UTM

UNIVERSITI TEKNOLOGI MALAYSIA

DATABASE - SECD2523

Section 6

Group 03

DATABASE CONCEPTUAL DESIGN

(Phase 2)

NAME	MATRIC NO.
ABDELRAHMAN ELFATIH	A23CS4032
LEENA ATAELMANAN ELSIDDIG AWADALLA	A23CS4043
TAGWA BASHIR ABDULLA KUBUR	A23CS4057
MOHAMED MAHMOUD MAHMOUD RAMADAN	A23CS4045
MOHAMED ABDELGAWWAD ABDELGHANI MOUSTAFA	A23CS4025

List of Content

1.0 Introduction

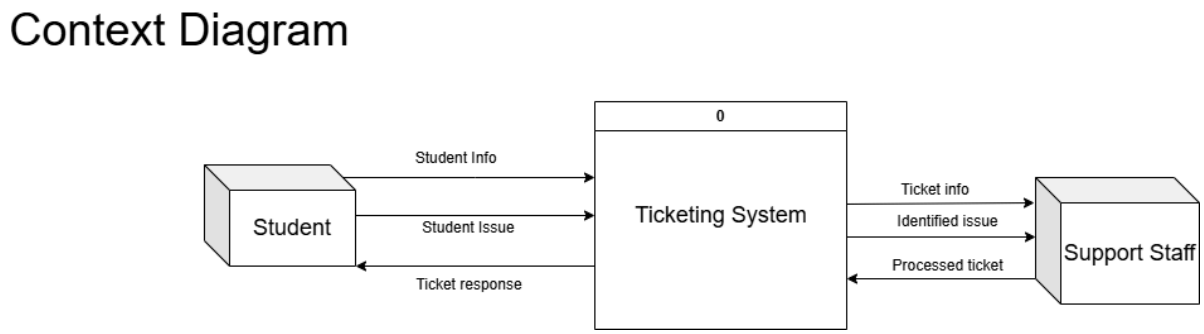
One of the most popular systems at UTM is UTM e-learning, however students frequently face challenges while using it .They typically attempt to contact the academic office or send an email for help when they run into problems, such as technical ones. However, the current e-learning system does not provide a specific space for answering students' questions directly through the UTM platform . This gap makes it harder for students to resolve their issues effectively and can lead to dissatisfaction and delays.Additionally, the current system places extra strain on support staff, who must respond directly to students or manage a lot of students' issues through their email, resulting in inefficiency and increased workload.

A well-designed system is essential for improving students' experience by providing quick and specific help for each of them . In an effort to address this, we had proposed an improved system that focuses on providing help and support for students in a structured and efficient way.The system will keep the process organized and ensure students can get help quickly while support staff can handle requests more efficiently. With this system, delays and unanswered questions are reduced, making the e-learning platform easier to use and more reliable in terms of technical support.

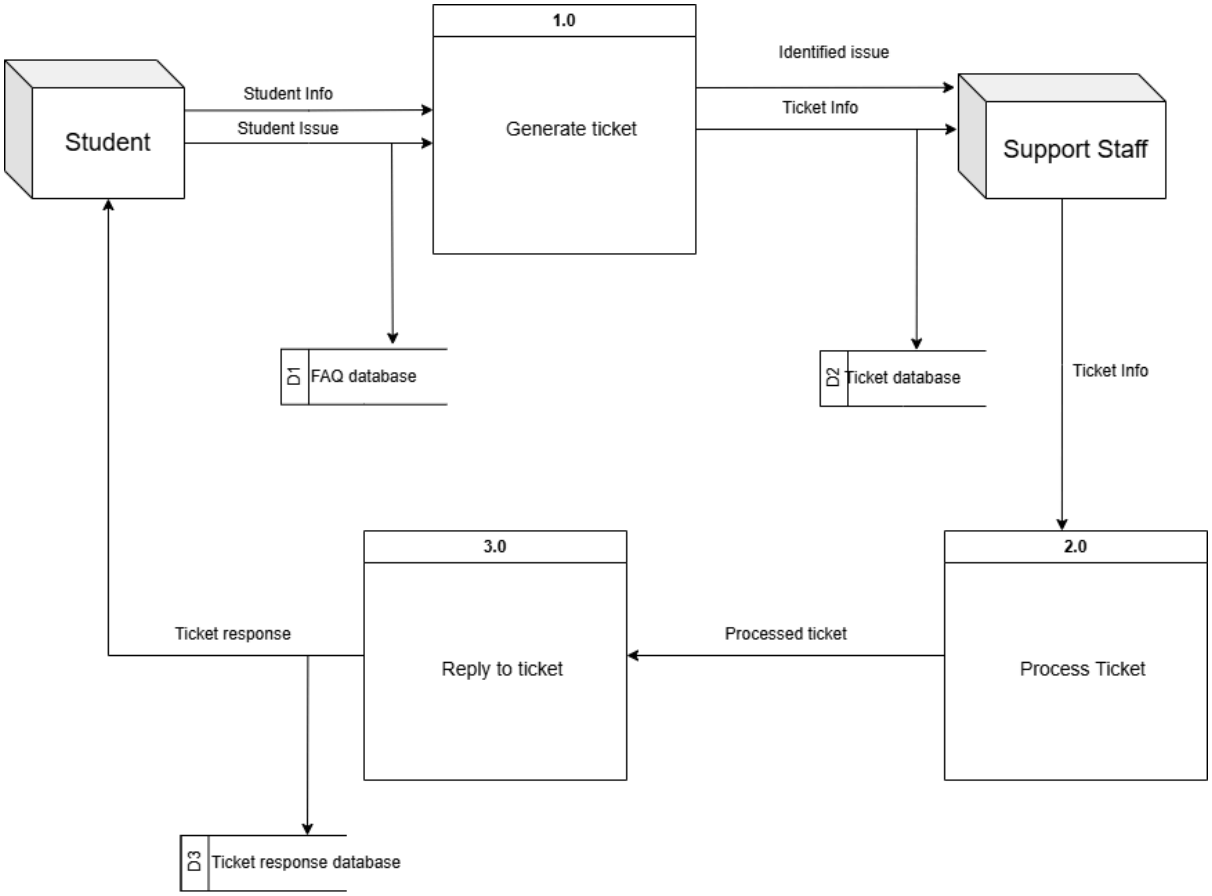
In this phase, we will focus on conceptual and structural design for the e-learning ticketing system. This includes, defining the entities, their attributes, and the relationships between them. To make the system easy to understand,We will also set the business rules and analyze the data requirements, including data types for each entity, we will create Data Flow Diagrams (DFDs) and Entity-Relationship Diagrams (ERDs) to show how the system works.This will provide a clear and strong foundation for building an effective system.

2.0 Data Flow Diagram (DFD)

2.1 Context Diagram

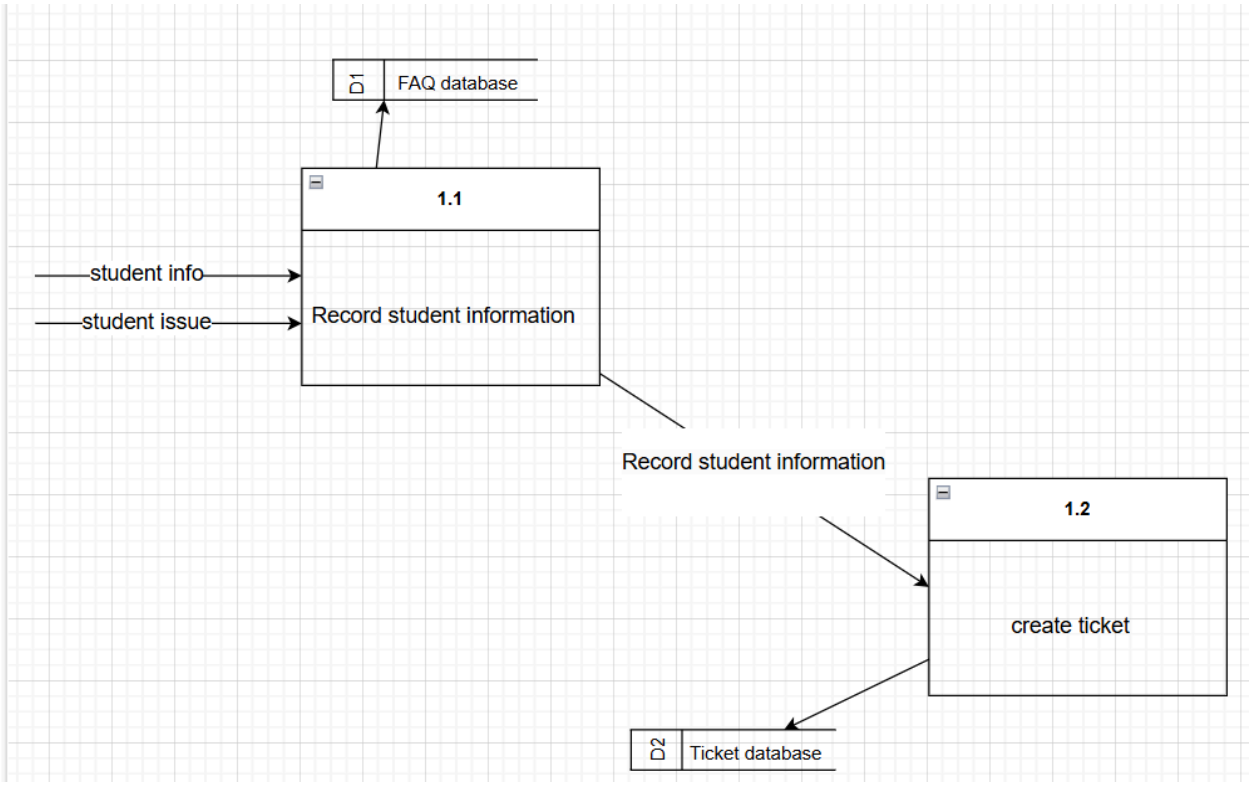


2.2 Parent Diagram (Level 0)

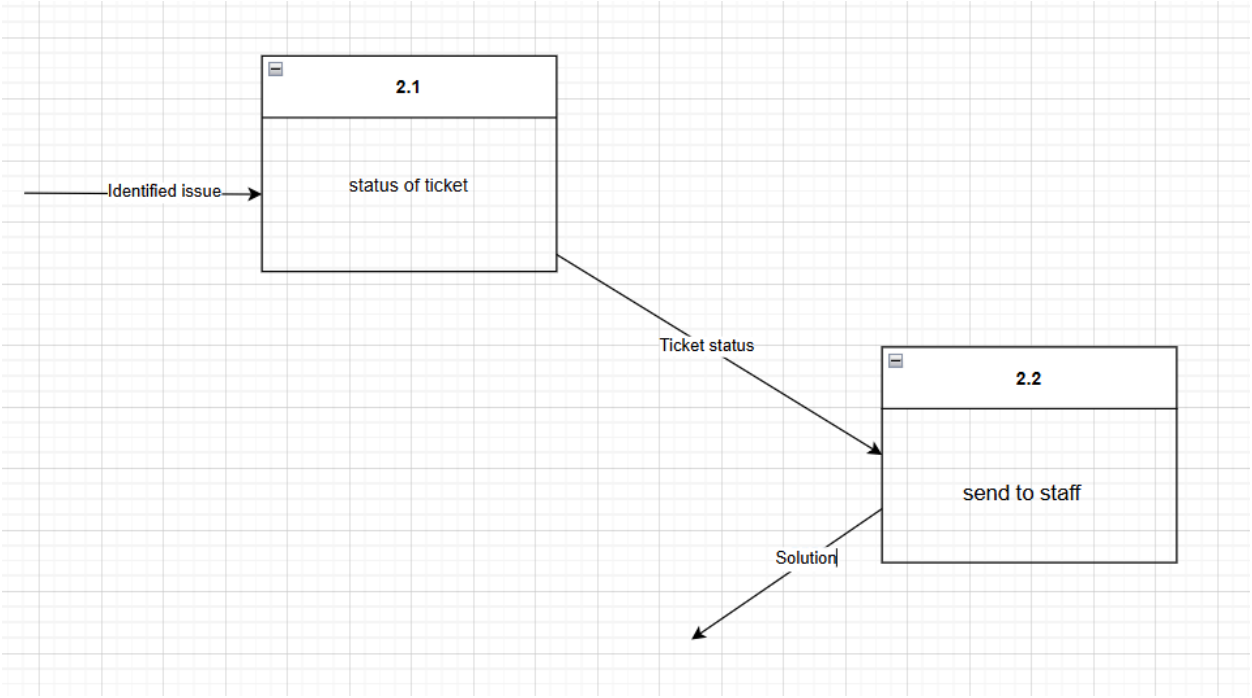


2.3 Child Diagram (Level 1)

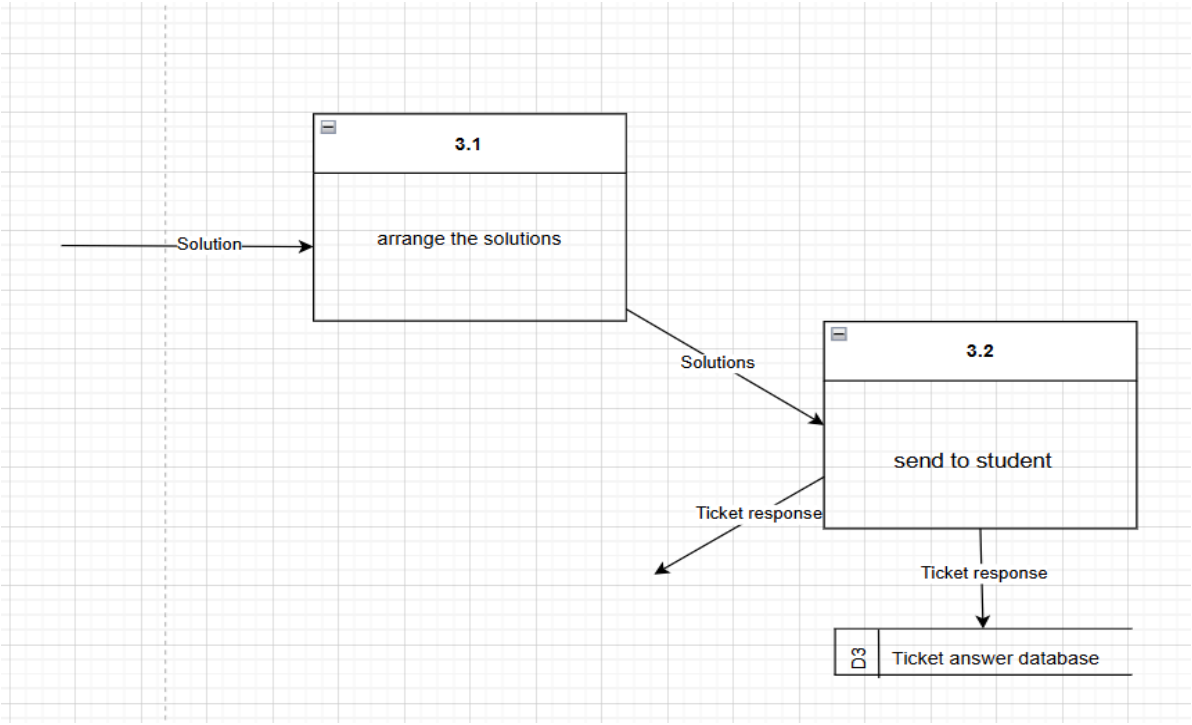
Child diagram for process 1



Child diagram for process 2



Child diagram for process 3


3.1
arrange the solutions
Solution
Solutions
3.2
send to student
Ticket response
Ticket response
D3
Ticket answer database

3.0 Data & Transaction Requirement

3.1 Proposed business rule

Student:

- Access FAQs and FAQs answers to find common issues before creating a ticket.
- Submit tickets with details such as subject description and urgency level.
- View the real_time status of submitted tickets (e.g., "In Progress," "On Hold", "Closed").
- Receive ticket responses provided by the support staff.
- View a history of submitted tickets and their responses.

Staff:

- View all submitted tickets, including student name and email , issue descriptions, and urgency levels.
- Assign tickets to themselves or other team members based on expertise and availability.
- Update the status of tickets reflecting progress(e.g., "In Progress," "On Hold," "Closed").
- Create a detailed ticket response .
- Create FAQs and FAQs answers .

Administrator :

- View FAQs and FAQs answers.
- View response to tickets with the time details .

3.2 Proposed data & transactional

3.2.1 Proposed Data Requirement

Student

The **Student** entity stores data such as student ID, name, and email address. The student ID acts as the primary key and uniquely identifies each student. The name is a composite attribute consisting of the first name and last name. Each student can submit zero or many tickets, receives zero to many ticket response, views zero or many FAQs and associated zero or many FAQ answers, and checks the status of zero or many submitted tickets.

Support Staff

The **Support Staff** stores data such as staff ID, staff name, email address, and department. The staff ID serves as the primary key, ensuring each staff member is uniquely identified. The name is a composite attribute, consisting of first name and last name. Support staff receives zero or many tickets, creates zero or many ticket responses, updates zero or many ticket status, creates zero or many FAQS and zero or many FAQS answers associated.

FAQs

The **FAQs** entity stores data such as FAQ ID, question_type and the question. FAQ ID represents the primary key. FAQs have only one associated FAQ answer.

FAQ Answers

The **FAQ Answers** entity holds data such as answer ID, FAQ ID, and the answer. Answer ID is defined as the primary key, FAQ ID is defined as a foreign key. The FAQ answer is associated with only one FAQ, viewed by zero or many students and created by one support staff.

Ticket

The **Ticket** entity holds data such as ticket ID, student ID, staff ID, subject, issue description, created at and ticket urgency. The ticket id is defined as the primary key, the student id and staff id are defined as foreign keys. The ticket has only one ticket status associated, and has only one ticket response.

Ticket Status

The **Ticket Status** entity holds data such as status ID, ticket ID, status and updated at. The status ID is defined as the primary key. The ticket ID is defined as a foreign key. Ticket status is associated with only one ticket, checked by only one student, updated by one or many support staff.

Ticket Response

The **Ticket Response** entity holds data such as response ID, ticket ID, staff ID, response text and the response time. The response ID is defined as the primary key. The ticket id and staff ID are defined as the foreign keys. Ticket response is associated with only one ticket, created by only one staff member, and received by only one student.

Administrator

The **Administrator** entity holds data such as administrator ID, name and email address. The administrator ID is defined as the primary key, uniquely identifying each administrator. The administrator name is a composite attribute, consisting of first and last name. Administrator views zero or many FAQs, and analyzes zero or many ticket responses.

3.2.2 Proposed Transactional Requirement

Data Entry

1. Enter Student Info: Add the details of students, such as ID, name, and email.
2. Enter Staff Info: Add the details of staff, such as ID, name, email, and department.
3. Enter Administrator Info: Add the details of administrators, such as ID, name, and email.
4. Enter Ticket Issue: Add the details of tickets, such as ID, student, staff, subject, description, urgency, and creation time.
5. Enter Ticket Response: Add the details of ticket responses, such as ID, ticket, staff, text, and response time.
6. Enter FAQs: Add the details of FAQs, such as ID, type, and question.
7. Enter FAQ Answer: Add the details of FAQ answers, such as ID, FAQ, and answer.
8. Enter Ticket Status: Add the details of ticket statuses, such as status type (e.g., "Open," "In Progress," "Solved").

Data Update/Delete

1. Update/Delete Student Info: Modify or delete the details of students.
2. Update/Delete Staff Info: Modify or delete the details of staff.
3. Update/Delete Administrator Info: Modify or delete the details of administrators.
4. Update/Delete Ticket Issue: Modify or delete the details of tickets.
5. Update/Delete Ticket Response: Modify or delete the details of ticket responses.
6. Update/Delete FAQs: Modify or delete the details of FAQs.
7. Update/Delete FAQ Answer: Modify or delete the details of FAQ answers.
8. Update/Delete Ticket Status: Modify or delete the details of ticket statuses..

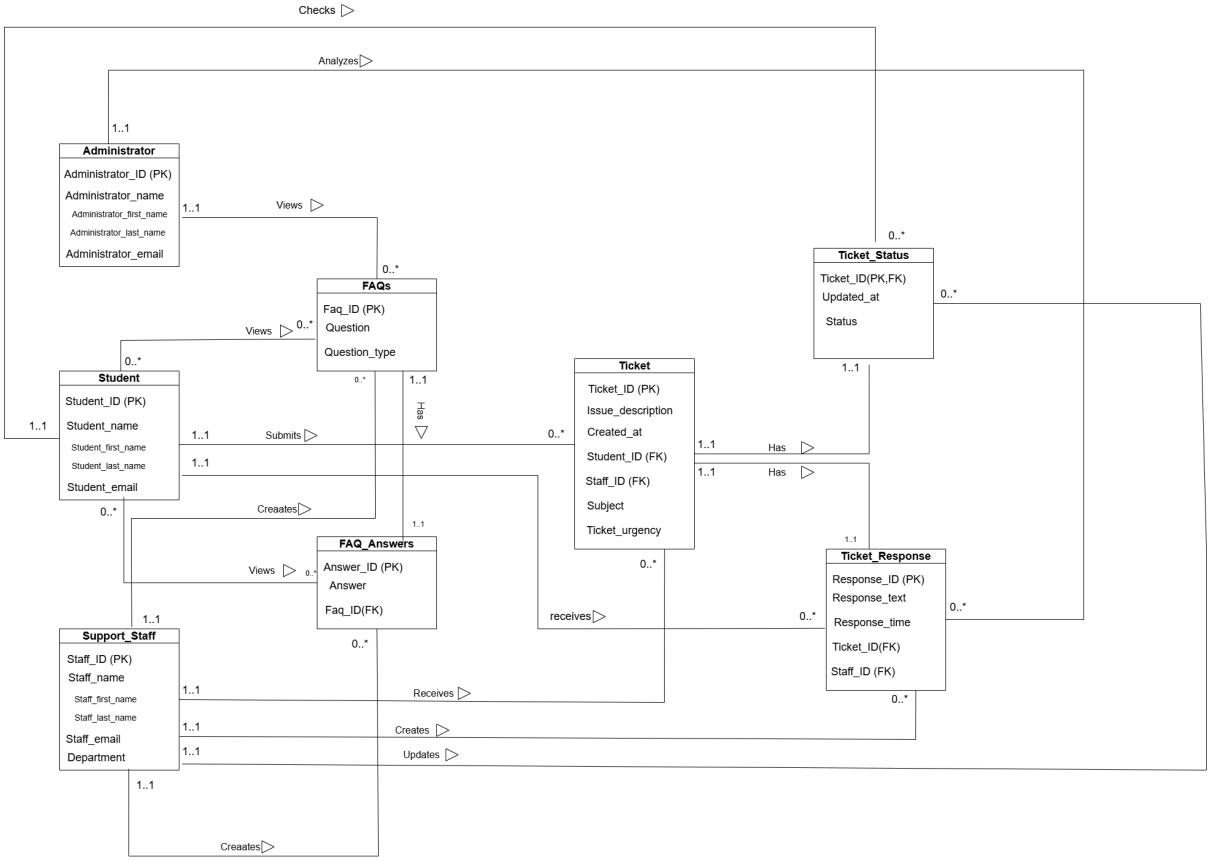
Data Queries

1. Display Student Info: Retrieve and display the details of students.
2. Display Staff Info: Retrieve and display the details of staff.
3. Display Administrator Info: Retrieve and display the details of administrators.
4. Display Ticket Issues: Retrieve and display the details of tickets, including their status.
5. Display Ticket Responses: Retrieve and display the details of ticket responses.
6. Display FAQs: Retrieve and display the details of FAQs.
7. Display FAQ Answers: Retrieve and display the details of FAQ answers.

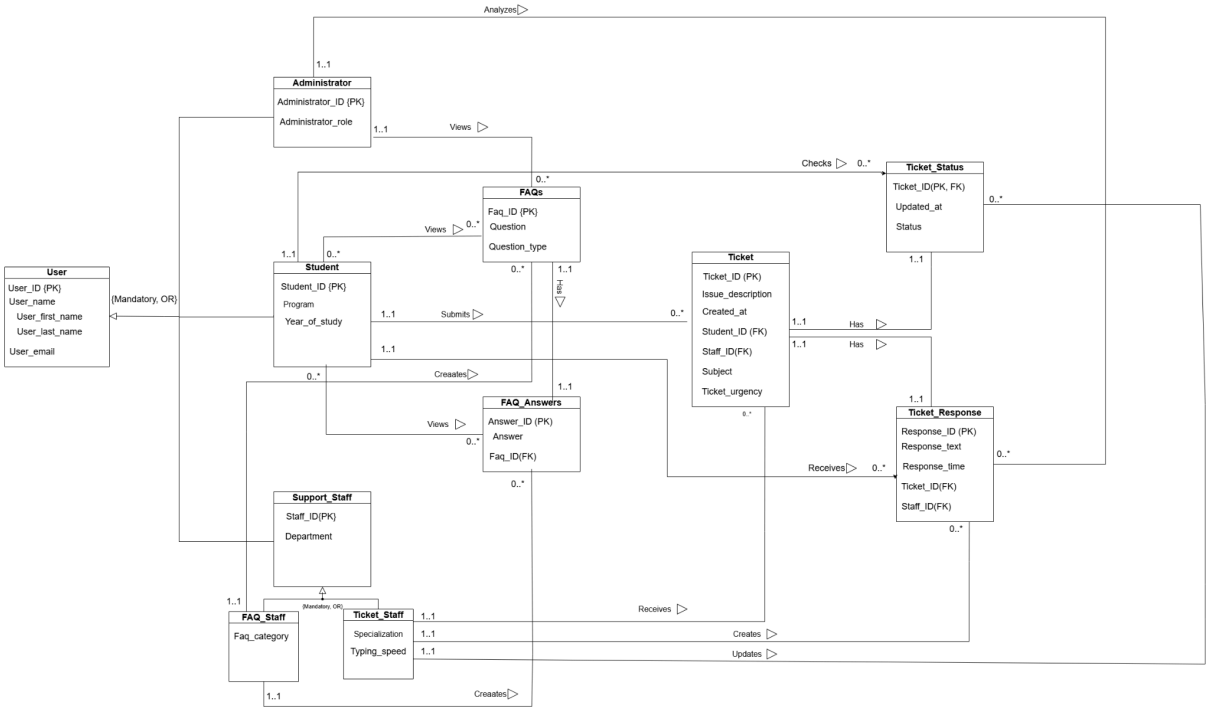
8. Display Ticket Status: Retrieve and display the current status of tickets, along with the update times.
9. Display Unresolved Tickets: Retrieve and display all unresolved tickets based on their status.
10. Display Tickets by Urgency: Retrieve and display tickets categorized by urgency level.
11. Display Tickets by Assigned Staff: Retrieve and display tickets based on the assigned staff.

4.0 Database conceptual design

4.1 Conceptual ERD



4.2 Enhanced ERD (EERD)



5.0 Data dictionary

5.1 Description of Entity

Entity	Description	Explanation
Student	Stores information about students who use the e-learning platform.	Students can read FAQs to find solutions to common problems or create a support ticket to ask questions directly to support staff.
Support_Staff	Stores information about support staff who manage and answer student inquiries.	Support staff can create FAQs and FAQ answers and respond to student tickets.
FAQs	Holds frequently asked questions (FAQs) that are accessible to students and support staff.	FAQs serve as a reference for answering questions and provide solutions students can read before contacting support.
FAQ_Answers	Stores answers to frequently asked questions.	FAQ answers can be updated or deleted by support staff and viewed by students as a self-service resource.
Ticket	Maintains information about tickets created by students when they submit a question or request assistance.	Each ticket uniquely identifies a student's question for support staff, details like the time it's created at.
Ticket_Status	Stores the current status of a ticket (e.g., under review, answered).	The ticket status is visible to both students to track and support staff to update it for students.
Ticket_ Response	Holds the details of support staff responses to tickets.	Ticket responses provide answers to student inquiries, which students can view upon completion.
Administrator	Stores information about the system administrators.	Administrator oversees the FAQs to analyze the areas frequently asked about to make improvements, and ticket

		responses to produce reports about the staff and time response.

5.2 Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
Student	1..1	Submits	0..*	Ticket
	1..1	Receives	0..*	Ticket_Response
	0..*	Views	0..*	FAQs
	0..*	Views	0..*	FAQ_Answers
	1..1	Checks	0..*	Ticket_Status
Support_Staff	1..1	Receives	0..*	Ticket
	1..1	Updates	0..*	Ticket_Status
	1..1	Creates	0..*	Ticket_Response
	1..1	Creates	0..*	FAQs
	1..1	Creates	0..*	FAQ_Answers
Ticket	1..1	Has	1..1	Ticket_Status
	1..1	Has	1..1	Ticket_Response
FAQs	1..1	Has	1..1	FAQ_Answers
Administrator	1..1	Views	0..*	FAQs
	1..1	Analyzes	0..*	Ticket_Response

5.3 Description of Attributes

Entity	Attributes	Description	DataType	Null	PK	FK
Student	student_ID	Uniquely identifies each student	VARCHAR(10)	No	Yes	-
	student_name	Full name of the student	VARCHAR(60)	No	No	-
	student_first_name	First name of the student	VARCHAR(30)	No	No	-
	student_last_name	Last name of the student	VARCHAR(30)	No	No	-
	student_email	Email address of the student	VARCHAR(50)	No	No	-

Entity	Attributes	Description	DataType	Null	PK	FK
Support_Staff	staff_ID	Uniquely identifies each support staff member	VARCHAR(10)	No	Yes	-

	staff_name	Full name of the support staff	VARCHAR(60)	No	No	-
	staff_first_name	First name of the staff	VARCHAR(30)	No	No	-
	staff_last_name	Last name of the staff	VARCHAR(30)	No	No	-
	staff_email	Email address of the support staff	VARCHAR(50)	No	No	-
	department	Department where the staff works	VARCHAR(30)	Yes	No	-

Entity	Attributes	Description	DataType	Null	PK	FK
FAQs	faq_ID	Unique identifier for each FAQ entry	VARCHAR(10)	No	Yes	-
	question_type	Category or type of the question	VARCHAR(20)	No	No	-
	question	The FAQ question	TEXT	No	No	-

Entity	Attributes	Description	DataType	Null	PK	FK
FAQ_Answers	answer_ID	Unique identifier for each answer	VARCHAR(10)	No	Yes	-

	faq_ID	Foreign key linking to FAQs	VARCHAR(10)	No	No	Yes
	answer	Text of the answer	TEXT	No	No	-

Entity	Attributes	Description	DataType	Null	PK	FK
Ticket	ticket_ID	Unique identifier for each ticket	VARCHAR(10)	No	Yes	-
	student_ID	Foreign key linking to Student	VARCHAR(10)	No	No	Yes
	staff_ID	Foreign key linking to Support_Staff	VARCHAR(10)	Yes	No	Yes
	subject	Subject of the issue	VARCHAR(50)	No	No	-
	issue_description	Detailed description of the issue	TEXT	No	No	-

	created_at	Date and time ticket was created	TIMESTAMP	No	No	-
	ticket_urgency	Level of urgency (e.g., "Low," "Medium," "High")	ENUM('Low', 'Medium', 'High')	No	No	-

Entity	Attributes	Description	DataType	Null	PK	FK
Ticket Status	ticket_ID	Foreign key linking to Ticket	VARCHAR(10)	No	Yes	Yes
	status	Current status of the ticket (e.g., "Open," "In Progress," "Completed")	ENUM('Open', 'In Progress', Completed)	No	No	-
	updated_at	Date and time of status update	TIMESTAMP	No	No	-

Entity	Attributes	Description	DataType	Null	PK	FK
Ticket_ Response	response_ID	Unique identifier for each response	VARCHAR(10)	No	Yes	-
	ticket_ID	Foreign key linking to Ticket	VARCHAR(10)	No	No	Yes
	staff_ID	Foreign key linking to Support_Staff	VARCHAR(10)	No	No	Yes

	response_text	Detailed response or follow-up text	TEXT	No	No	-
	response_time	Date and time of the response	TIMESTAMP	No	No	-

Entity	Attributes	Description	DataType	Null	PK	FK
Administrator	administrator_ID	Unique identifier for each administrator	VARCHAR(10)	No	Yes	-
	administrator_name	Full name of the administrator	VARCHAR(60)	No	No	-
	administrator_first_name	First name of the administrator	VARCHAR(30)	No	No	-
	administrator_last_name	Last name of the administrator	VARCHAR(30)	No	No	-
	administrator_email	Email address of the administrator	VARCHAR(50)	No	No	-

6.0 Summary

In this phase, our team has developed a comprehensive understanding of the structure, functionality, interactions and data stored for the proposed ticketing system within UTM's e-learning platform by completing the conceptual database design.

By creating data flow diagrams, we have gained a deeper understanding of visualizing the system’s processes that will interact with the databases and entities, and the data flow in general.

By stating the data and transactions requirements we have understood the operational logic of the system, and how it handles operations and interactions while maintaining consistency and accuracy of data stored. The data dictionary also highlighted the entities, its attributes and relationships and served as a reference for the conceptual design.

Through designing the conceptual ERD by referring to the data dictionary, we have modelled our system’s relations visually and their data structure and relationships. While designing the enhanced ERD, we have understood how to present the complexity of our system using additional data modeling concepts like specialization, generalization, which have added depth and flexibility to our system’s design.