

Diary Management System

Database Management

CMPT 308-210

Team Sweet Dreams



Marist College

School of Computer Science and Mathematics

Submitted To:

Dr. Reza Sadeghi

Fall 2023

Project Report of Diary Management System

Team Name

Team Sweet Dreams

Project

Diary Management System

Team Members

1. Evan Spillane Evan.Spillane1@marist.edu (Team Head)
2. Connor Fleischman Connor.Fleischman1@marist.edu (Team Member)
3. Lilli Cartiera Lillianana.Cartiera1@marist.edu (Team Member)
4. Abel Scholl Anna.Scholl1@marist.edu (Team Member)
5. Neo Pi Neo.Pi1@marist.edu (Team Member)

Description of Team Members

1. Evan Spillane

I am a dual major in Cybersecurity and Philosophy, and think they balance each other out very well. Computer Science is based on math which is a field based on concrete answers whereas Philosophy is a field that lacks any concrete answers. I am from Valley Forge, PA (which is a little bit outside Philadelphia). Our group consists of the people sitting in the front/front-left of the classroom but we quickly got acclimated with each other. Lilli is my best friend (hence why they were sitting next to me in class) and I also lived with Neo last year, and he is in the Esports program with me. I volunteered to lead the group because I am confident in my project management and communication skills. I think my experience as 2025 Class president as well as the president of Marist Esports, and the Chair of the Rules & Administration Committee will be useful in helping my group thrive and do our best work!

2. Connor Fleischman

I am a Computer Science major from Wingdale, NY. I've got a little brother, Shane, who's currently a sophomore in high school and the captain of the varsity soccer team. My love of coding originated when I was little. Although opening the Command Prompt may not seem like much, as a ten-year-old it made me feel like a spy. Choosing my group was easy as all I did was turn around and ask. But the hard part came when we

tried to think up a name for us. It took a while but after many failed attempts we finally came up with one that fit. Sweet Dreams, brought up by Abel initially, was a perfect fit as by 9:00 p.m. on a Monday, everyone just wants to go home and have sweet dreams. As for the decision of the team head, Evan stepped up to fill the role and the group unanimously agreed. Along with this, we also chose the task of creating a Diary Management system as our project which I, along with the rest of the group, look forward to working on.

3. Lilli Cartiera

I am an Applied Math and Data Science double major from Wethersfield, Connecticut. I enjoy my major because making sense of the world using numbers is my strength. In my free time, I play piano and video games with my friends. On campus, I am the Vice President of the Math Club and I represent Marist's chapter of AWM (American Women in Mathematics). I am best friends with Evan and therefore I know he is a great leader. To me, he was an obvious choice. He is active on campus and passionate about creating tangible positive change. Our team was drawn together because we all sat near each other in class. Now that I know them better I look forward to working with them; so far they have taken great initiative and are enthusiastic about this project.

4. Abel Scholl

I am a Computer Science major from Wilkes-Barre, PA. I enjoy coding, puzzle games, and anything art-related, especially drawing and crocheting. I approach my major like an artist and let the creation aspect of writing code drive me to learn as much as I can and apply it wherever I can. This group is made up of people who were sitting closest to me when it came time to choose groups, but the collective effort to actively include everyone is what stood out to me about everyone here. It is easy to ignore and avoid what is unfamiliar or to prematurely judge based on initial perception, so I am looking forward to working together with the different personalities and perspectives brought to the table. Furthermore, I believe Evan will be a good team head judging by the way he volunteered. To me, this shows confidence in management which is essential for a good leader.

5. Neo Pi

I am a Cybersecurity major born and raised in South Jersey. Growing up, I always wanted to be a doctor and did not think I would learn to code until my freshman year of college. I chose this as my major over medicine because growing up as a kid I was always interested in video games and what a computer can do. The simplest way to choose a group was to team up with people around you, which led me to join this group. I

chose Evan as the team head because I know based on past experience that he is well-organized, intelligent, and capable of being a group leader.

Table of Contents

Project Report of Diary Management System.....	2
Table of Contents.....	5
Table of Figures.....	6
Project Objective.....	7
Related Work.....	8
Project Merits.....	9
GitHub Repository.....	10
Entity Relationship Model.....	11
Enhanced Entity Relationship Model.....	18
References.....	20

Table of Figures

1. Figure 1: Entity Relationship Model (ER).....	11
2. Figure 1: Enhanced Entity Relationship Model (EER).....	18

Project Objective

The purpose of the Diary Management System is to create a Diary Management System GUI using Python, the Tkinter library, and SQL. This system should allow a user to record daily events and experiences and store the data in separate SQL tables. It will include separate administrator and normal user logins with different role-based permissions that allow different levels of access to features. Users should be able to add, edit, and delete records as well as search for records by time, place, or duration. Finally, the Diary Management System will also have settings where the admin can moderate users and their information, user-friendly warnings and alerts, a feature menu, and an exit function.

Related Work

Businesses related to our Project

Business 1: EventReference - <https://www.eventreference.com/diary-system.php>

Positives: Efficient ideas, organized concept

Negatives: UI is too complicated for new users to navigate

Business 2: Setmore - <https://www.setmore.com/diary-management>

Positives: Extremely friendly UI, many different tools available

Negatives: Does not have multi-user data management features in certain versions

Business 3: Client Diary - <https://clientdiary.com/>

Positives: Large scale with many tools (especially in its premium version)

Negatives: Expensive, some tools may be useless to users with less complex data

The diary system can have multiple fields for each employee. These fields can include job title, country, company details, organization type, and various questions that are relevant to the diary system's purpose. For example: what type of events do you organize? What is your annual budget for meetings and events? Which products or services are you interested in for your meetings and events? What locations are you interested in for your meetings and events?

A diary can also be used for staff to message each other and set up meetings. The diary can also be used to match buyers with suppliers based on their individual preferences. Each party can select those they are interested in meeting with. A ranking system can also be utilized.

Lastly, the information each user puts into the diary can be used to create a schedule of meetings throughout a period of time. We can do this by examining each user's rankings of other users. The timing and frequency of these meetings can be planned based on this information.

Project Merits

Adopting the Diary Management System will allow all workplace employees to manage their day-to-day happenings. Whether the entry is a meeting, special event, time away, reminder, or just a simple note, keeping track of these details helps take the hassle out of planning and keeps life running smoothly. Sort your team into organizations, assign roles to members, color code entries, and share diaries to help make teamwork as productive, efficient, and stressless as possible, or simply use it for personal business. Either way, the Diary Management System will carry the weight of event management so you do not have to.

GitHub Repository

https://github.com/CainSpillane/201_DiaryManagementSystem_SweetDreams

Entity Relationship Model

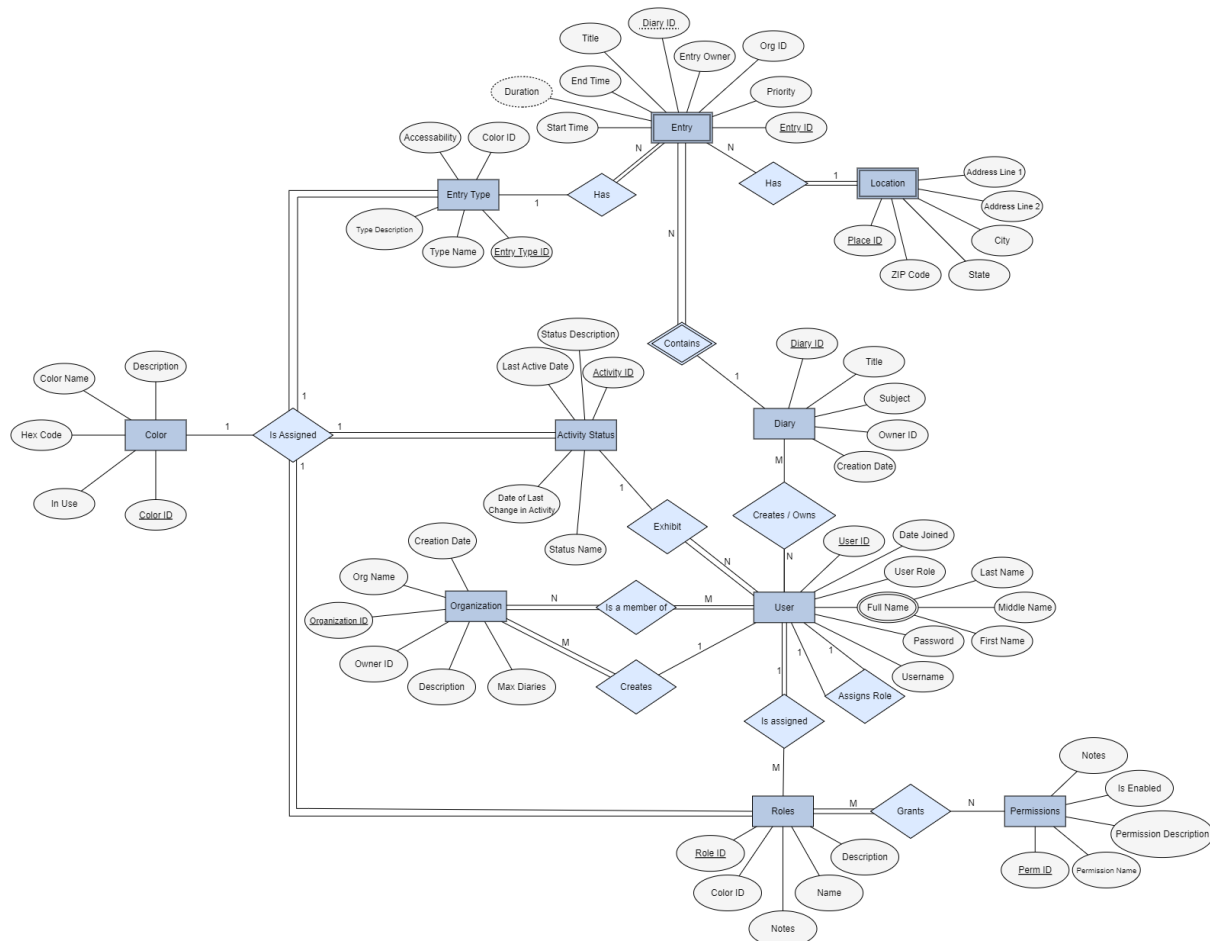


Figure 1: Entity Relationship Model (ER)

To display the mini-world we selected entities that relate to each other. First, we selected a user entity that represents individuals who can access the system. This entity is a central part of the mini-world as it interacts with other entities. This was a starting point for us. Then, we added a list of applicable attributes to help lay out what each entity can do. For example, in user, the attributes contained important information: full name, User ID, user role, username, and password. To be more specific, the full name is split into more attributes: first, middle, and last. Then, we represented the relationships of users with a “create/owns action”, “is assigned”, and “is a member of”. With the use of these relationships, we can create further entities by extending connecting participations. For example, “users” are related to “organizations” by the relationship “is a member of”. Since all users have to be members in order to have access, we get an instance of total participation. Likewise, all members are required to be in an organization. Then we related this organization entity to further entities: shared diary, diary, and entries. Determining

cardinalities is based on the relationship between two entities. For example, the cardinality of users to organizations is many to many. Finally, we created another entity that can branch off the first, and input more attributes and relationships. By starting from a user entity, we were able to break down the diary management system.

Listed below are important details and descriptions of all of the entities, as well as their attributes, relationships, and cardinality.

User

- Description
 - A representation of a user for the purpose of interacting with the DMS.
- Attributes
 - User ID
 - Uniquely identifies the user and thus is the Primary Key
 - Full Name (Multivalued Attribute)
 - First Name
 - Middle Name
 - Last Name
 - User Role – determines the permissions a user has
 - Date Joined – the date the user’s information was created by the admin
 - Username
 - Password
- Relationships
 - Users can create diaries
 - Users (Administrator) can assign roles to users; This relationship is recursive
 - Users are a member of an organization
 - Users (Administrators) create Organizations
- Cardinality
 - All users are members of an organization, which makes this a many-to-many (M:N) relationship
 - The cardinality for users creating diaries is 1:N
 - Users can create an organization, which is a 1:N Relationship
 - Users assign roles to users in a 1:N relationship

Organization

- Description
 - An organization is an entity that has 1 or more user members and can be used to have shared diaries and shared entries with users that belong to the same organization. For example, this could be used by a company to have their

employees have a shared diary space, or by a family who wants to have a shared diary space

- Attributes
 - Organization ID
 - Uniquely identifies an organization and, thus is the Primary Key
 - Organization Name
 - Owner ID
 - Creation Date
 - Diary Limit
- Relationships
 - Organizations contain users
 - Organizations are created by users
- Cardinality
 - All organizations must contain members (full participation)
 - All organizations must be created by a user (full participation)

Diary

- Description
 - The diary entity is used to house and organize multiple user inputs while keeping them separate based on certain criteria. Diaries contain Entries, which contain the user's textual inputs to the DMS. Diaries can also be sorted and organized based on dates, subjects, and titles
- Attributes
 - Diary ID
 - Uniquely identifies an organization and, thus is the Primary Key
 - Owner ID
 - Title
 - Subject
 - Creation Date
- Relationships
 - Diaries must be created by a user
 - Diaries contain Entries
- Cardinality
 - Diary has an M:N relationship with User
 - Diary has a 1:N relationship with Entry

Entry

- Description
 - Entry is a weak entity that contains data about the entries within a given diary

- Attributes
 - Entry ID
 - Uniquely identifies an Entry and, thus is the Primary Key
 - Diary ID (Partial Key)
 - Entry Owner
 - Organization ID
 - Priority
 - Start Time
 - End Time
 - Duration
- Relationships
 - An Entry must have a Diary in which it is contained.
 - Entries must have an Entry Type
 - Entries can have a Location
- Cardinality
 - Entry has an N:1 relationship with Diary
 - Entry has an N:1 relationship with Entry Type
 - Entry has an N:1 relationship with Location

Entry Type

- Description
 - Entry type is an entity that establishes characteristics that are applied to an Entry.
All Entries must have an entry type.
- Attributes
 - Entry Type ID
 - Uniquely identifies an Entry Type and, thus is the Primary Key
 - Entry Type Name
 - Entry Type Description
 - Accessibility
 - Color ID
- Relationships
 - Entry Type Can be associated with an entry
 - Entry Type is assigned a color
- Cardinality
 - Has a 1:N relationship with Entry
 - Has a 1:1 relationship with Color

Location

- Description

- Location is a weak entity that contains information about the place of a given entry, should a user wish to input one
- Attributes
 - Location ID
 - Uniquely identifies a Place/Location and, thus is the Primary Key
 - Address Line 1
 - Address Line 2
 - City
 - State
 - ZIP Code
- Relationships
 - Location is associated with an Entry
- Cardinality
 - Location has a 1:N relationship with Entry

Role

- Description
 - Role is an entity that determines levels of system privilege for a given user. One example of a role is Administrator, and Admins can assign roles to other users
- Attributes
 - Role ID uniquely identifies a Role and, thus is the Primary Key
 - Role Name
 - Role Description
 - Color ID
 - Role Notes
- Relationships
 - A Role is assigned to a user by another user (administrator)
 - Roles grant permissions (from the Permission entity) to the role recipient based on the ID of the Role that was assigned
- Cardinality
 - Has an N:1 relationship with User
 - Has an N:M relationship with Permission

Permission

- Description
 - Permission is an entity that dictates rules regarding what a user is capable of doing on the DMS. Permissions are associated with a role, and roles are granted to users. Users then inherit the permissions from their role, so long as they still have that role.

- Attributes
 - Permission ID
 - Permission Name
 - Permission Description
 - Is Enabled
 - Is Enabled is a boolean value determining whether an individual permission is enabled within a role.
 - Notes
- Relationships
 - Permissions are granted by Roles
- Cardinality
 - Has an N:M relationship with Roles

Activity Status

- Description
 - Activity Status is an entity that displays information about a user's recent activity (or lack thereof) on the DMS.
- Attributes
 - Activity ID
 - Status Description
 - Last Active Date
 - Status Name
 - Date of last change in activity
- Relationships
 - Activity Status is exhibited by a user
 - Activity Status is assigned a color based on which type of status it is.
- Cardinality
 - Has a 1:1 relationship with Color
 - Has a 1:N relationship with User

Color

- Description
 - Color is an entity that contains information about the color and associated properties, which is used to help differentiate between different types of entries, statuses, and roles.
- Attributes
 - Color ID
 - Hex Code
 - Color Name
 - Description

- In use
- Relationships
 - Color is assigned to the Entry Type entity
 - Color is assigned to the Activity Status entity
 - Color is assigned to the Role entity
- Cardinality
 - Has a 1:1 relationship with Entry Type
 - Has a 1:1 relationship with Activity Status
 - Has a 1:1 relationship with Roles

Enhanced Entity Relationship Model

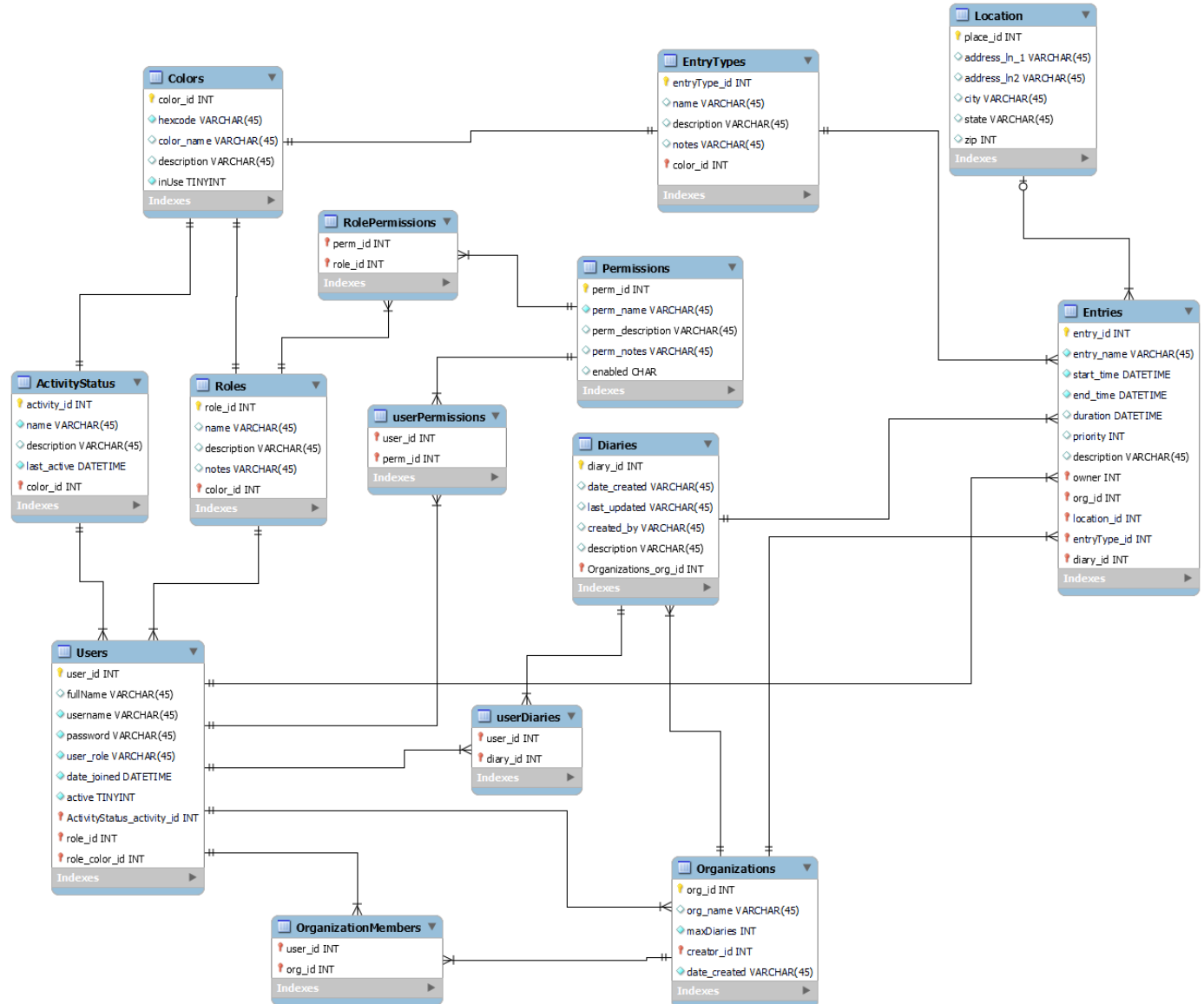


Figure 2: Enhanced Entity Relationship Model (EER)

In an EER model, keys help uniquely identify entities as their information is needed across the database. These keys help keep track of relationships between different entities without duplicating data across tables. In the DMS EER model, the entities Users, ActivityStatus, Roles, Permissions, Organizations, Diaries, Entries, Location, Colors, and EntryTypes all have specific IDs that act as primary keys within their own table and foreign keys within other tables with their data associated. For instance, Users has a primary key user_id that is used as a foreign key within Entries (shown as “owner” in Entries table) in order to establish a one-to-many relationship between Users and Entries where one user can own many entries.

Users also has foreign keys from ActivityStatus, and Roles, because every user has associated information relating to these entities; Every user must have one activity status and one role, but no more at a single time, making the relationships one to many. Though User is related to ActivityStatus, Role, and Entries, Entries does not carry a foreign key of any entity in this list except for Users, because an entry does not need to know anything about a user except which specific user it is. Some relationships, such as Users to Diaries, have many-to-many relationships. In this case, One user can have access to many diaries, and a diary can be accessed by many users. Rather than have Users carry the key of every single diary it has access to, another table is implemented that holds only the primary keys of Users and Diaries and keeps track of every combination as necessary. Finally, other different types of relationships include one-to-one and zero-or-one-to-many. The relationship between ActivityStatus and Colors is one-to-one, because one status can be associated with only one color and vice versa. The relationship between Entries and Location is zero-or-one-to-many, because a location can be attached to more than one Entry at a time, but not every entry has to have a location. If it is associated with a location, it can have only one. The other relationships within the model are all handled in the same ways described above and are as follows:

- ActivityStatus to Users: one to many
- Roles to Users: one to many
- Users to Entries: one to many
- Users to Organizations: many to many
- Users to Diaries: many to many
- Roles to Permissions: many to many
- Colors to ActivityStatus: one to one
- Colors to Roles: one to one
- Colors to EntryTypes: one-to-one
- EntryTypes to Entries: one to many
- Location to Entries: zero or one to many
- Organizations to Entries: one to many
- Diaries to Entries: one to many
- Organizations to Diaries: one to many

References

1. <https://www.eventreference.com/diary-system.php>
2. <https://www.setmore.com/diary-management>
3. <https://clientdiary.com/>