CSE 412 - Proposal-Phase 1

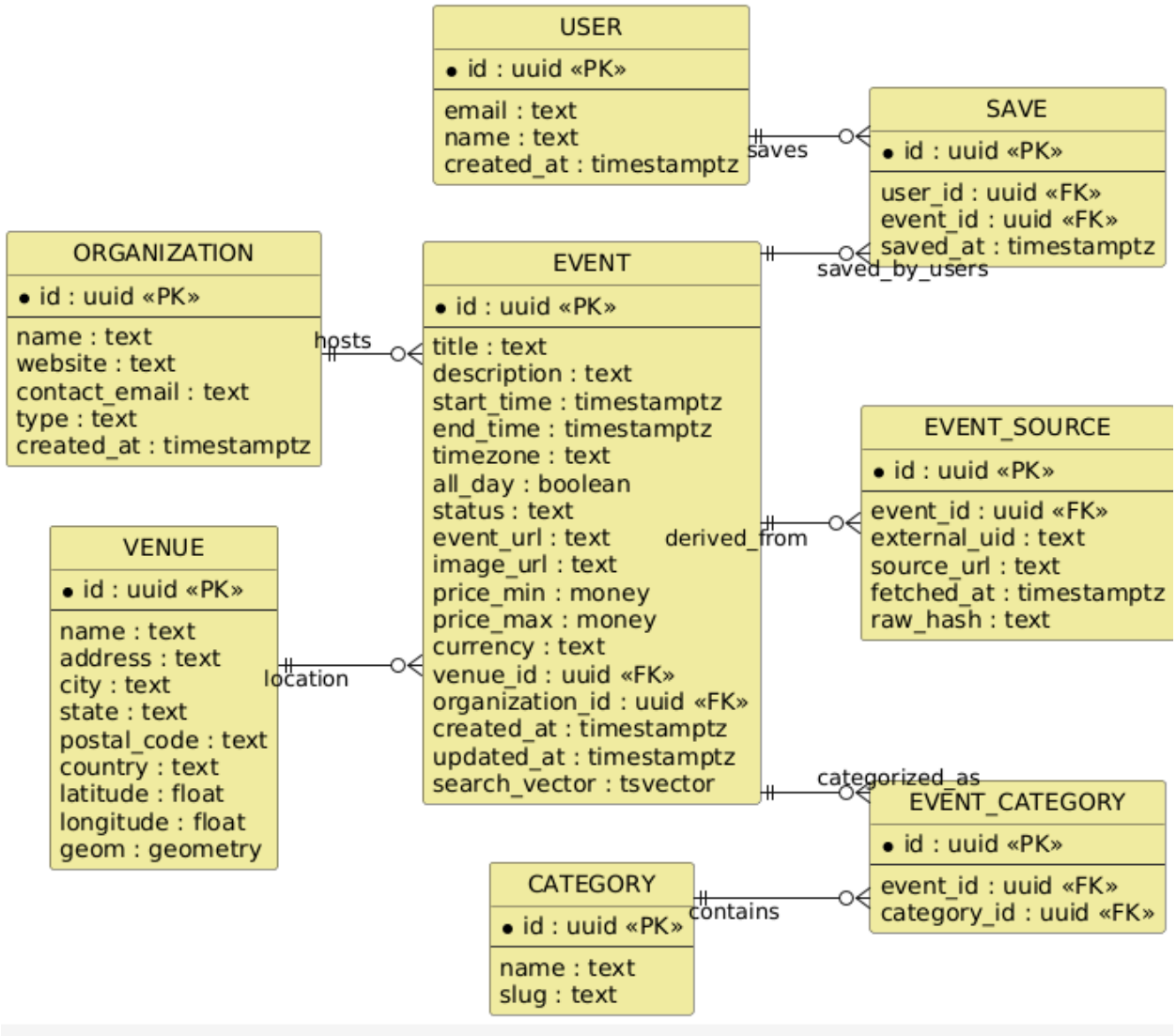
Dens Sumesh
Basel Kurian
Maadhava Subramanian

1. Detailed Application:

For our project we decided to create a web app that aggregates public events in/around ASU from multiple sources (yelp, public records, social media), normalizes them, deduplicates overlaps, and lets people discover events via filtered search. We would use free data sources that give information about events that are upcoming, happening, or that have passed. Our key users and interactions would be visitors that would be able to browse upcoming events; filter by date range, category, location, search by keyword, and access the original source page (if applicable). We would also have organizers who could also post directly onto the website. They would be able to post events, with a description, title, and image. We would also have admins that can approve/ban certain events and resolve any discrepancies.
Core features of ours would be:

- Ingestion: to ingest new/updated/cancelled events.
- Deduplication: to match events by title, time and venue.
- Search and filter: full text search on titles, descriptions, and organizations.
- Event detail page: URL per event

2. ER Diagram


USER
• id : uuid «PK»
email : text
name : text
created_at : timestamptz
saves
SAVE
• id : uuid «PK»
user_id : uuid «FK»
event_id : uuid «FK»
saved_at : timestamptz
saved_by_users
ORGANIZATION
• id : uuid «PK»
name : text
website : text
contact_email : text
type : text
created_at : timestamptz
hosts
EVENT
• id : uuid «PK»
title : text
description : text
start_time : timestamptz
end_time : timestamptz
timezone : text
all_day : boolean
status : text
event_url : text
image_url : text
price_min : money
price_max : money
currency : text
venue_id : uuid «FK»
organization_id : uuid «FK»
created_at : timestamptz
updated_at : timestamptz
search_vector : tsvector
derived_from
EVENT_SOURCE
• id : uuid «PK»
event_id : uuid «FK»
external_uid : text
source_url : text
fetched_at : timestamptz
raw_hash : text
VENUE
• id : uuid «PK»
name : text
address : text
city : text
state : text
postal_code : text
country : text
latitude : float
longitude : float
geom : geometry
location
categorized_as
EVENT_CATEGORY
• id : uuid «PK»
event_id : uuid «FK»
category_id : uuid «FK»
CATEGORY
• id : uuid «PK»
name : text
slug : text
contains

3. Implementation Plan

We'll power the application with PostgreSQL, using its spatial extension (PostGIS) so we can store and search by event locations (latitude/longitude) and do things like "show me events near me." Our database design will follow the entities in our ER diagram: events, venues, feeds, organizers, categories, etc. We'll make sure things that need fast searching (like looking up events by keyword or filtering by time and place) are indexed properly so the app feels responsive.

To get event data into our system, we'll build a small service that runs regularly (e.g. once every hour or day). This service will check all of our event feed sources (such as public calendar feeds in ICS format, RSS/JSON from department/event sites), pull in the latest events, update ones that changed, and remove or mark cancelled ones. We'll also take care to figure out if two events are really the same (duplicate) by comparing things like title, venue, and time. For any venue whose exact coordinates we don't already know, we'll geocode its address (i.e. turn the address into a map location) so events can be shown on a map and distance filters can work.

For the backend of the web app, we'll build APIs that let the frontend ask for lists of events (with filters like date range, category, or location), get detailed info for a single event, manage categories/venues, and handle "saved events" by users. We'll also provide the option for users to subscribe to a custom calendar (e.g. ICS feed) based on filters they choose (for instance, only "lecture" events from certain departments).

https://asu.zoom.us/rec/share/hCvyfVNDuKMmUuMaGBGq4c6u7Bdq9jHv1NdutWITf4E9BI46uyhbOUK3TA5Ozq1W.sX43vmPRuBWVTdyq
Passcode: 47v8=ckb