrator | (encrypted) |
| 3 | robbiyahMohd | Staff | (encrypted) |

Figure 3.4: 1NF for table User

Above table is the user table that has been normalized to 1NF. the primary key is UserId. The user table didnt contain any repeating data.

- 1NF for Program

| ProgramId | programName | tahapMQF | codeNEC | modePenawaran | fakulti | jenisPengajian | tempoh Minimum | tempoh Maksimu | modPenyampaian | StrukturProgram | ProgramKerjasama | MQANumber |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Cloud Computing Security | 6 | 613 | Kerja Kursus | FSKM | Sepenuh Masa | 3 tahun 6 Bulan | 7 Tahun | Konvensional | Major | Tidak | MQA7223 |

Figure 3.5: 1NF for Program

The table above show 1NF for program's table. The primary key is programID.There is no repeating data in the table, so the table are already in 1NF.

- 1NF for Evaluator

| EvaluatorID | Evaluator Name | Evaluator Faculty | Evaluator Status | Evaluator Phone | Evaluator Email | Program ID |
|---|---|---|---|---|---|---|
| 1 | Dr. Rabei Mamat | FSKM | Accept | 199085545 | rab@umt.edu.my | 1 |

Figure 3.6: 1NF for Evaluator

The 1NF table above is Evaluator table. The primary key is ProgramID and it have ProgramID as the foreign key referencing to Program table.No repeating data is in this table.

- 1NF for Accreditation

| AccreditationId | ProgramId | uploadDate | accreditationType | accreditationFilePath |
|---|---|---|---|---|
| 1 | 1 | 4-Jan | FA | images/accreditation/PA/CloudComputing.pdf |

Figure 3.7: 1NF for Accreditation

The accreditation table above is 1NF. The primary key is accreditationID while the foreign key is programID referenced to the Program table.

- 1NF for Payment

| PaymentId | ProgramId | Amount | PaymentType | PaymentStatus | receiptPath |
|---|---|---|---|---|---|
| 1 | 1 | RM 500.00 | Accreditation Certificate | Paid | images/receipt/CloudComputing.pdf |

Figure 3.8: 1NF for Payment

The table payment contain paymentID as the primary key for the table and programID as Foreign key, referenced towards program table.

- 1NF for feedback

| feedbackId | ProgramId | comments | feedbackDate | isFined |
|---|---|---|---|---|
| 1 | 1 | Pihak UMT perlu mempunyai strategi dalam memastikan terdapat permintaan kemasukan... | 4-Jan | FALSE |

Figure 3.9: 1NF for feedback

The table feedback contain feedbackID as the primary key for the table and programID as Foreign key, referenced towards program table.

### 3.1.3 Second Normal Form: 2NF

An entity with a primary key consisting of only one attribute is in the second normal form (2NF).

- 2NF for table user

| userId | userName | roles | userPassword |
|---|---|---|---|
| 1 | mohamadNoor | Administrator | (encrypted) |
| 2 | fauziyanilbrahim | Administrator | (encrypted) |
| 3 | robbiyahMohd | Staff | (encrypted) |

Figure 3.10: 2NF for table user

The table user have no partial dependency. So, there will be no changes to the user's table.

- 2NF for Program

| ProgramId | programName | tahapMQF | codeNEC | modePenawaran | fakulti | jenisPengajian | tempoh Minimum | tempoh Maksimu | modPenyampaian | StrukturProgram | ProgramKerjasama | MQANumber |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Cloud Computing Security | 6 | 613 | Kerja Kursus | FSKM | Sepenuh Masa | 3 tahun 6 Bulan | 7 Tahun | Konvensional | Major | Tidak | MQA7223 |

Figure 3.11: 2NF for Program

The table above show 2NF for program's table. The above table have no different from 1NF because it already fulfilled 2NF rules .

- 2NF for Evaluator

| EvaluatorProgram | EvaluatorID | ProgramId |
|---|---|---|
| 1 | 1 | 1 |

Figure 3.12: 2NF for Evaluator

Created a separate table EvaluatorProgram to handle the many-to-many relationship between evaluators and programs. This avoids partial dependency on a composite key.

- 2NF for Accreditation

| AccreditationProgram | AccreditationId | ProgramId |
|---|---|---|
| 1 | 1 | 1 |

Figure 3.13: 2NF for Accreditation

Added a junction table Accreditation to manage the relationship between accreditation records and programs.

- 2NF for payment

| PaymentProgram | PaymentId | ProgramId |
|---|---|---|
| 1 | 1 | 1 |

Figure 3.14: 2NF for payment

Added a junction table PaymentProgram to link payments with programs.

- 2NF for feedback

| FeedbackProgram | feedbackId | ProgramId |
|---|---|---|
| 1 | 1 | 1 |

Figure 3.15: 2NF for feedback

Added a junction table FeedbackProgram to link feedback with programs.

### 3.1.4 Third Normal Form: 3NF

The third normal form means that no attribute within an entity is dependent on an non-prime attribute that, in turn, depends on the primary key.

- 3NF for paymentType

| paymentTypeId | paymentType |
|---|---|
| 1 | Accreditation Fee |

Figure 3.16: 3NF for paymentType

Removed redundancy by normalizing PaymentType into a separate table to eliminate transitive dependencies.

All the other tables already satisfied the 3NF normalizations. So all the table except payment remain the same.

## 3.2 Entity Relational Diagram

An entity relationship diagram (ER diagram or ERD) is a visual representation of how items in a database relate to each other.Belcic (2024) ERDs are a specialized type of flowchart that conveys the relationship types between different entities within a system. They use a defined set of symbols, including rectangles, ovals and diamonds, and link them with connecting lines.

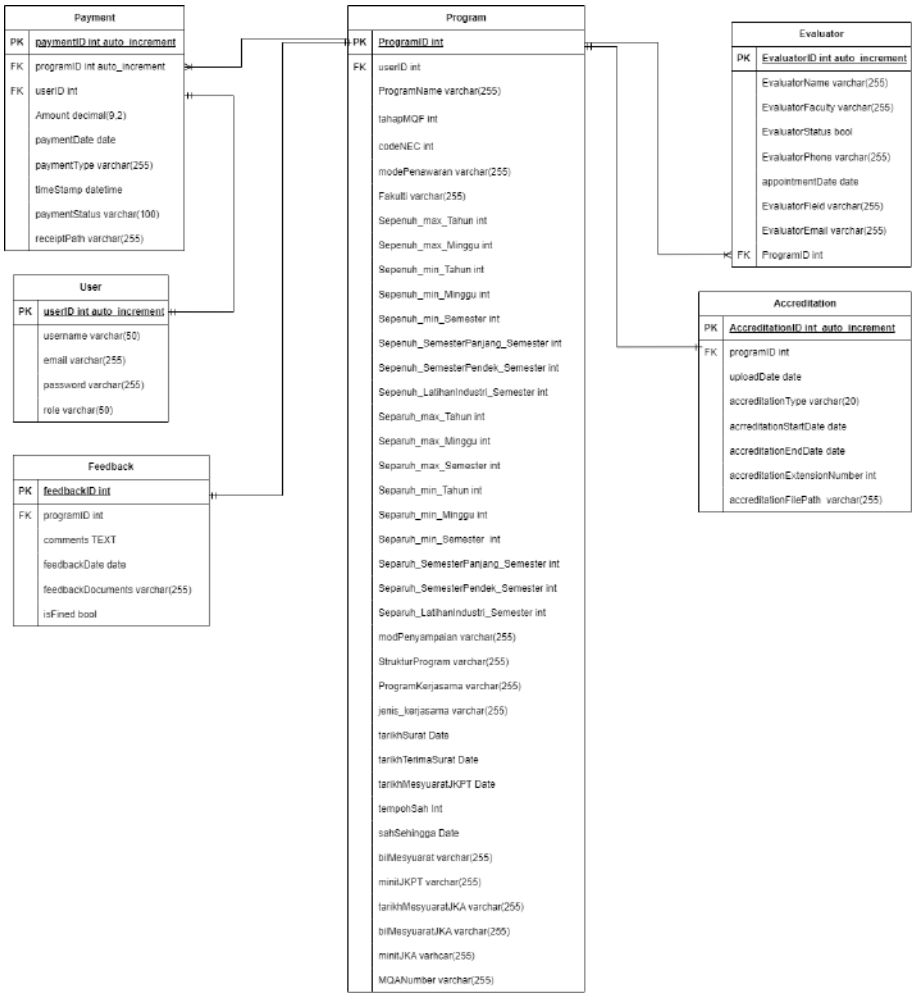


Figure 3.17: ERD Diagram for UMT Academic Program Accreditation System

## 3.3 Data Dictionary

Data Dictionary is a list of data elements (entity/table and attribute/column) with their attributes and descriptions. It has a form of a set of tables.

| Attribute Name | Data Type | Description |
|---|---|---|
| AccreditationId | Integer | Unique ID for each accreditation record. |
| accreditationFilePath | String | Path to the uploaded accreditation file. |
| accreditationType | String | Type of accreditation (e.g., PA, FA). |
| Amount | Float | Amount of payment made. |
| codeNEC | String | NEC code associated with the program. |
| comments | Text | Feedback or comments from MQA. |
| evaluatorEmail | String | Email address of the evaluator. |
| evaluatorFaculty | String | Faculty of the evaluator. |
| evaluatorID | Integer | Unique ID of the evaluator. |
| evaluatorName | String | Name of the evaluator. |
| evaluatorPhone | String | Phone number of the evaluator. |
| evaluatorStatus | String | Status whether he Reject or Accept. |
| fakulti | String | Faculty offering the program. |
| feedbackDate | Date | Date of the feedback submission. |
| feedbackId | Integer | Unique ID for feedback. |
| isFined | Boolean | Indicates whether a fine has been applied. |
| modePenwaran | String | Mode of offering (e.g., coursework). |
| modPenyampaian | String | Mode of delivery for the program. |
| MQANumber | String | Accreditation number assigned by MQA. |
| PaymentId | Integer | Unique ID for each payment record. |
| PaymentStatus | String | Status of the payment (e.g., Paid, Pending). |
| PaymentType | String | Type of payment (e.g., Accreditation Certificate). |
| ProgramId | Integer | Unique ID for the academic program. |
| programName | String | Name of the academic program. |
| ProgramKerjasama | Boolean | Indicates whether the program is collaborative. |
| receiptPath | String | Path to the payment receipt file. |
| roles | String | Role of the user (e.g., Administrator, Staff). |
| timeStamp | datetime | The time and date when of the payment update time and date. |
| accrreditationStartDate | date | The start date of the accreditation certificate. |
| accrreditationEndDate | date | The End date of the accreditation certificate. |
| accreditationExtensionNumber | int | The number of time of the accreditation time extension. |

Table 3.1: Data Dictionary

| Attribute Name | Data Type | Description |
|---|---|---|
| Sepenuh_max_Tahun | Integer | the amount of maximum years for full time . |
| Sepenuh_max_Minggu | Integer | the amount of maximum weeks for full time . |
| Sepenuh_min_Tahun | Integer | the minimum number of years required for a full-time program . |
| Sepenuh_min_Minggu | Integer | the minimum number of weeks required for a full-time program . |
| Sepenuh_min_Semester | Integer | the minimum number of semesters required for a full-time program . |
| Sepenuh_SemesterPanjang | Integer | the number of long semesters in a full-time program . |
| Sepenuh_SemesterPendek | Integer | the number of short semesters in a full-time program . |
| Sepenuh_LatihanIndustri | Integer | the number of internship semesters in a full-time program . |
| Separuh_max_Tahun | Integer | the maximum number of years allowed for a part-time program . |
| Separuh_max_Minggu | Integer | the maximum number of weeks allowed for a part-time program . |
| Separuh_min_Tahun | Integer | the minimum number of years required for a part-time program . |
| Separuh_min_Minggu | Integer | the minimum number of weeks required for a part-time program . |
| Separuh_min_Semester | Integer | the minimum number of semesters required for a part-time program . |
| Separuh_SemesterPanjang | Integer | the number of long semesters in a part-time program . |
| Separuh_SemesterPendek | Integer | the number of short semesters in a part-time program . |
| Separuh_LatihanIndustri | Integer | the number of internship semesters in a part-time program . |
| uploadDate | Date | Date when the accreditation file was uploaded. |
| userId | Integer | Unique ID for the user. |
| userName | String | Username of the user. |
| userPassword | String | Password for the user account. |

Table 3.2: Data Dictionary

| **Attribute Name** | **Data Type** | **Description** |
|---|---|---|
| feedbackDocuments | varchar(255) | Path of the feedback document in the MySQL server. |
| appointmentDate | date | The date of the appointment for the Evaluator. |
| EvalutorField | varchar(255) | The field of expertise of the Evaluator. |

Table 3.3: Data Dictionary

# CHAPTER 4

# UI/UX DESIGN

## 4.1 Overview of User Interface

The User Interface (UI) of the UMT Academic Program Accreditation System is designed to be intuitive, user-friendly, and efficient for all stakeholders. The system provides clear navigation and well-organized sections, allowing users to perform tasks seamlessly.

From the user's perspective, PPPK staff and administrators can access modules such as program registration, accreditation certificate management, payment records, feedback tracking, and evaluator assignments. The interface offers dedicated forms for data entry, real-time status updates, and notification features to keep users informed about pending tasks and upcoming deadlines. Users will be able to search, view, and update records easily, with interactive dashboards providing a high-level overview of all key activities.

Additionally, the system provides contextual feedback for user actions. For example, successful submissions, errors in form entries, and validation messages are displayed to guide users through their tasks. This ensures accuracy and transparency while minimizing errors and improving productivity. The UI prioritizes accessibility, ensuring that users can complete their work effectively while accessing all the necessary information and features at their fingertips.

## 4.2 Screen Images

- Program Registration

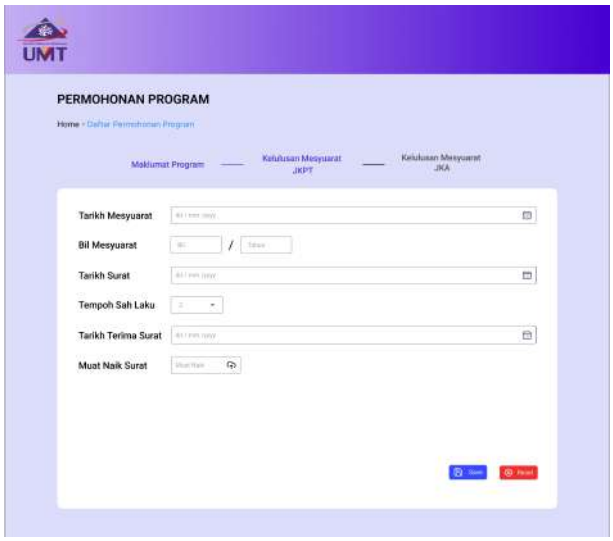


Figure 4.2: Program Registration: Mesyuarat JKPT

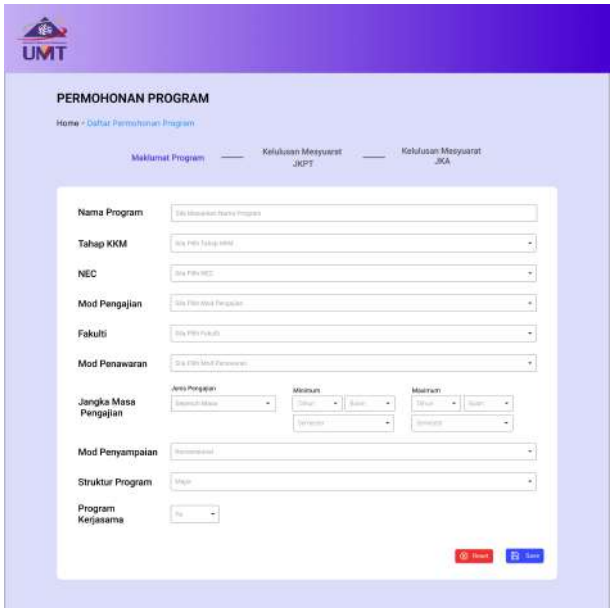


Figure 4.1: Program Registration: Maklumat Program

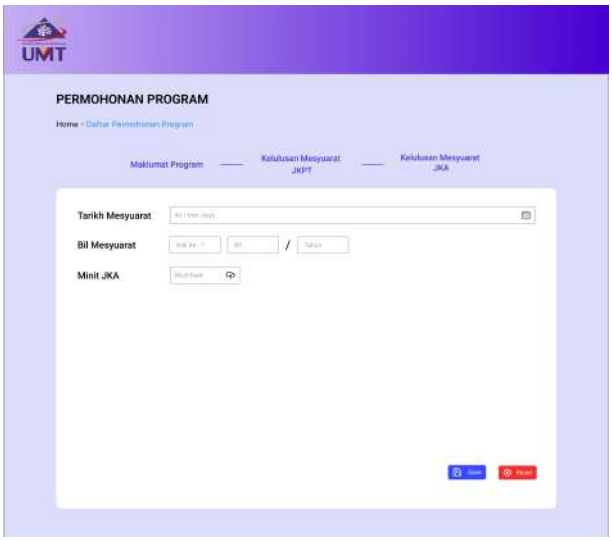


Figure 4.3: Program Registration: Mesyuarat JKA

Maklumat Program section focuses on collecting detailed information about the program being proposed. It includes fields such as the program name, qualification level (Tahap KKM), and NEC code. Additionally, it captures important details like the mode of study (e.g., full-time or part-time), faculty offering the program, and the mode of delivery (e.g., conventional or online). Fields for the minimum and maximum duration of study, along with program structure and whether it's a collaboration program, ensure that all key aspects of the program are recorded accurately.

Mesyuarat JKPT Section is tailored for recording the details of the approval meeting from the JKPT . Key fields include the meeting date, meeting number, and any associated documents. This ensures that all approvals and discussions from the JKPT meeting are tracked systematically and linked to the program application for reference and compliance purposes.

This section mirrors the functionality of the previous one but focuses on the JKA (Jawatankuasa Akademik) approval meeting. It includes fields to capture the meeting date, meeting number, and any minutes or documents that need to be uploaded. This provides a structured and transparent way to record the approval journey of the program through the JKA process, ensuring traceability and alignment with academic governance

## 4.3 Screen Objects and Actions

- **Home**

- Home Button: Navigates to the Home screen.
- Daftar Permohonan Program Button: Navigates to the ”Daftar Permohonan Program” screen.

- **Daftar Permohonan Program**
  - Maklumat Program Tab: Accesses the ”Maklumat Program” section.
  - Kelulusan Mesyuarat JKPT Tab: Accesses the ”Kelulusan Mesyuarat JKPT” section.
  - Kelulusan Mesyuarat JKA Tab: Accesses the ”Kelulusan Mesyuarat JKA” section.
  - Nama Program Input: Text input field for program name.
  - Tahap KKM Dropdown: Dropdown for selecting KKM level.
  - NEC Dropdown: Dropdown for selecting NEC code.
  - Mod Pengajian Dropdown: Dropdown for selecting mode of study.
  - Fakulti Dropdown: Dropdown for selecting faculty.
  - Mod Penawaran Dropdown: Dropdown for selecting mode of offering.
  - Jangka Masa Pengajian Dropdown: Dropdown for selecting duration of study.
  - Mod Penyampaian Dropdown: Dropdown for selecting mode of delivery.
  - Struktur Program Dropdown: Dropdown for selecting program structure.
  - Program Kerjasama Checkbox: Checkbox for indicating a collaborative program.
  - JKPT Meeting Date Input: Date input field for JKPT meeting date.
  - JKPT Meeting Number Input: Text input field for JKPT meeting number.
  - JKPT Document Upload: File upload field for JKPT documents.
  - JKA Meeting Date Input: Date input field for JKA meeting date.
  - JKA Meeting Number Input: Text input field for JKA meeting number.
  - JKA Document Upload: File upload field for JKA documents.
  - Save Button: Saves the entered information.
  - Reset Button: Resets all fields.

### 4.4 Actions

- Home Button Click: Navigates to the Home screen.

- Daftar Permohonan Program Button Click: Navigates to the ”Daftar Permohonan Program” screen.
- Maklumat Program Tab Selected: Displays the ”Maklumat Program” section.
- Kelulusan Mesyuarat JKPT Tab Selected: Displays the ”Kelulusan Mesyuarat JKPT” section.
- Kelulusan Mesyuarat JKA Tab Selected: Displays the ”Kelulusan Mesyuarat JKA” section.
- Save Button Click: Validates all fields, saves the entered data to the database, and displays a success message.
- Reset Button Click: Resets all fields to their default values.
- JKPT Document Upload Changed: Handles the uploading of JKPT documents.
- JKA Document Upload Changed: Handles the uploading of JKA documents.

# CHAPTER 5

# CONCLUSION

## 5.1 Conclusion

In summary, this Software Design Document (SDD) outlines the key components, design principles, and system architecture for the UMT Academic Program Accreditation System. The document provides a detailed overview of the database design, user interface (UI/UX) structure, and the rationale behind the system's architectural choices to ensure reliability, scalability, and usability.

The system is designed to address the current challenges of managing accreditation processes by automating workflows, ensuring data integrity, and enhancing accessibility for stakeholders such as PPPK staff and administrators. The focus on normalization, entity relationships, and structured data storage ensures that the system will handle accreditation data efficiently and effectively.

This SDD serves as a guide for developers to implement the system and as a reference for stakeholders to understand the system's design and functionality. By adhering to the specifications provided, the UMT Academic Program Accreditation System will support the institution in achieving its goals of streamlined operations and improved accreditation management.

## REFERENCES

Belcic, I. (2024). Entity relationship diagram. https://www.ibm.com/think/topics/entity-relationship-diagram.

helenclu (2024). Database normalization description - microsoft 365 apps. https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description

Proofreading.org (2023). Using abbreviations in academic writing. https://proofreading.org/blog/abbreviations-in-academic-writing/.