UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING

SEMESTER 1 / 20232024

SECD2523 - 03 DATABASE

GROUP PROJECT

PHASE 2

DATABASE CONCEPTUAL DESIGN (ERD)

LECTURER NAME: DR IZYAN IZZATI BINTI KAMSANI

GROUP MEMBERS

NO	NAME	MATRIC NUMBER
1	OSAMA HAMOOD MAHYOOB AHMED	A22EC4025
2	HAMDAN SALEH OMAR AL-MOHAMADI	A22EC4027
3	AMMAR YASSER SALAH	A22EC3001
4	AHMED MOHAMMED MAZEN ALYASIN	A21EC4030
5	SARANYA A/P JAYARAMA REDDY	B22EC3013

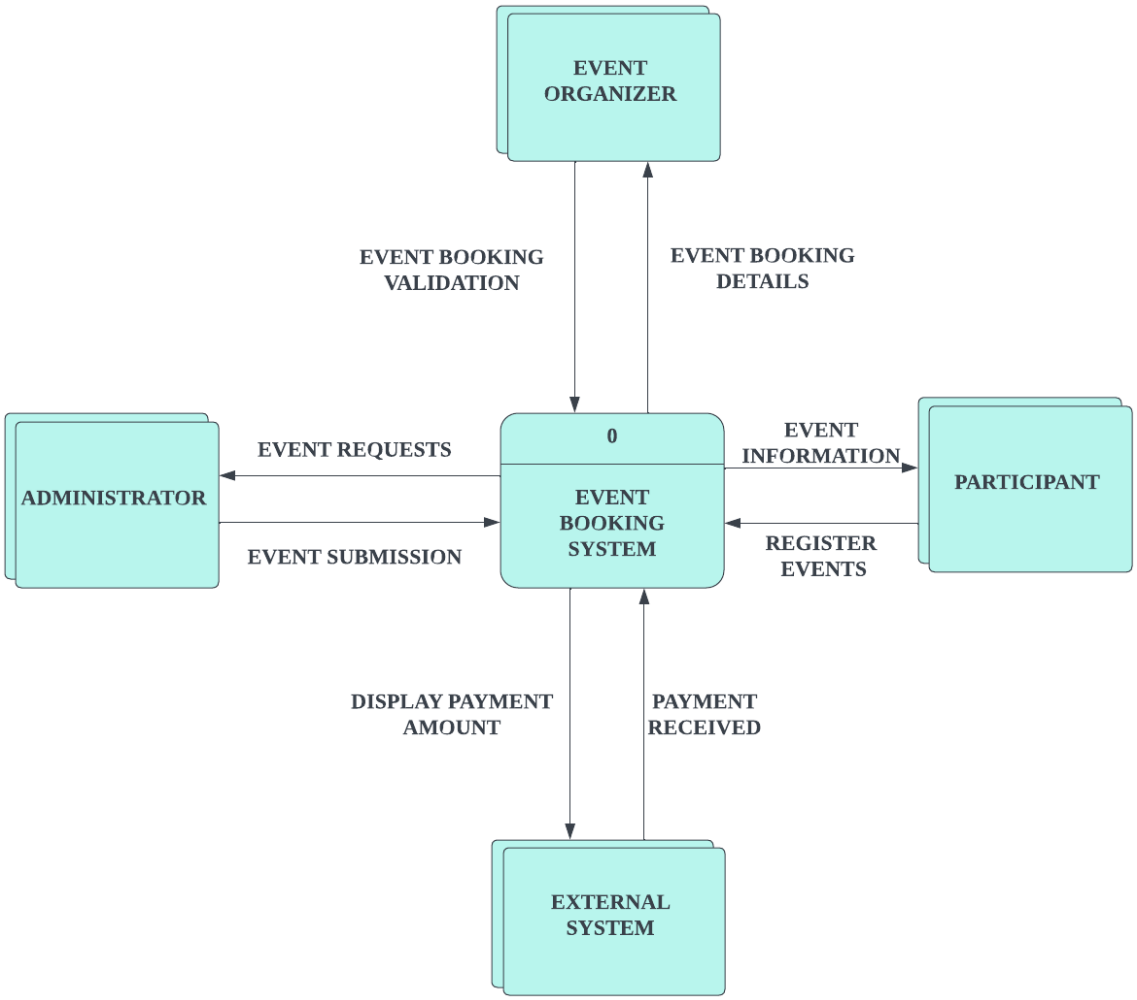
TABLE OF CONTENTS

1.0 INTRODUCTION

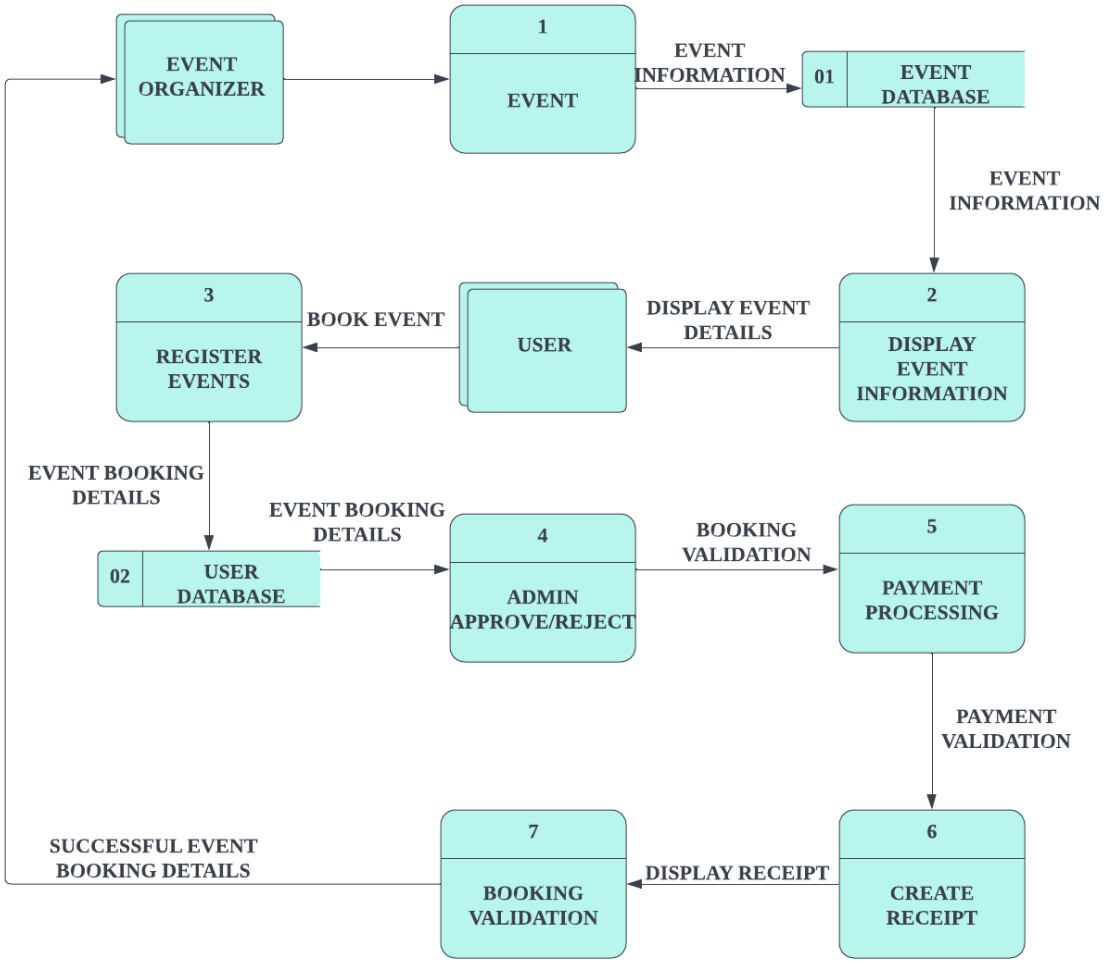
In this project, the owner is the user of the current system in the NexScholar website which is an event booking system which is designed to provide services such as booking events. Throughout the business process, the user encountered challenges while using the event booking system to register the event. Therefore, we are assigned to develop a database system that enhances the flow of event bookings. Additionally, we aim to create a user-friendly experience, enabling users to easily access event information, sign up for events, and allow organizers to submit and manage their events on the platform. Following the completion of phase 1, where we gathered data requirements from the owner and users, we are now advancing to phase 2, which involves designing our proposed system. This paper will encompass the creation of a data flow diagram to visually represent our proposed system. Furthermore, we will delve into discussions on current business data and transaction requirements, the updated conceptual design of the database, an enhanced ERD and the data dictionary. These processes will provide us with an advantage as we move forward into phase 3 of this project.

2.0 DFD

2.1 CONTEXT DIAGRAM



2.2 DIAGRAM 0


EVENT ORGANIZER
1
EVENT
EVENT INFORMATION
01
EVENT DATABASE
EVENT INFORMATION
2
DISPLAY EVENT INFORMATION
DISPLAY EVENT DETAILS
USER
BOOK EVENT
3
REGISTER EVENTS
EVENT BOOKING DETAILS
02
USER DATABASE
EVENT BOOKING DETAILS
4
ADMIN APPROVE/REJECT
BOOKING VALIDATION
5
PAYMENT PROCESSING
PAYMENT VALIDATION
6
CREATE RECEIPT
DISPLAY RECEIPT
7
BOOKING VALIDATION
SUCCESSFUL EVENT BOOKING DETAILS

3.0 DATA & TRANSACTION REQUIREMENT

3.1 PROPOSED BUSINESS RULE

3.1.1 **Entities**:

1. **Organizer**: Represents Event organizers on the NexScholar platform.
2. **Event**: Represents the events organized on the NexScholar platform.
3. **Participant**: Represents users who register for and participate in events.
4. **Payment**: Represents the payments made by participants for events that require payment.
5. **Admin**: Represents the administrative role responsible for creating events on NexScholar based on event info given by the organizer.

3.1.2 **Relationships**:

1. The **Organizer entity** and the **Event entity** have a **one-to-many** relationship, since one user can **organize several events**, but each event is **organized by a single user.**
2. There is a **many-to-many** relationship between the **Event entity** and the **Participant entity**, as an event can **have** multiple participants, and a participant can **register for multiple events**.
3. There is a **one-to-many** relationship between the **Participant entity** and the **Payment entity**, indicating that a participant can **make** multiple payments, but each payment is **made by exactly one participant.**
4. There is a **one-to-many** relationship between the **Admin entity** and the **Event entity**, due to the fact that an admin can **create** multiple events, but each event is **created by exactly one admin.**

3.1.3 **Attributes**:

1. The Organizer entity attributes: **OrgID, Orgname, Email.**
2. The Event entity attributes: **EventID, EventName, Date, Location.**
3. The Participant entity attributes: **ParticipantID, ParticipantName, Email.**
4. The Payment entity attributes: **PaymentID, Amount, PaymentDate.**
5. The Admin entity attributes: **AdminID, AdminName, Email**.

3.1.4 **Cardinality**:

1. **Organizer**: One-to-Many [1..*].
2. **Event**: Many-to-Many [*..*].
3. **Participant**: One-to-Many [1..*].
4. **Admin**: One-to-Many [1..*].

3.2 PROPOSED DATA & TRANSACTIONAL

Event Registration Process:

1. User Registration:

- Description: Users must register on the NexScholar platform to participate in events.

- Transaction Steps:

    - User provides required information (ParticipantName, Email) to register.

    - System validates the information and generates a unique ParticipantID.

    - User registration details are stored in the Participant entity.

2. Event Creation Process:

- Description: Organizers create events on the NexScholar platform.

- Transaction Steps:

    - Organizer provides event details (EventName, Date, Location) during event creation.

    - System validates the information and generates a unique EventID.

    - Event details are stored in the Event entity and associated with the Organizer.

3. Event Registration by Participant:

- Description: Participants register for events on the NexScholar platform.

- Transaction Steps:

    - Participant selects an event and submits the registration form.

    - System validates the information and generates a unique RegistrationID.

    - Participant details and event association are stored in the Participant-Event relationship.

4. Payment Processing:

- Description: Participants make payments for events that require payment.

- Transaction Steps:

    - Participant selects the event with a payment requirement.

    - Participant provides payment details (Amount, PaymentDate).

    - System validates the payment information and generates a unique PaymentID.

    - Payment details are stored in the Payment entity and associated with the Participant.

5. Event Approval by Admin:

- Description: Admins review and approve/disapprove event submissions.

- Transaction Steps:

    - Admin receives notifications for new event submissions.

    - Admin reviews event details and makes an approval decision.

    - System updates the event status based on the admin's decision.

TRANSACTIONAL CONTROL:

1. Concurrency Control:

- To handle simultaneous event registration and payment processes by participants, and implement concurrency control mechanisms to ensure data consistency.

- Use techniques like locking or timestamp-based protocols to prevent conflicts.

2. Transaction Isolation:

- Ensure transaction isolation levels to manage the visibility of concurrent transactions.

- Choose an appropriate isolation level to balance data consistency and performance.

3. Transaction Durability:

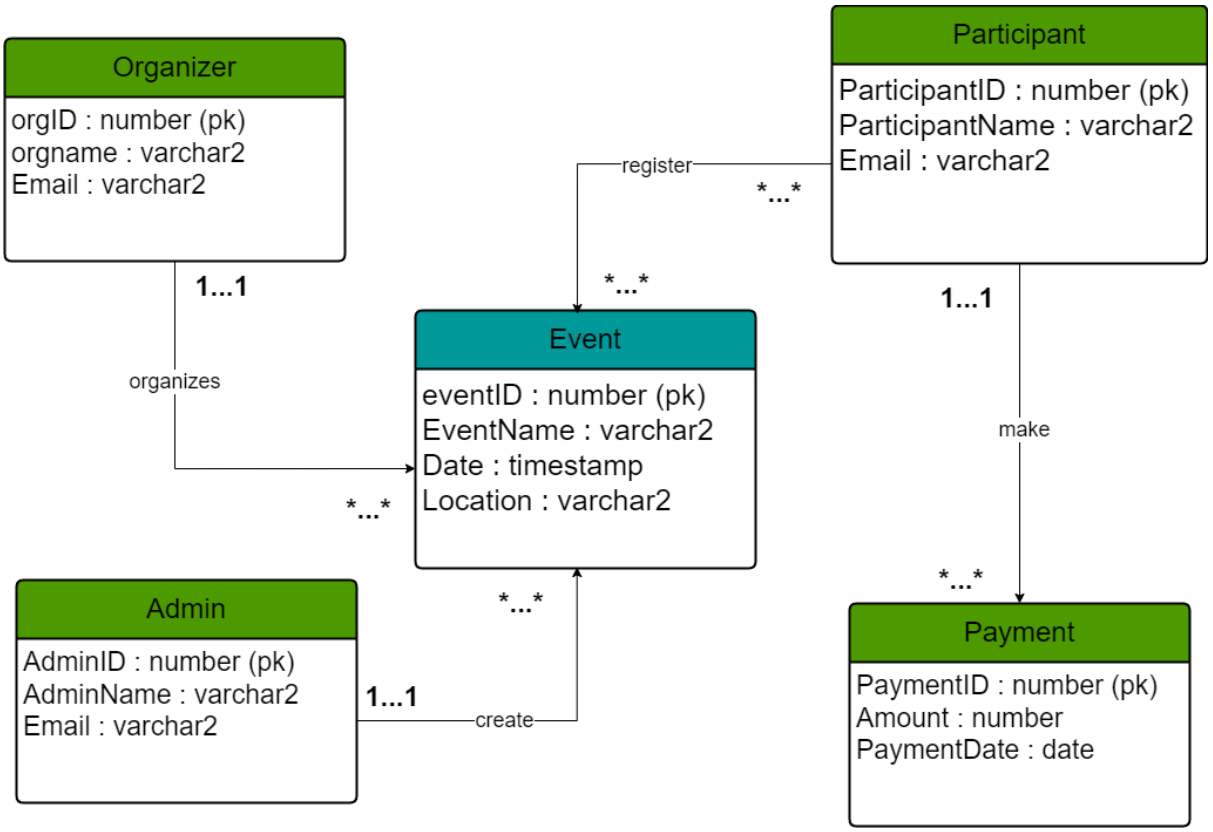
- Implement mechanisms to guarantee the durability of transactions, ensuring that committed transactions persist even in the case of system failures.

4. Transaction Atomicity:

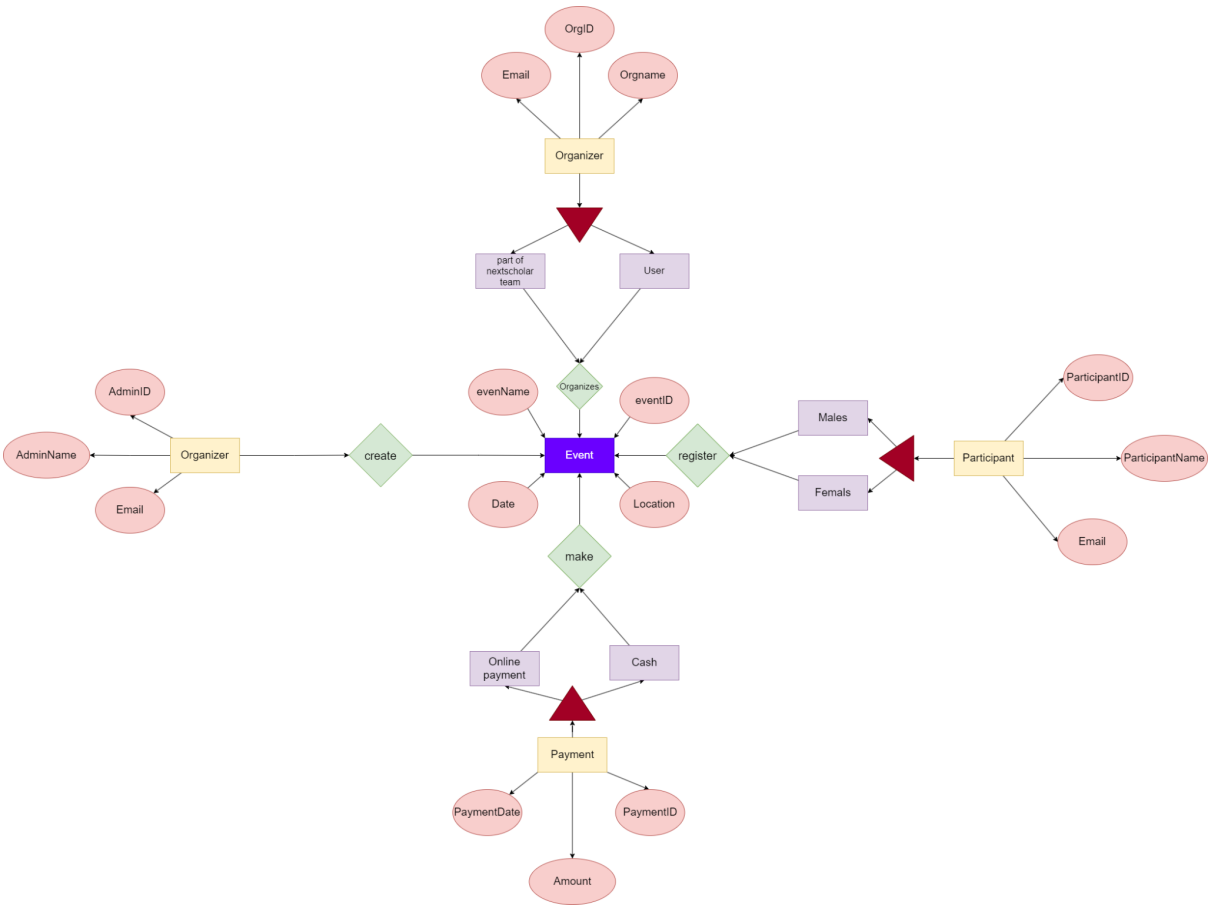
- Maintain atomic transactions to ensure that all steps within a transaction are completed successfully, or none of them are.

4.0 DATABASE CONCEPTUAL DESIGN

4.1 CONCEPTUAL ERD



4.2 ENHANCED ERD (EERD)



5.0 DATA DICTIONARY

Data Dictionary					
Entity Name	**Attributes**	**Description**	**Data Type**	**Nulls**	**Multi-Valued**
Organizer	OrgID	Unique identifier for organizer	Number(5)	No	No
	OrgName	Name of the organizer	Varchar2(20)	No	No
	Email	Email address of the organizer	Varchar2(100)	No	No
Event	EventID	Unique identifier for the event	Number(5)	No	No
	EventName	Name of the event	Varchar2(20)	No	No
	Date	Date and time of the event	Timestamp	No	No
	Location	Location where the event is organized	Varchar2(100)	No	No
Participant	ParticipantID	Unique identifier for the participant	Number(5)	No	No
	ParticipantName • fName • lName	first Name and Last Name of participants.	Varchar2(20) Varchar2(20)	No	No
	Email	Email address of the participant	Varchar2(100)	No	No
Payment	PaymentID	Unique identifier for the payment	Number(5)	No	No
	Amount	Amount of the payment	Number(10)	No	No
	PaymentDate	Date and time of when the payment was made	Timestamp	No	No
Admin	AdminID	Unique identifier for the admin	Number(5)	No	No
	AdminName	Name of the admin	Varchar2(20)	No	No
	Email	Email address of the admin	Varchar2(100)	No	No

6.0 SUMMARY

During Phase 2 of the proposed system (NexScholar) we designed a data flow diagram (DFD) incorporating data and transaction requirements for better visualization on how the system-to-be-implemented will actually work. We have also provided key information for creating our database in the next phase where we identified entities like Organizer, Event, Participant, Payment, and Admin. Transactional processes are detailed, and transactional controls are discussed. Conceptual Entity Relationship Diagrams (ERD) and Enhanced ERD are created to address challenges and advance to Phase 3.