CS 5200 Project Proposal

1. Group name and members

Team Name: *SunXHoC*
Team Members: *Ho, Chak Sing; Sun, Xueyuan*

2. Project description

PaperTrail is a research-publication management platform for professors, graduate students, and research groups. It organizes papers, collaborators, venues, funding sources, and revision history in one relational database. The application provides **CRUD** operations through a web-based interface built with Next.js and a MySQL backend.

Key User Capabilities

- Add and manage papers with metadata (title, abstract, venue, authors).
- Track co-author contributions and author order.
- Record submission status, revisions, and funding associations.
- Visualize research output trends and collaboration networks.

3. Database description

The database is **relational**, implemented in **MySQL 8.0**, normalized to **Third Normal Form (3NF)**.
Each table uses an id as the primary key and foreign keys to enforce referential integrity.

Entity	Attributes
User	id, name, email, affiliation, orcid, role_id, password
Role	id, role_name
Paper	id, title, abstract, status, submission_date, publication_date, pdf_url, primary_contact_id, venue_id, is_deleted
Venue	id, name, type, ranking
Authorship	id, paper_id, user_id, author_order, contribution_notes
Topic	id, topic_name
Paper Topic	id, paper_id, topic_id
Grant	id, grant_name, sponsor, start_date, end_date, reporting_requirements
Paper Grant	id, paper_id, grant_id
Revision	id, paper_id, version_label, notes, created_at
Activity Log	id, paper_id, user_id, action_type, action_detail, timestamp

4. SQL vs NoSQL

This project uses a **SQL (MySQL)** relational model to manage complex many-to-many relationships between papers, authors, grants, and revisions. The relational model ensures data consistency, normalization, and efficient querying through joins—advantages that NoSQL models lack for this type of interconnected academic data.

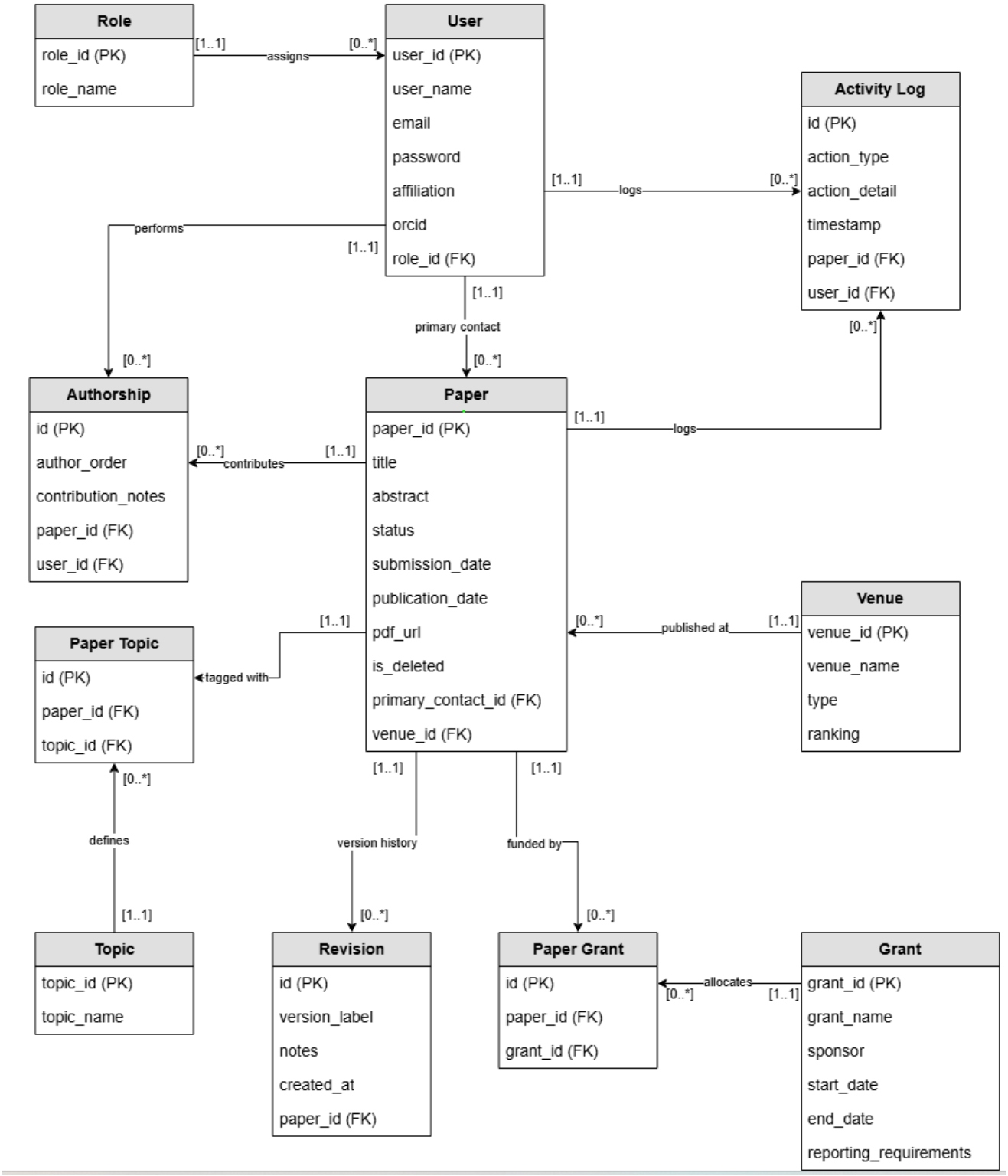
5. Tech stack

Category	Tool / Frameworks
Database	MySQL 8.0 (Docker container)
Frontend / Backend	Next.js 14 (API + Frontend)
Deployment	Deployed on Vercel, and instructions for local instalment on README.md
Object Relational Model	Prisma
Host Language	JavaScript / Typescript (prefer)
Version Control	Git
Visualization	Chart.js

6. Why interest you

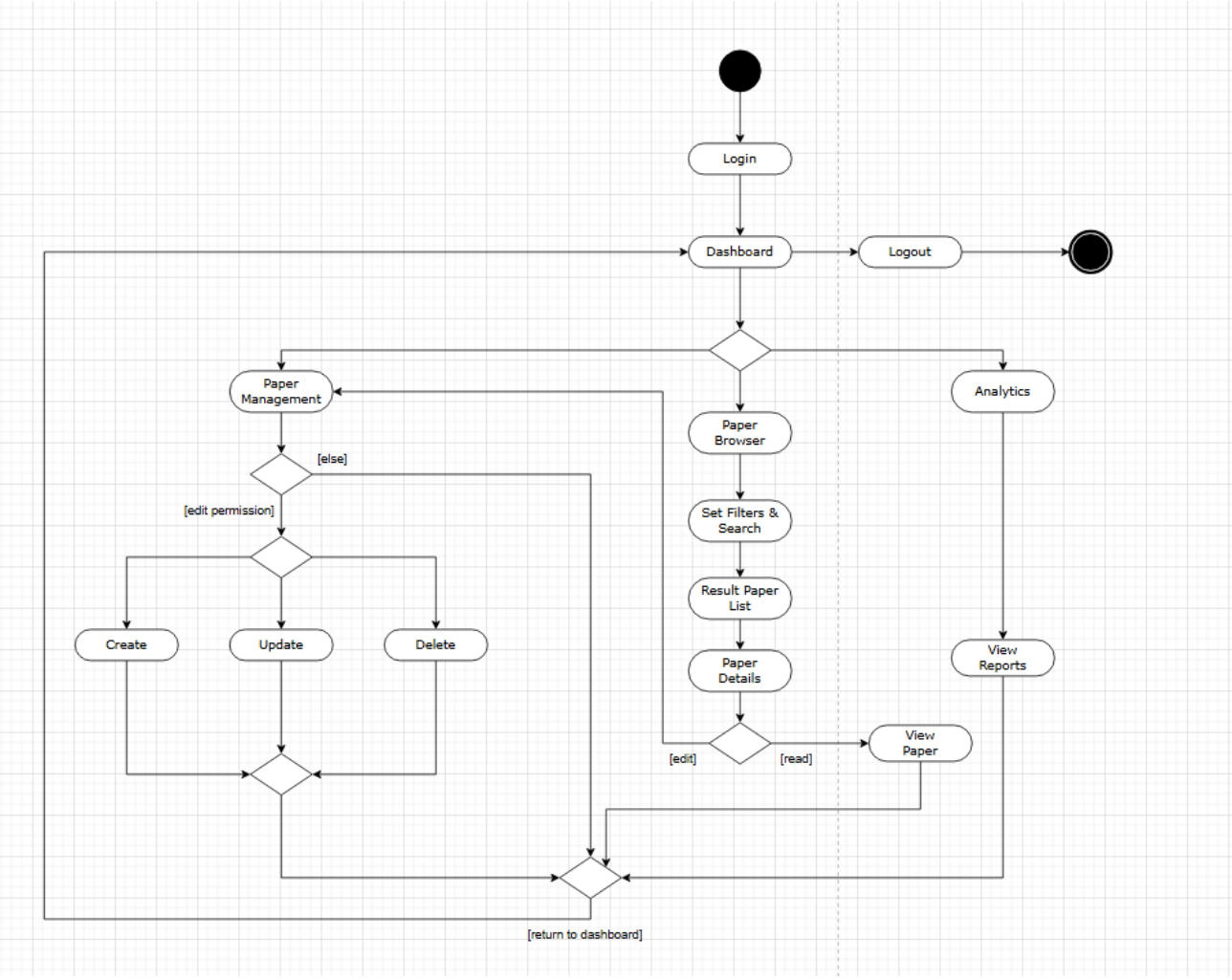
Research teams often rely on scattered spreadsheets and emails to manage papers and grants, leading to **inefficiency** and **data fragmentation**. PaperTrail addresses this by providing a unified, role-based platform built on a structured relational database that **streamlines publishing workflows** and supports **data analysis**. This domain interests us because it demonstrates strong SQL design—using keys, constraints, triggers, and events—to tackle a real academic problem.

7. UML Diagram



8. User Interaction Flow

1. **User Login** – Users access the system and log in with email and password. The system authenticates users against the User table (using *email* and *password*) and loads the dashboard.
2. **Paper Dashboard Interface** – The dashboard displays quick access to main modules:
    - **Paper Management** (to Step 3) – For authorized users to create or modify papers.
    - **Paper Browser** (to Step 5) – Search for papers, browse, edit, or view analytics.
    - **Analytics** (to Step 8) – View aggregated data and visualizations.
    - **Exit** (to Step 9) – Log out, exit the system, and end the session.
3. **Paper Management Interface**
    - **Permission Check** – The system checks the user's *role_id* (from the User table) and their relationship to the paper (Authorship table) to determine access. If permissions are insufficient, the user is redirected back to the dashboard (Step 2).
    - **Manage Operations** – Authorized users can perform the following actions:
        - **Create** – Users with the required Role (e.g., PI, Ph.D. Candidate) can create a new paper entry, filling in metadata (Paper, Authorship).
        - **Update** – Users listed in the Authorship table for a paper (or those with an elevated Role) may edit the paper. Updates include:
            i. Editing metadata in the Paper table (title, abstract, status).
            ii. Managing authors in the Authorship table.
            iii. Managing linked grants in the PaperGrant table.
            iv. Adding a new revision entry in the Revision table.
        - **Delete** – Authors or users with high-level roles may choose to delete a paper. Instead of permanent removal, the system sets the *is_deleted* attribute in the Paper table to True (soft delete to avoid losing critical data).
    - **Logging and Redirection** – All actions are recorded in the ActivityLog table. The user is redirected back to the dashboard (Step 2).
4. **Paper Browser Interface** – All logged-in users can use the system to search and view papers.
    - **Set Filters and Search** – Users can search using keywords and filter by criteria derived from the linked entities: Author (User), Topic (PaperTopic), Grant (PaperGrant), Venue (Venue), Status (Paper), and Date Range (Paper).
    - **Browse Qualifying Papers** – Users may choose to:
        - Read – Display the Paper Details page showing metadata, authors, linked topics, linked grants, revision history, and the activity log.
        - Edit – Editing papers redirects to the Management Interface (Step 3) and requires the necessary permissions check.
5. **Analytics Interface** – Users can view analysis results based on the aggregated data:
    - Papers per year (Uses Paper.publication_date and Paper counts).
    - Grant analysis (Uses PaperGrant to link papers to Grant for usage/output statistics).
    - Venue statics (Uses Venue and Paper counts to show papers per venue).
    - Topic Analysis (Uses PaperTopic to show papers per topic).
    - Author Contribution (Uses Authorship to count papers per User).
6. **User Logout and Exit** – User clicks Logout from Dashboard. The system ends the session.


Login
Dashboard
Logout
Paper Management
Analytics
Paper Browser
[else]
[edit permission]
Set Filters & Search
Result Paper List
Create
Update
Delete
Paper Details
View Reports
View Paper
[edit]
[read]
[return to dashboard]