CS353 DATABASE SYSTEMS

2018-2019 SPRING SEMESTER

SCENEZ

PROJECT PROPOSAL

GROUP 19

Dawood Muzammil **MALIK**

Masna **AHMED**

Muhammad Umair **AHMED**

Talha **ZEESHAN**

Contents

1.Introduction

This project proposal aims to provide a brief overview of the event-hosting web service “Scenez”. After a brief introduction, a description highlighting the main functionality of the service will be included, such as interface and features, followed by the functional and non-functional requirements, database limitations and a detailed E-R Diagram.

2.A Brief Overview

The service will initially require a user to create an account. This will require the user to give enter his full name, age, occupation, interests, a username that will be viewable in his interactions as well as upload an optional picture. Once an account is setup, the user will be directed to a page where he will be able to view his feed.

A user’s feed consists of updates from a number of features such as events, friends, notifications etc. Users will be able to send requests to other users which upon acceptance result in the two users becoming *friends*. Users hold the ability to send and view messages to their friends as well as invite them to join groups and events.

Events form an integral part of the service. Users have the ability to create or host custom events to which they can invite friends as well as group members. Once a user creates an event, a notification is sent to his friends based on the privacy option of the event which can be either “private”, “public”, or “invite-only”. A notification is only sent for the last two options. Hosts can set the event’s venue as well as date and time. Users can also view public events relevant to their interests and location in their feed and can provide feedback on the event along with giving it a rating. Events can also be associated with certain tags which allow for better content filtering when users search for events based on their interest. Users can use these tags to look for events they are interested in.

Users have the ability to post text, images and videos on events which can be viewed by users relevant to the event. Such users can *upvote* and *downvote* such events based on their usefulness. Users can further comment on such posts.

3.Why use a database system?

Scenez allows its users to locate, host or take part in events and gatherings that align with their interests. Such an application mandates the use of an organized data system from which data can be efficiently fetched so that the user can easily and quickly find events or meet up spots with individuals or groups that match with their interests. The use of a database system also allows for:

3.1. Sharing of Data and Multi-User System

A Database system allows for the simultaneous access of multiple users. That is, it allows for many users to access the database at the same time. Due to this feature, a single user in Scenez can find out information about all events that are created on the platform.

3.2. Effective Data Integration

The use of a database system allows us to better keep track of system structure and make it easier for us to enhance the application. It also allows us to keep track of the various relationships between operations in the application structure.

3.3. Support for Multiple Views of Data

Database systems allows for applications like Scenez to add support for multiple views of the data. That is, it allows multiple users of the application to have different views based on their interest and choices. In an application like Scenez, individuals with interest in Meetups related to sports will have a different display compared to individuals whose interest

lie in painting. Hence each user is only able to view a subset of the entire data.

4.Functional Requirements

4.1. Users

- Users will be able to create their personal accounts.
- Users will be able to set their own profile pictures and edit their account description.
- Users will be able to add/remove other users as *friends*.
- Users will be able to create events.
    - Creator of events will be knows as *hosts* for that specific event.
- Users will be able to invite their friends to specific events.
- Users will be able to create *groups* (see section 4.6).
- Users will able to chat with their friends.
- Users will be able to rate and give feedback to the events they attended.
- Users will be able to search people and events.
- Users will be able to respond to invites by accepting or rejecting them.
- Users will be able to follow hashtags.
- Users will be able to mark themselves as *going* or *not going*.
- Users will be able to report fraudulent events as well as their hosts.

4.2. Events

- Each event will have a title and a description.
- Each event will have a category (e.g. Finance, Technology, etc.)
- Each event can have tags. For example, for an event related to database programs, the tags will be *databse, sql, mongo* and etc.
- Each event will have a location specified.

- Each event will have a privacy setting (Public, Invite-only, Private).
    - Public: Open to everyone. Anyone can share it and invite their friends.
    - Invite-only: Only open to those who have been invited by the event host. Others can see the event title but not the details.
    - Private: Can be seen as a *secret event*. Only the ones invited by the host are aware of this type of event.
- The host will be able to specify a time slot and the location of the event.
- The host will be able to post posters, video promos on the event page.
- The host as well as interested users will be able to post on the event page.
- The host as well as interested users will be able to comment on posts made on the event page.
- Each event will have a list of people interested in attending the event.
- The host will be able cancel the event.
- The host can specify a dress code for the event.

4.3. Hashtags

- A number tags can be associated with a specific event.
- Upon the choice of users, events can be searched using tags.

4.4. Chat

- Users will able to chat with their friends.
- Users will be able to view past chats.
- Users will be able to share events via chat.

4.5. Friends

- Users can send friend requests to other users.
- Users will be able to accept/reject friend requests.
- Users will be able view their friends' profiles.
- Users will be able to view their friends' past hosted and attended events.

4.6. Groups

- Users will be able to add all their friends in a group.
- Users will be able to post in the groups.
- User who create a group will be regarded as the *group moderators* of the group.
- A moderator can add other users as admins.
- Only the moderators will be able to delete a group.

5.Non-Functional Requirements

5.1. Reliability

- System should be running 24/7 all the year round except for during scheduled maintenance.
- System should be able execute given instructions without failing.
- During regular backups and maintenance, data should not be lost.

5.2. Scalability

- The application should support a large number of concurrent users without compromising performance and without crashing.

5.3. Performance

- The system should have a maximum response time of 500 ms.
- The application should be able to support at least 1000 concurrent users without crashing or lagging.

5.4. Usability

- Simple and easy to learn user interface
- The user must not need additional information and training when using the application
- 95% of the users must be able to comfortably use the features of the application after initial interaction

5.5. Capacity

- The database should be able to hand at least 2000 requests per second.
- The database should be easily scaled up if required without incurring data loss or crashes.

6.Limitations

- Users will not be able to send friend requests to themselves
- Users will not be able to send messages or send invite links to themselves
- Hosts will not be able to provide feedback or rate their created events
- The website allows a “moderator” feature for group admins and event hosts:
    - A moderator can delete content that is deemed inappropriate for the audience of the group
    - A moderator can remove members of a group or individuals interested in an event if he considers their behavior improper

- All posts posted to a group or event feed must be approved by the moderator
- A group/event can have only a single moderator

- Due to space-considerations users will only be able to view the last 1000 messages sent to a particular friend and vice-versa.

7.Entity Relationship Diagram

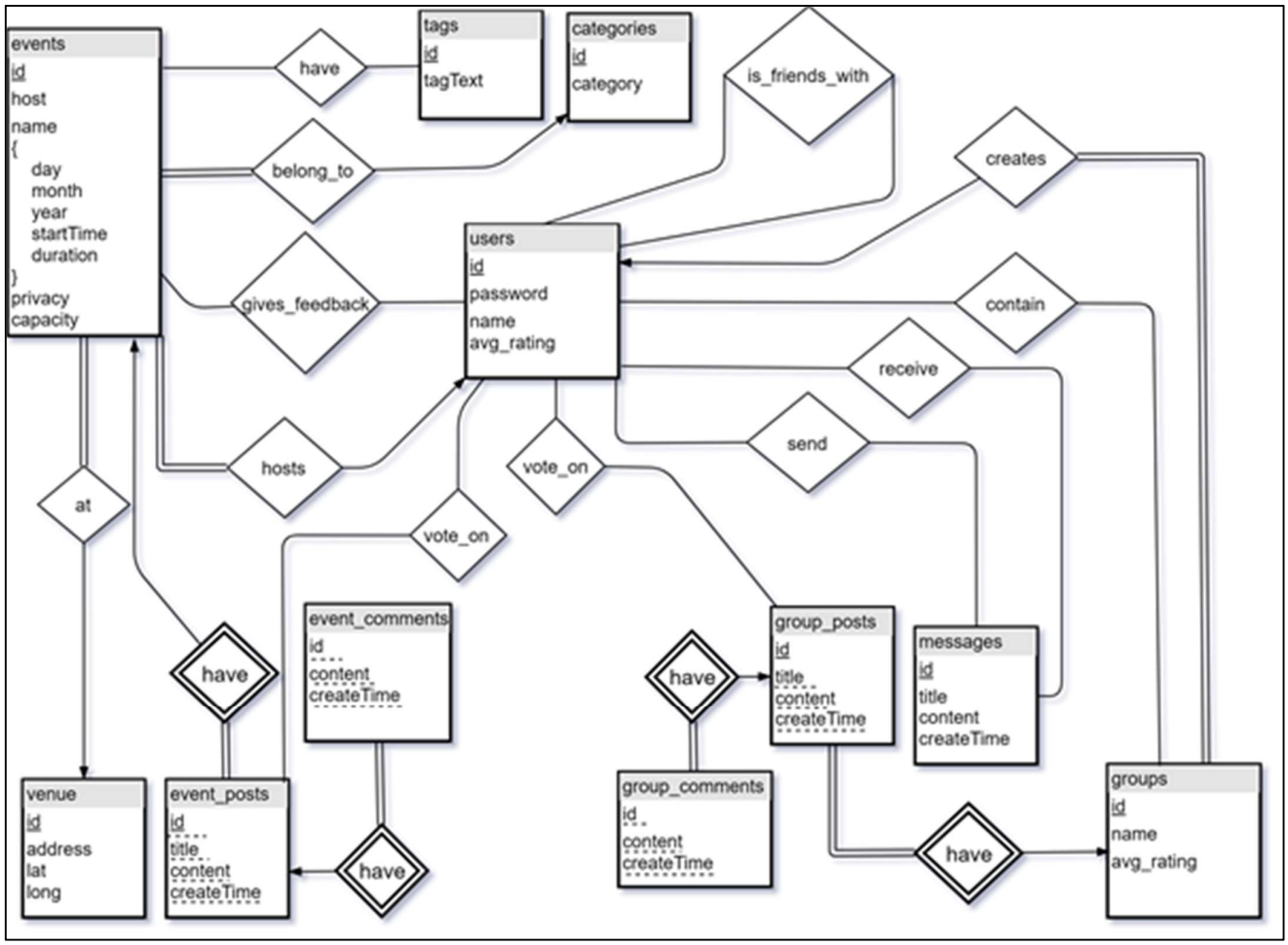

Figure 1: Entity Relationship Model

8.References

- Diagrams made using draw.io (https://www.draw.io/)
- Website link to our documentation about Scenez: https://github.com/ahmed-umair/scenez