Database Design Document

for

Event Management System

Version 1.1.0

Prepared by

Group Number: 16

Name	***Roll Number***	***Email***
Vimal Rajesh	*B180336CS*	*vimrajesh@gmail.com*
Puchakayla Dheeraj Reddy	*B180902CS*	*dheerajreddy.puchakayala@gmail.com*
P Arjun	*B180454CS*	*arjun_b180454cs@nitc.ac.in*
Alok Raj	*B180411CS*	*alok_b180411cs@nitc.ac.in*
Kunal Ravikumar Jagtap	*B180921CS*	*ravikumar_b180921cs@nitc.ac.in*
Argha Pratim Mandal	*B180331CS*	*argha_b180331cs@nitc.ac.in*

Instructors: **Dr. Abdul Nazeer K A,**

Dr. Prabhu M

Course: **CS3002D Database Management Systems**

Date: **27 November 2020**

Table of Contents

1. Introduction

This document is a Database Design Document(DDD) for the Event Management System application (EMS) for NITC. A database is a collection of data, usually stored in electronic form. A database is typically designed so that it is easy to store and access information. A good database is crucial to any company or organisation. This is because the database stores all the pertinent details about the EMS such as event details, student records, registrant details for an event etc.

It converts logical or the conceptual data to physical storage constructs. It is the organization of data according to the database model. The DDD is organized into several sections to help to assist the development of the database and also explains the functional design decisions taken. This enables both the client side and the developer side to have a clear picture of the database design and work on developing and interfacing with the database.

The EER conceptual model was constructed depending on the SRS and other functional requirements. The Relational Database Schema was constructed from the conceptual model by following the rules for converting ER -> RDB. The relations were further normalized to reduce data redundancy and remove partial and transitive functional dependencies. The methods followed were 1NF, 2NF, and 3NF to implement this. As of now, all relations in the database are in 3NF.

2. Overview

The intended database is an integral part of a web application intended to convey the student fraternity of the campus regarding all the scheduled events. The intended database will be connected with PHP as part of the backend. This use of the above application is mainly intended for the student community all well as the Event Organising groups of the National Institute of Technology Calicut (NITC).

● This project enables all the Department Associations like CSEA, MEA, Technical & Cultural Clubs, and Social Initiatives like NSS to post all their events on a unified platform for students to view.

● The organizing teams of the respective events are the biggest benefactors, as the word of their event reaches out to the entire student community.

● This enables the organizing team to schedule the event accordingly, in order to avoid clashes between two events.

● Students can easily view and register for the event of their choice.

This **Database Design Document (DDD)** is intended for the database designer,system designers,and web developers.

● The client can use the DDD to verify if the product is acceptable to his/her specifications. Specifications here imply speed and correctness of the data queried and retrieved.

● The system designers can use the DDD to design the system in such a way that it meets the requirements of the client.

● The web developer can use the DDD to develop the web functionality to design the webpage(front-end and back-end) as per the clients requirements.

3. Assumptions/Constraints/Dependencies

3.1 Assumptions

- ➔ As this software is made to work exclusively for NITC, the roll no. is assumed to be of length 9 alpha-numeric characters of the format ({A-Z,1}{0-9,6}{A-Z,2}).
- ➔ All the NITC users are already added to our database and everyone is given an initial default password for their login to the portal.
- ➔ There is a separate relation which contains the unique user ids and password for the login of the Organizers. This is to ensure that in case of any data breaches, the sensitive access to the registrants list and all other event related data is not compromised.

3.2 Constraints

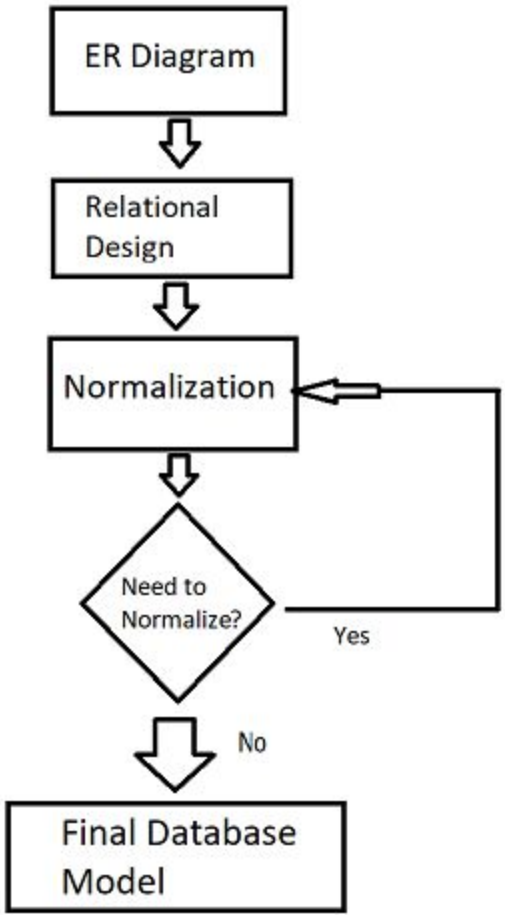
We already made the User database which has the essentials for login and authentication so that to avoid unauthorized logins

- Safety and Security: Users are authorized to access the website and login with their respective username and password credentials only.
- Availability: The database server should be active all day long.
- Faster Query Time: The challenges in developing the product will involve scaling it to the required number of stakeholders. The expected number of registered users once launched will be over 4000(strength of NITC community). So providing fast request response time will be a necessity.

3.3 Dependencies

1. Apache, MySQL and PHP for runtime environment.
2. Separate User Login and Event Organizer Login Relations are maintained so as to ensure no theft of login data in case of data breaches.

4. Design Decisions



How our Database Design was approached

In the initial stages when we made the relational schema, it had some redundancies and was not normalised, so we tried to reduce the redundancies and ensured every table is 3NF normalised.

A database which is unnormalized may suffer data redundancy, where multiple values are present in a single field or attribute of a relation or fields are repeated within the same table.To avoid or reduce such redundancies, the database should be normalized.

When we converted the EER diagram to RDB, all relations in our database already satisfied 1NF. We had no partial dependencies and transitive dependencies when we looked at the final database.

4.1 Key Factors Influencing Design

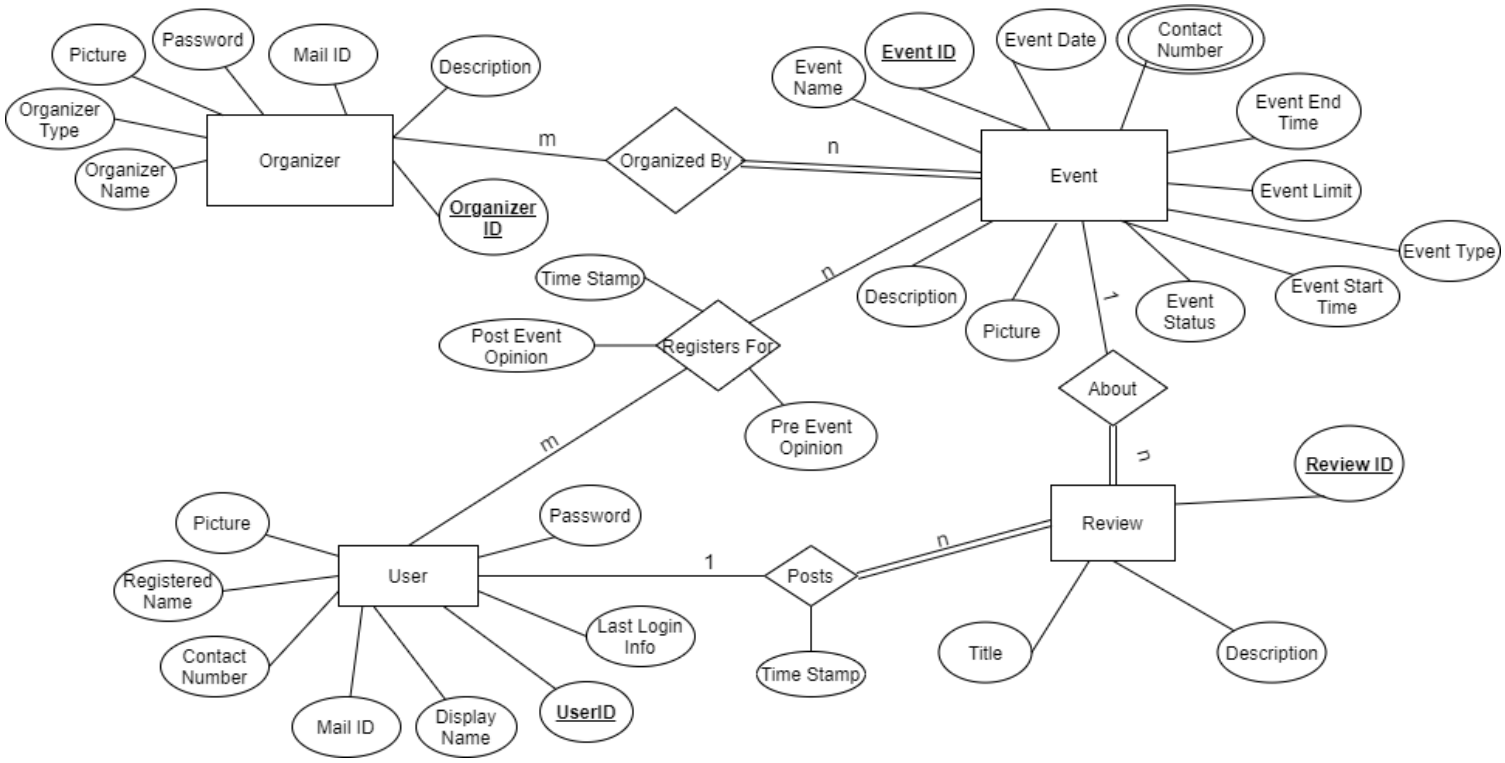
- ➔ The redundancy on update of some attributes like location and type of the organizer.
- ➔ The above redundancy is corrected by having a separated table for these attributes and referencing them as foreign keys.
- ➔ A separate table is created for the login of user and organizer. This hides the authentication details of the users from direct queries on the database.

4.2 Mapping Rules

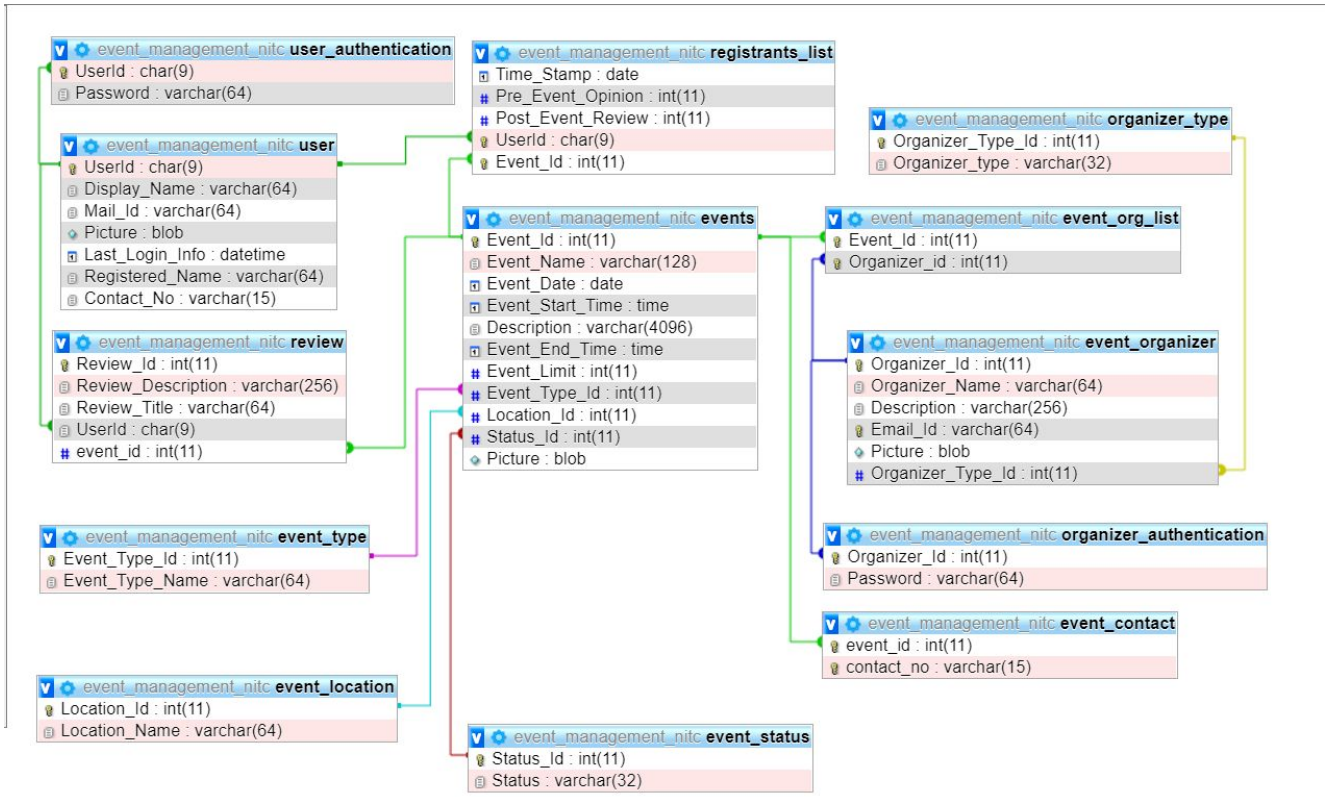
- ➔ Create Tables for each entity
- ➔ Entities attributes are the fields of the tables with their respective data types.
- ➔ Declare primary Key for each table
- ➔ Create a Table for each relationship
- ➔ Add primary keys of all participating entities as fields of the table with their respective data types.
- ➔ The attributes of the relationship are also added as the fields of this table.
- ➔ The primary key of this table is the set of all the primary keys of the participating entities.
- ➔ ForeignKey constraints are declared.

5. Detailed Database Design

5.1 Database Management ER Diagram



5.2 Database Management Relational Diagram



5.3 Relations in the Database

Event:

Column	Type	Null	Default	ForeignKey
Event_Id (Primary)	int(11)	No		
Event_Name	varchar(128)	No		
Event_Date	DATE	No		
Event_Start_Time	TIME	No		
Description	varchar(4096)	No		
Event_End_Time	TIME	No		
Event_Limit	int(11)	No		
Organizer_Id	int(11)	No		Yes
Event_Type_Id	int(11)	No		Yes
Location_Id	int(11)	No		Yes
Status_Id	int(11)	No		Yes

Picture	blob	Yes	NULL	

Event_Org_List:

Column	Type	Null	Default	ForeignKey
Event_Id (Primary)	int(11)	No		Yes
Organizer_Id (Primary)	int(11)	No		Yes

Organizer:

Column	Type	Null	Default	ForeignKey
Organizer_Id (Primary)	int(11)	int(11)	No	
Organizer_Name	varchar(64)	No		
Description	varchar(256)	No		
Email_Id	varchar(64)	No		
Picture	blob	Yes	NULL	
Organizer_Type_Id	int(11)	No		ForeignKey

Organizer_Authentication:

Column	Type	Null	Default	Foreignkey
Organizer_Id	int(11)	No		Yes
Password	varchar(64)	No		

Event_Contact:

Column	Type	Null	Default	Foreignkey
Event_Id (Primary)	int(11)	No		Yes
Contact Number (Primary)	varchar(15)	Yes	NULL	

Event Location:

Column	Type	Null	Default	Foreignkey
Location_Id (Primary)	int(11)	No		
Location_Name	varchar(64)	No		

Event Status:

Column	Type	Null	Default	ForeignKey
Status_Id (Primary)	int(11)	No		
Status	varchar(32)	No		

Event Type:

Column	Type	Null	Default	ForeignKey
Event_Type_Id (Primary)	int(11)	No		
Event_Type_Name	varchar(64)	No		

Organizer Type:

Column	Type	Null	Default	ForeignKey
Organizer_Type_Id (Primary)	int(11)	No		
Organizer_type	varchar(32)	No		

Registrants List:

Column	Type	Null	Default	ForeignKey
Time_Stamp	date	No		
Pre_Event_Opinion	int(11)	Yes	NULL	
Post_Event_Review	int(11)	Yes	NULL	
UserId (Primary)	char(9)	No		Yes
Event_Id (Primary)	int(11)	No		Yes

Review:

Column	Type	Null	Default
Review_Id (Primary)	int(11)	No	
Review_Description	varchar(256)	No	
Review_Title	varchar(64)	No	
UserId	char(9)	No	Yes
event_id	int(11)	No	Yes

User:

Column	Type	Null	Default	ForeignKey
UserId (Primary)	char(9)	No		
Display_Name	varchar(64)	No		
Mail_Id	varchar(64)	No		
Picture	blob	Yes	NULL	
Last_Login_Info	datetime	Yes	CURRENT_TIMESTAMP	
Registered_Name	varchar(64)	No		
Contact_No	varchar(15)	Yes	NULL	

User Authentication:

Column	Type	Null	Default	ForeignKey
UserId (Primary)	char(9)	No		Yes
Password	varchar(64)	No		

5.4 Some Common Queries Associated with the Database

1. Retrieve names and contact number of all registered participants for event “X”

   ```
   SELECT U.UserId, U.Registered_Name, U.Mail_Id, U.Contact_No
   FROM REGISTRANTS_LIST as R_L
   JOIN USER as U
   JOIN EVENTS AS E
   ON U.UserId = R_L.UserId
   AND R_I.Event_Id = E.Event_id
   WHERE E.Event_Name = “X”
   ```

2. Retrieve all the events a User “T” participated

   ```
   SELECT U.Registered_Name, U.Mail_Id, U.Contact_No, E.Event_Name
   FROM User as U
   JOIN Events as E
   JOIN Registrants_List as R_L
   ON U.UserId = R_I.UserID AND
   R_I.Event_Id = E.Event_id
   WHERE U.Registered_Name = “T
   ```

Appendix

Appendix A: Record of Changes

Table 1 - Record of Changes

Version Number	Date	Author/Owner	Description of Change
1.0.0	15/11/2020	G-16	Relational Database Schema before Normalization
1.1.0	24/11/2020	G-16	Relational Database Schema Post Normalisation

Appendix B: Acronyms

Table 2 - Acronyms

Acronym	Literal Translation
DDD	Database Design Document
ER Diagram	Entity Relationship Diagram
RDBMS	Relational Database Management System
1NF	1st Normal Form
2NF	2nd Normal Form
3NF	3rd Normal Form

Appendix C: Glossary

Table 3 - Glossary

Term	Definition
Database	Organized collection of information or data.
ER Diagram	ER Diagram is a graphical representation of entities and their relationships.
Functional Dependency	If the information stored in a table can uniquely determine another information in the same table, then it is called functional dependency.
Normalisation	Process of minimizing redundancy from a relation or set of relations.
RDBMS	A relational database is a collection of information that organizes data points with defined relationships for easy access

Appendix D: Referenced Documents

Table 4 - Referenced Documents

Title/Website
Fundamentals of Database Systems, 7/e, Pearson Education, 2016
https://www.guru99.com/database-design.html
https://www3.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html