FACULTY OF COMPUTING
UTM Johor Bahru

Phase 3 (P3) – Database Conceptual Design (ERD)

<NEXSCHOLAR EVENT/TICKET MANAGEMENT SYSTEM>

SECD2523-03 DATABASE

Lecturer: Dr. Izyan Izzati Binti Kamsani

Group Name: NetNerd

Team Members:

1. LUVINESH SUDESH (A21EC0198)

2.MOHAMAD SYAKIRIN BIN MOHAMAD YUSOF (A22EC0195)

3.MUHAMMAD ABDU BIN ABDUL BA'ARI (A22EC0199)

4.MUHAMMAD AFFIF FARHAN BIN ZAMZURI (A22EC0200)

5. TENGKU MUHAMMAD AIMAN ALIFF BIN TENGKU AZEEZEE (A22EC0283)

TABLE OF CONTENTS

1. INTRODUCTION

Due to the rising use of social media, we have actually come up with an idea for the usage of postgraduate students to do research and connect with real people who share the same interest together. This helps them to actually connect with one another and do research. Not only that, Nexscholar can be used to also book events and purchase tickets. This could make postgraduates' life easier where they don't have to break their heads thinking about where they should go to make their research worthwhile. This app would give them notifications if there's an upcoming event where they can buy the tickets straight away from the app itself. Sounds convenient isn't it?

Encik Najmi can only post it and acts as middle-men. This event can be booked online and subscribed to tickets. Want to know how to attend the event.The event details would be sent to whatsapp. Anyone would be able to join the event. This would be a good opportunity for undergraduate students just in case if they want to get more informations.The tickets can be purchased online from any account.

We want to improve Nexsholar performance in terms of online banking, order stocking, and analysis of order data. By this Nexscholar would be better and many other postgraduate students will enjoy using this application.

2. OVERVIEW OF PROJECT

In this phase, we need to refine the conceptual Entity-Relationship Diagram (ERD) from Phase 2 into a logical ERD by removing non-relational features and complex relationships. To maintain data integrity and reduce redundancy, this involves creating relational schema tables derived from the logical ERD and executing Boyce-Codd Normal Form (BCNF) normalization. A structured representation of the system data model will be the ultimate logical ERD. The phase also involves producing a comprehensive report that includes the logical ERD, standardized relational database schemas, data dictionary, and mapped interface design, which includes SQL statements and individual demonstrations offered to show how they interact with the database design. Additionally, the phase involves updating the data dictionary based on standardized relationships and validating the logical ERD against system transaction requirements using interface design.

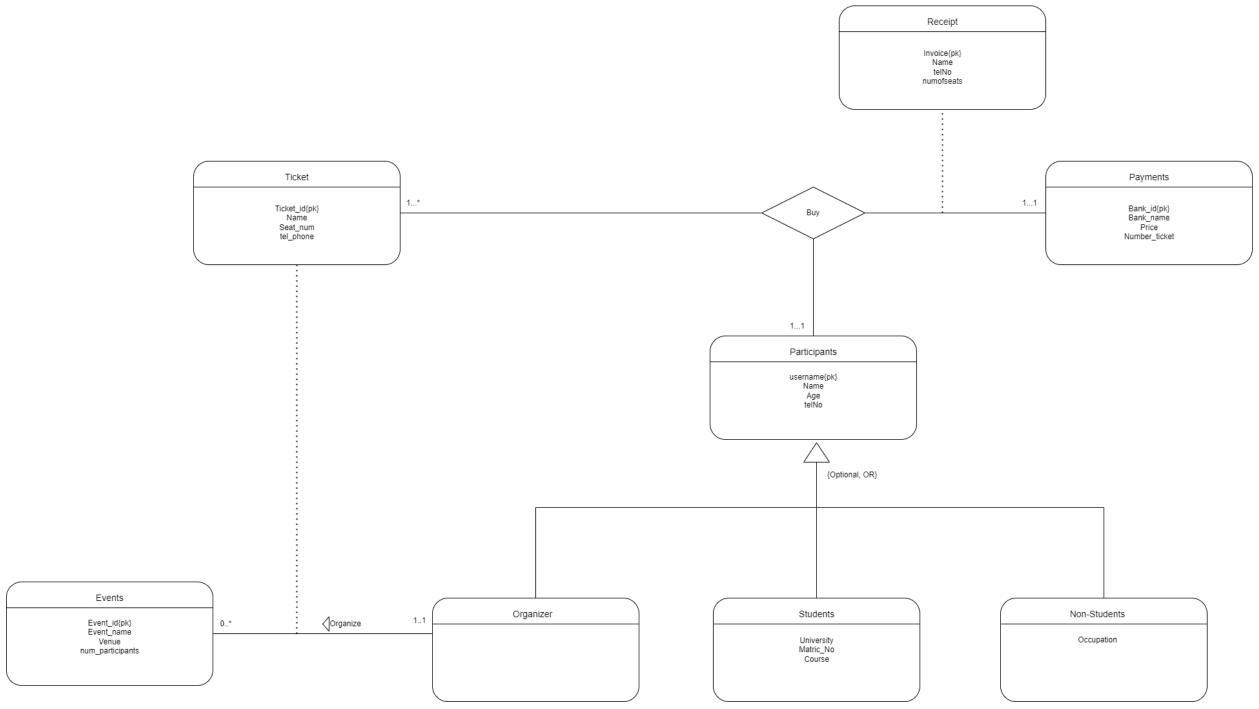
3. DATABASE CONCEPTUAL DESIGN

3.1 Updated Business Rule

1. Organizer can organize many events and produce tickets
2. Participants can buy at least one tickets in one payments and produce receipts

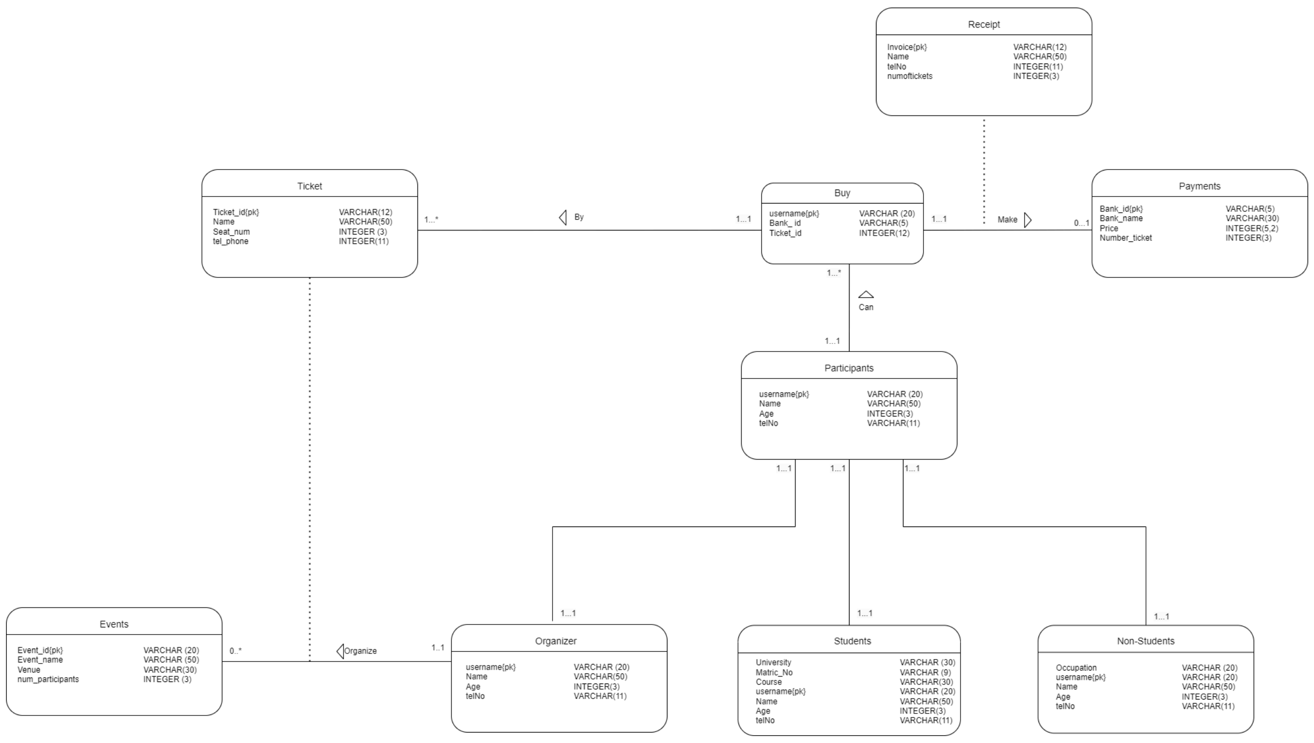
3.2 Conceptual ERD

Below is our updated conceptual erd based on our updated business rule



4. DB LOGICAL DESIGN

4.1 Logical ERD



4.2 Updated Data Dictionary

Participant table

Field	Data Type	Nullability	Description
Participant_id	VARCHAR2(20)	NO	Unique ID for Participant
fName	VARCHAR2(10)	NO	First name of participant
lName	VARCHAR2(15)	NO	Last name of participant
phone_number	VARCHAR2(15)	NO	Participant phone number
email_address	VARCHAR2(50)	NO	Participant email address

Payment table

Field	Data Type	Nullability	Description
payment_id	VARCHAR2(20)	NO	Unique ID for payment
user_id	VARCHAR2(20)	NO	Foreign key to User table
postcode	NUMBER(10)	NO	Address postcode

Student table

Field	Data Type	Nullability	Description
Username	VARCHAR2(20)	NO	Unique name for user
fName	VARCHAR2(10)	NO	First name of user
lName	VARCHAR2(15)	NO	Last name of user
phone_number	VARCHAR2(15)	NO	User phone number
email_address	VARCHAR2(50)	NO	Email address of user
matric_no	VARCHAR2(10)	NO	Unique string for students

Non-Student table

Field	Data Type	Nullability	Description
Username	VARCHAR2(20)	NO	Unique name for user
fName	VARCHAR2(10)	NO	First name of user
lName	VARCHAR2(15)	NO	Last name of user
phone_number	VARCHAR2(15)	NO	User phone number
email_address	VARCHAR2(50)	NO	Email address of user

Organizer table

Field	Data Type	Nullability	Description
Username	VARCHAR2(20)	NO	Unique name for user
fName	VARCHAR2(10)	NO	First name of organizer
lName	VARCHAR2(15)	NO	Last name of organizer
phone_number	VARCHAR2(15)	NO	Organizer phone number
email_address	VARCHAR2(50)	NO	Organizer email address

Event table

Field	Data Type	Nullability	Description
event_id	VARCHAR2(20)	NO	Unique ID for event
maxParticipant	NUMBER(3)	NO	Maximum participants for event
cost	NUMBER(40)	NO	Cost of the event

Venue	VARCHAR2(50)	NO	Venue for the event
user_id	VARCHAR2(20)	NO	Foreign key to Organizer table

Ticket table

Field	Data Type	Nullability	Description
Ticket_id	VARCHAR2(20)	NO	Unique ID of the ticket
name	NUMBER(3)	NO	Name of the customer
Seat_num	NUMBER(40)	NO	Seat number for the customer
Tel_num	VARCHAR2(15)	NO	Telephone number of the customer

Receipt table

Field	Data Type	Nullability	Description
Receipt_id	VARCHAR2(20)	NO	Unique ID of the Receipt

4.3 Normalization

Participant Table (username, fName, lName, phone_number, email_address)
fd1: Participant_id → fName, lName, phone_number, email_address
1NF, 2NF, 3NF, & BCNF:
Participant(<u>username</u>, fName, lName, phone_number, email_address)

Payment Table (payment_id, user_id, postcode)
fd1: payment_id → user_id, postcode
1NF, 2NF, 3NF, & BCNF:
Payment(payment_id, user_id, postcode)

Student Table (username, fName, lName, phone_number, email_address, matric_no)
fd1: Student_user_id → fName, lName, phone_number, email_address, matric_no
1NF, 2NF, 3NF, & BCNF:
Student(username, fName, lName, phone_number, email_address, matric_no)

Non-Student Table (username, fName, lName, phone_number, email_address)
fd1: Participant_IC_Number → fName, lName, phone_number, email_address
1NF, 2NF, 3NF, & BCNF:
NonStudent(username, fName, lName, phone_number, email_address)

Organizer Table (username,event_id, fName, lName, phone_number, email_address)
fd1: event_id → fName, lName, phone_number, email_address
1NF, 2NF, 3NF, & BCNF:
Organizer(username,event_id, fName, lName, phone_number, email_address)

Event Table (event_id, maxParticipant, cost, Venue, user_id)
fd1: event_id → maxParticipant, cost, Venue, user_id
1NF, 2NF, 3NF, & BCNF:
Event(event_id, maxParticipant, cost, Venue, user_id)

Ticket Table (Ticket_id, name, Seat_num, Tel_num)
fd1: Ticket_id → name, Seat_num, Tel_num
1NF, 2NF, 3NF, & BCNF:
Ticket(Ticket_id, name, Seat_num, Tel_num)

Receipt Table (Receipt_id,fName, lName, phone_number, num_of_seats)
fd1: Receipt_id → fName, lName, phone_number, num_of_seats
1NF, 2NF, 3NF, & BCNF:
Receipt(Receipt_id,fName, lName, phone_number, num_of_seats)

5. RELATIONAL DB SCHEMAS

Participant

participant_id	fName	lName	phone_number	email_address
P001	John	Doe	123-456-7890	john.doe@example.com
P002	Jane	Smith	987-654-3210	jane.smith@example.com
P003	Mike	Johnson	555-123-4567	mike.john@example.com

Payment

payment_id	user_id	postcode
PAY001	P001	12345
PAY002	P002	54321
PAY003	P003	67890

Student

username	fName	lName	phone_number	email_address	matric_no
S001	Emily	Jones	111-222-3333	emily.jones@example.com	M12345
S002	Alex	Brown	444-555-6666	alex.brown@example.com	M67890
S003	Chris	Miller	777-888-9999	chris.miller@example.com	M54321

Non-Student

username	fName	lName	phone_number	email_address
NS001	Sophia	Wilson	111-222-3333	sophia.wilson@example.com
NS002	Daniel	Clark	444-555-6666	daniel.clark@example.com

NS003	Olivia	Davis	777-888-9999	olivia.davis@example.com

Organizer

username	fName	lName	phone_number	email_address
org_user1	Mark	Taylor	111-222-3333	mark.taylor@example.com
org_user2	Emma	Hill	444-555-6666	emma.hill@example.com
org_user3	Ryan	Lee	777-888-9999	ryan.lee@example.com

Event

event_id	maxParticipant	cost	venue	user_id
E001	100	50	Conference Hall A	org_user1
E002	150	75	Ballroom B	org_user2
E003	8	40	Meeting Room C	org_user3

Ticket

ticket_id	fName	lName	seat_num	tel_num
T001	John	Doe	15	123-456-7890
T002	Jane	Smith	50	987-654-3210
T003	Mike	Johnson	100	555-123-4567

Receipt

receipt_id	fName	lName	phone_number	num_of_seats
R001	John	Doe	123-456-7890	2
R002	Jane	Smith	987-654-3210	1

R003	Mike	Johnson	555-123-4567	3

6. SQL STATEMENTS

DDL

```
CREATE TABLE Participant (
    Participant_id VARCHAR2(20) PRIMARY KEY NOT NULL,
    fName VARCHAR2(10) NOT NULL,
    lName VARCHAR2(15) NOT NULL,
    phone_number VARCHAR2(15) NOT NULL,
    email_address VARCHAR2(50) NOT NULL
);

CREATE TABLE Payment (
    Payment_id VARCHAR2(20) PRIMARY KEY NOT NULL,
    user_id VARCHAR2(20) CONSTRAINT participant_fk REFERENCES Participant
(Participant_id) NOT NULL,
    postcode NUMBER(10) NOT NULL
);

CREATE TABLE Student (
    Participant_id VARCHAR2(20) PRIMARY KEY NOT NULL,
    fName VARCHAR2(10) NOT NULL,
    lName VARCHAR2(15) NOT NULL,
    phone_number VARCHAR2(15) NOT NULL,
    email_address VARCHAR2(50) NOT NULL,
    matric_no VARCHAR2(10) UNIQUE NOT NULL
);

CREATE TABLE NonStudent (
    Participant_id VARCHAR2(20) PRIMARY KEY NOT NULL,
    fName VARCHAR2(10) NOT NULL,
    lName VARCHAR2(15) NOT NULL,
    phone_number VARCHAR2(15) NOT NULL,
   email_address VARCHAR2(50) NOT NULL
);
```

```
CREATE TABLE Organizer (
   Username VARCHAR2(20) PRIMARY KEY NOT NULL,
   fName VARCHAR2(10) NOT NULL,
   lName VARCHAR2(15) NOT NULL,
   phone_number VARCHAR2(15) NOT NULL,
   email_address VARCHAR2(50) NOT NULL
);

CREATE TABLE Event (
   Event_id VARCHAR2(20) PRIMARY KEY NOT NULL,
   maxParticipant NUMBER(3) NOT NULL,
   cost NUMBER(38) NOT NULL,
   Venue VARCHAR2(50) NOT NULL,
   user_id VARCHAR2(20) CONSTRAINT organizer_pk REFERENCES
Organizer(Username)
);

CREATE TABLE Ticket (
   Ticket_id VARCHAR2(20) PRIMARY KEY NOT NULL,
   fName VARCHAR2(10) NOT NULL,
   lName VARCHAR2(15) NOT NULL,
   Seat_num NUMBER(38) NOT NULL,
   Tel_num VARCHAR2(15) NOT NULL
);

CREATE TABLE Receipt (
   Receipt_id VARCHAR2(20) PRIMARY KEY NOT NULL,
   fName VARCHAR2(10) NOT NULL,
   lName VARCHAR2(15) NOT NULL,
   phone_number VARCHAR2(15) NOT NULL,
   num_of_seats NUMBER(3) NOT NULL
);
```

DML

```
INSERT INTO Participant
VALUES ('P001', 'John', 'Doe', '123-456-7890', 'john.doe@example.com');

INSERT INTO Participant
VALUES ('P002', 'Jane', 'Smith', '987-654-3210', 'jane.smith@example.com');

INSERT INTO Participant
VALUES ('P003', 'Mike', 'Johnson', '555-123-4567', 'mike.johnson@example.com’);
```

```
INSERT INTO Payment
VALUES ('PAY001', 'P001', 12345);

INSERT INTO Payment
VALUES ('PAY002', 'P002', 54321);

INSERT INTO Payment
VALUES ('PAY003', 'P003', 67890);

INSERT INTO Student
VALUES ('S001', 'Emily', 'Jones', '111-222-3333', 'emily.jones@example.com', 'M12345');

INSERT INTO Student
VALUES ('S002', 'Alex', 'Brown', '444-555-6666', 'alex.brown@example.com', 'M67890');

INSERT INTO Student
VALUES  ('S003', 'Chris', 'Miller', '777-888-9999', 'chris.miller@example.com', 'M54321');

INSERT INTO NonStudent
VALUES ('NS001', 'Sophia', 'Wilson', '111-222-3333', 'sophia.wilson@example.com');

INSERT INTO NonStudent
VALUES ('NS002', 'Daniel', 'Clark', '444-555-6666', 'daniel.clark@example.com');

INSERT INTO NonStudent
VALUES ('NS003', 'Olivia', 'Davis', '777-888-9999', 'olivia.davis@example.com');

INSERT INTO Organizer
VALUES ('org_user1', 'Mark', 'Taylor', '111-222-3333', 'mark.taylor@example.com');

INSERT INTO Organizer
VALUES ('org_user2', 'Emma', 'Hill', '444-555-6666', 'emma.hill@example.com');

INSERT INTO Organizer
VALUES ('org_user3', 'Ryan', 'Lee', '777-888-9999', 'ryan.lee@example.com')

INSERT INTO Event
VALUES ('E001', 100, 50.00, 'Conference Hall A', 'org_user1');

INSERT INTO Event
VALUES  ('E002', 150, 75.00, 'Ballroom B', 'org_user2');
INSERT INTO Event
VALUES ('E003', 80, 40.00, 'Meeting Room C', 'org_user3');
```

```
INSERT INTO Ticket
VALUES ('T003','Mike','Johnson','100','555-123-4567');

INSERT INTO Ticket
VALUES ('T002','Jane','Smith','50','1987-654-3210');

INSERT INTO Ticket
VALUES ('T001','John','Doe','15','123-456-7890');

INSERT INTO Receipt
VALUES ('R001', 'John', 'Doe', '123-456-7890', 2);

INSERT INTO Receipt
VALUES ('R002', 'Jane', 'Smith', '987-654-3210', 1);

INSERT INTO Receipt
VALUES ('R003', 'Mike', 'Johnson', '555-123-4567', 3);
```

7. SUMMARY

To conclude, the project aims to create NexScholar, a platform for postgraduate students to connect, organize, and attend events seamlessly. The detailed design encompasses data flow diagrams, conceptual and logical ERDs, relational database schemas, and SQL statements. The data dictionary serves as a guide for understanding the database structure, facilitating the development process.