Building Web-based Applications with MERNStack

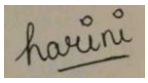
Milestone: Conceptual Model(EER & UML)

Group 13
Harini Prasad Vasisht
Sushmitha Sudharsan

+1(984)374-4836 (Harini Prasad Vasisht)
+1(857)565-8800 (Sushmitha Sudharsan)
vasisht.h@northeastern.edu
sudharsan.s@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

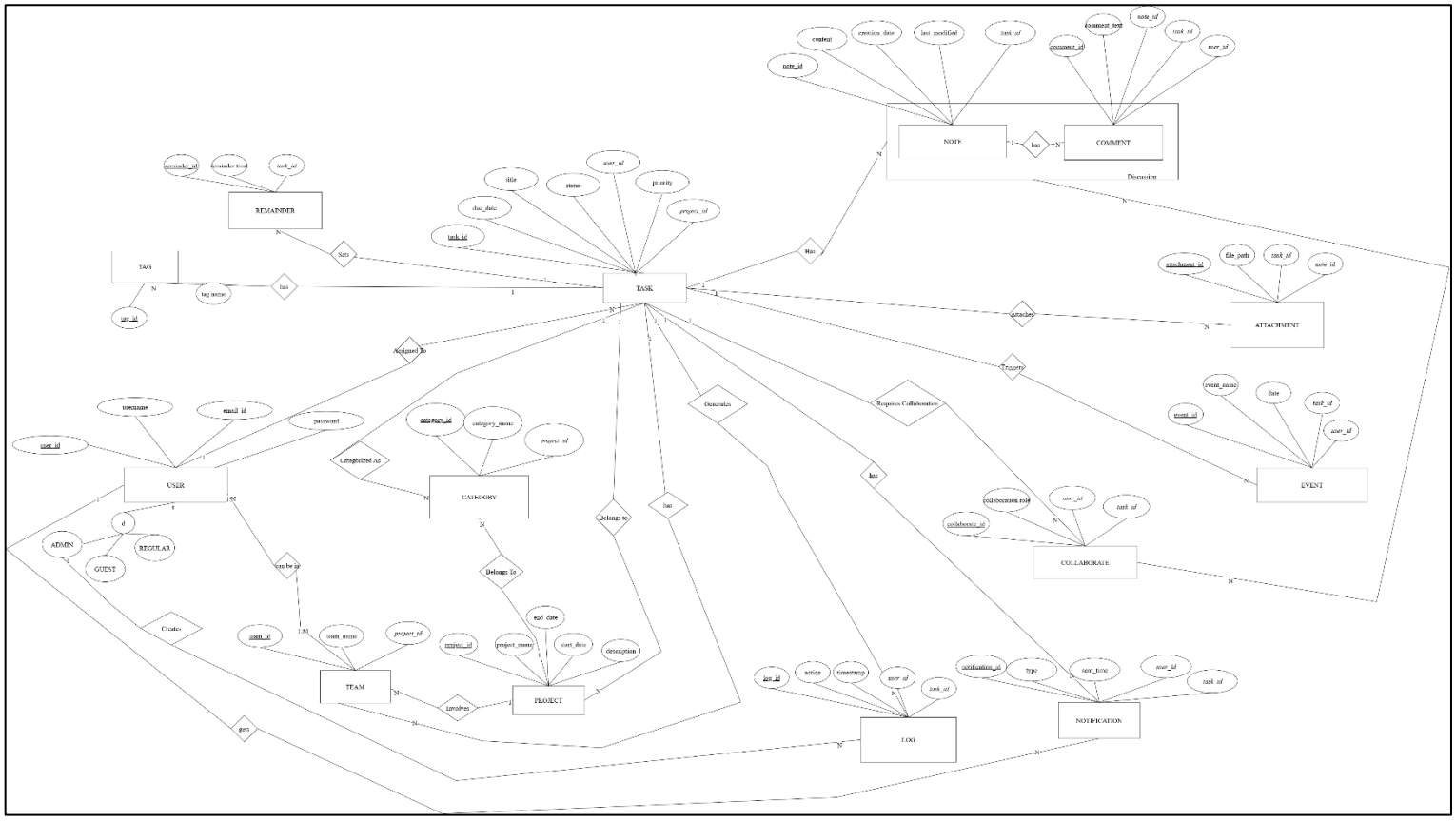
Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: 

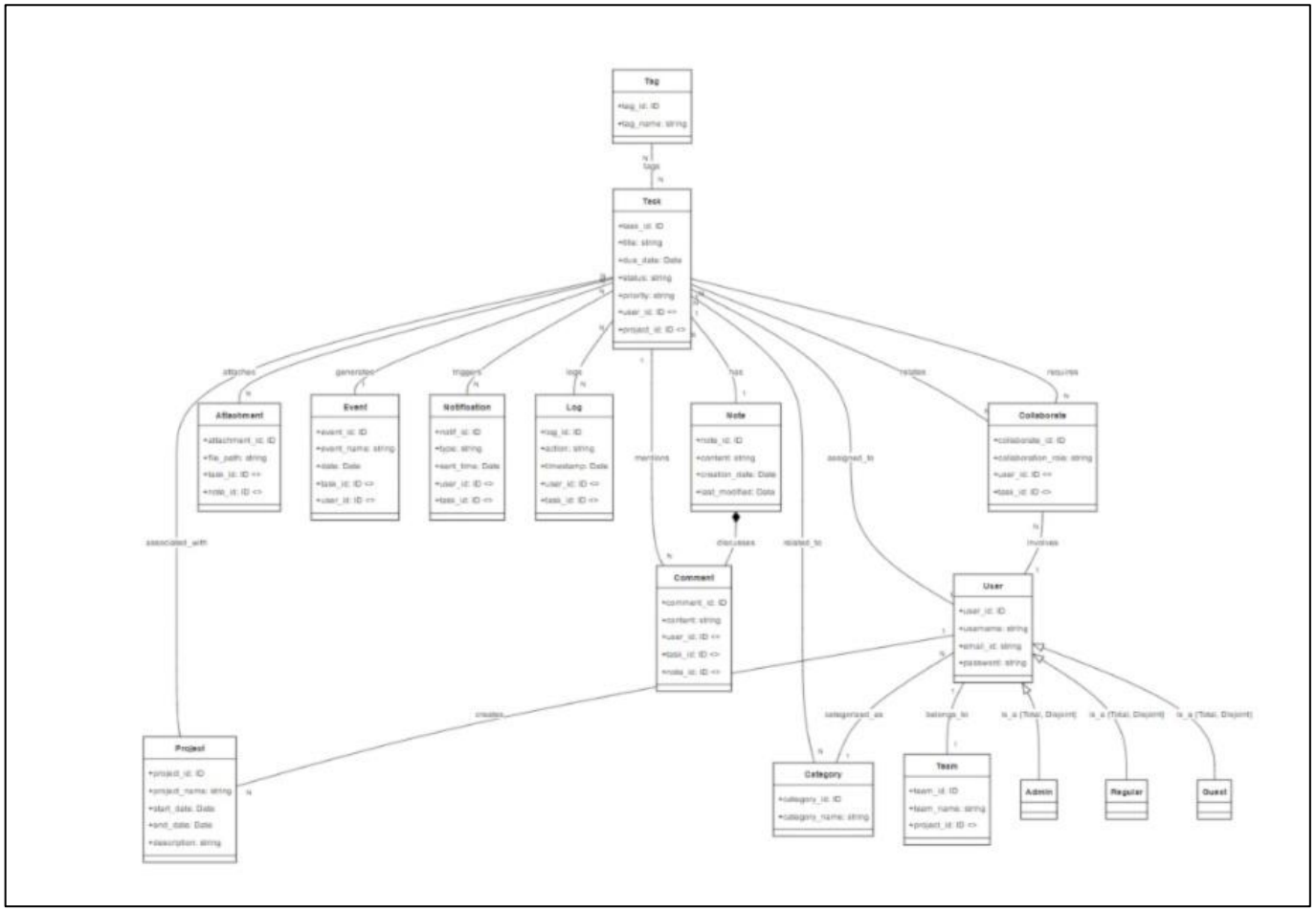
Signature of Student 2: 

Submission Date: 10/13/2024

EER DIAGRAM:


NOTE
COMMENT
Has
Discussion
REMAINDER
Sets
TAG
has
tag name
tag_id
TASK
title
status
priority
due_date
task_id
project_id
user_id
ATTACHMENT
Attaches
EVENT
Records
Assigned To
Generates
Requires Collaboration
USER
username
email id
password
user_id
Categorized As
CATEGORY
category_id
category_name
project_id
has
ADMIN
GUEST
REGULAR
can be of
Belongs To
Belongs to
Creates
TEAM
team_id
team_name
project_id
PROJECT
end_date
start_date
description
involves
gets
COLLABORATE
collaboration role
collaboration_id
LOG
log_id
action
timestamp
NOTIFICATION
notification_id
type
sent_time

UML DIAGRAM:


Tag
Task
Attachment
Event
Notification
Log
Note
Collaborate
Comment
User
Project
Category
Team
Admin
Regular
Guest

EXPLANATION:

1. ENTITIES:

Entity	Attributes
1.User	`user_id (PK)`, `username`, `password`,email
2.Task	`task_id (PK)`, `title`, `status`, `due_date`, `user_id (FK)`,project_id(FK)
3.Category	`category_id (PK)`, `category_name`, `project_id (FK)`
4.Log	`log_id (PK)`, `action`, `task_id (FK)`, `user_id (FK)`, `timestamp`
5.Notification	`notification_id (PK)`, `event_type`, `sent_time`, `user_id (FK)`, `task_id (FK)`
6.Note	`note_id (PK)`, `content`, `creation_date`, `last_modified`, `task_id (FK)`
7.Comment	`comment_id (PK)`, `comment_text`, `note_id (FK)`, `task_id (FK)`, `user_id (FK)`
8.Reminder	`reminder_id (PK)`, `reminder_time`, `task_id (FK)`
9.Tag	`tag_id (PK)`, `tag_name`
10.Attachment	`attachment_id (PK)`, `file_path`, `task_id (FK)`, `note_id (FK)`
11.Project	Project_id(PK),Project_name,start_date,end_date,description
12.Team	Team_id(PK),team_name,project_id
13.Collaborate	Collaborate_id(PK),user_id(FK),note_id(FK),collaboration_role
14.Event	Event_id(PK),event_name,date,task_id(FK),user_id(FK)

2. Relationships:

Relationship	Entities Involved	Cardinality	Description
1. Creates	User → Task	1:M	A user can create multiple tasks.
2. Triggers	User→Notification	1:M	A user can trigger multiple notifications.
3. Logs	User → Log	1:M	A user can log multiple actions in tasks.
4. Organizes	USER → CATEGORY	1:M	A user organizes tasks into multiple categories.
5. Has	User → Comment	1:M	A user can have multiple comments.
6. Has	Task → Note	1:M	A task can have multiple notes.
7. Sets	Task → Category	1:1	A task belongs to one category.

8. Labels	Task → Tag	M:M	A task can have multiple tags.
9. Attached	Task→ Attachment	M:M	A task can have multiple attachments.
10. Reminds	Task → Reminder	1:M	A task can have multiple reminders.
11. Has	Note → Comment	1:M	A note can have multiple comments.
12. Logs	Task → Log	1:M	A task can have multiple log entries.
13. Attached	Attachment → Task	1:M	An attachment can belong to multiple tasks.
14. Attached	Attachment → Note	1:M	An attachment can be related to multiple notes.
15.Belong	User → Team	1:M	A user can belong to multiple teams and each team can have multiple users
16.Collaborate	User →Collaborate	1:M	A user can collaborate on multiple notes via the collaborate entity
17.Schedule	User →Event	1:M	A user can schedule multiple events but each event is scheduled by one user
18. Belongs	Task →Project	1:M	A task belongs to one project but a project can have multiple tasks
19.Worked	Project →Team	M:M	A project can be worked on by multiple teams and each team can be a part og multiple projects
20.Belongs	Project →Category	1:M	A category belongs to one project but a project can have multiple categories
21.Collaborate	Collaborate→Note	1:M	A user can collaborate on multiple notes via the collaborate entity. Each entity collaboration entry relates to one note but a note can have multiple collaborators

3. Foreign Keys:

Entity	Foreign Key	References (Primary Key in)
REMAINDER	task_id	TASK
TASK	user_id	USER
TASK	project_id	PROJECT
CATEGORY	project_id	PROJECT
NOTE	task_id	TASK
COMMENT	note_id	NOTE
COMMENT	user_id	USER
ATTACHMENT	task_id	TASK
ATTACHMENT	note_id	NOTE
TEAM	project_id	PROJECT
COLLABORATE	task_id	TASK
COLLABORATE	user_id	USER
NOTIFICATION	user_id	USER
NOTIFICATION	task_id	TASK
EVENT	task_id	TASK
EVENT	user_id	USER
LOG	user_id	USER
LOG	task_id	TASK

3. Inheritance, Specialization, and Generalization**:**

USER Subclass(Specialization):
The user entity is specialized into three subclasses:

- Admin: Represents users with administrative privileges.
- Regular User:Represents standard users who can create tasks,organize them and interactwith notifications,comments and logs
- Guest: Represents users with restricted or limited access.

This specialization is disjoint, meaning user can only belong to one subclass at a time.

5. Aggregation:

Aggregation in an EER diagram is a way to group multiple entities into a larger, composite entity to show a meaningful relationship. Here's what it means for **Note** and **Comment**:

- **Note** and **Comment** both contribute to an overarching concept of a **Discussion**.
- A **Note** represents an individual piece of content, while a **Comment** is typically a response or addition to that Note.

By aggregating them, the model implies that **Comments** and **Notes** together form a **Discussion** entity, which represents a more complex concept than either entity alone.
In practical terms, aggregation here indicates that **Discussion** includes both notes and comments as integral parts, providing a structured view of user interactions or collaborative conversations around tasks or projects. Aggregation clarifies the conceptual structure by showing that **Note** and **Comment** are not isolated but are part of a broader **Discussion** context.