

Full Design

Team: Ham

Overview

Author: Victor Cheng

Brief description:

- ADT (Asian Dance Team) and DT (Dance Troupe) are two of the biggest student groups at MIT. Each semester, hundreds of students audition for these groups, and exec members spend hours manually placing people into different dances. Furthermore, during the semester, each choreographer communicates with their dances differently, some using FB groups, others using email chains. Especially for dancers who are in multiple dances, this can get confusing and important announcements may get lost. Our proposed project attempts to solve both these issues, by providing a platform that will have a more efficient, automated rostering system as well as an announcement portal for more consistent communication between members.

Key Purposes:

- Display important information to student group members
 - After the sign up phase, dancers will be able to view announcements from the student groups and teams they are a part of
 - The announcements will by default be displayed in time order (most recent to least recent) and users can choose to filter by team or by type (if the announcement contains a video, etc)
 - Announcements will be separated to group and team announcements so as to not let group announcements spam the team announcement feed
- Automatically match new users to teams based on multiple parameters
 - New members can sign up with their preferences, which includes the teams they want to join, in order of preference, and the new members are labeled with their tryout score
 - Teams will be filled up based on the user's preference as well as their tryout score

Deficiencies in existing solutions

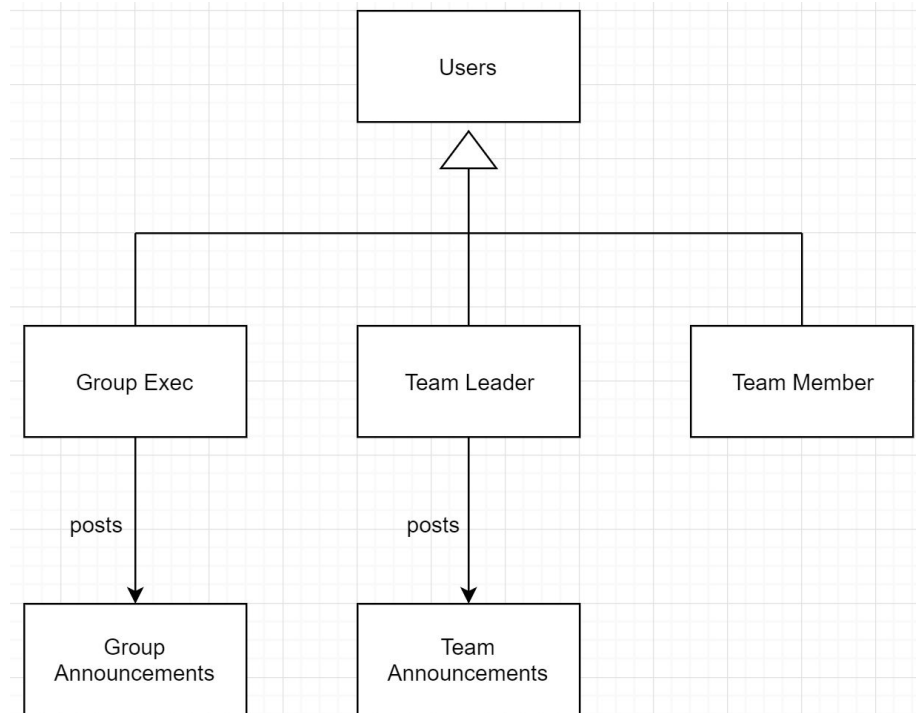
- Facebook Messenger
 - Very spammy in nature and important information can get easily lost in all the messages
- Gmail
 - Message threads causes important emails to get lost in long conversations
- Slack
 - Does not have an automatic rostering solution

Conceptual Design

Author: Victor Cheng

Announcements

- Purpose: Display important information to student group members
- Structure:



- Textual Constraint
 - Group exec, team leader, and team member subsets are mutually exclusive
- Behavior:

User actions:

- postTeamAnnouncement(team: Team, announcement: String, user: User)
 - Requires
 - user is a team leader
 - Effects
 - Posts announcement to the team's announcement page
- postGroupAnnouncement(group: Group, announcement: String, user: User)
 - Requires
 - user is part of group exec
 - Effects

- Posts announcement to the group's announcement page

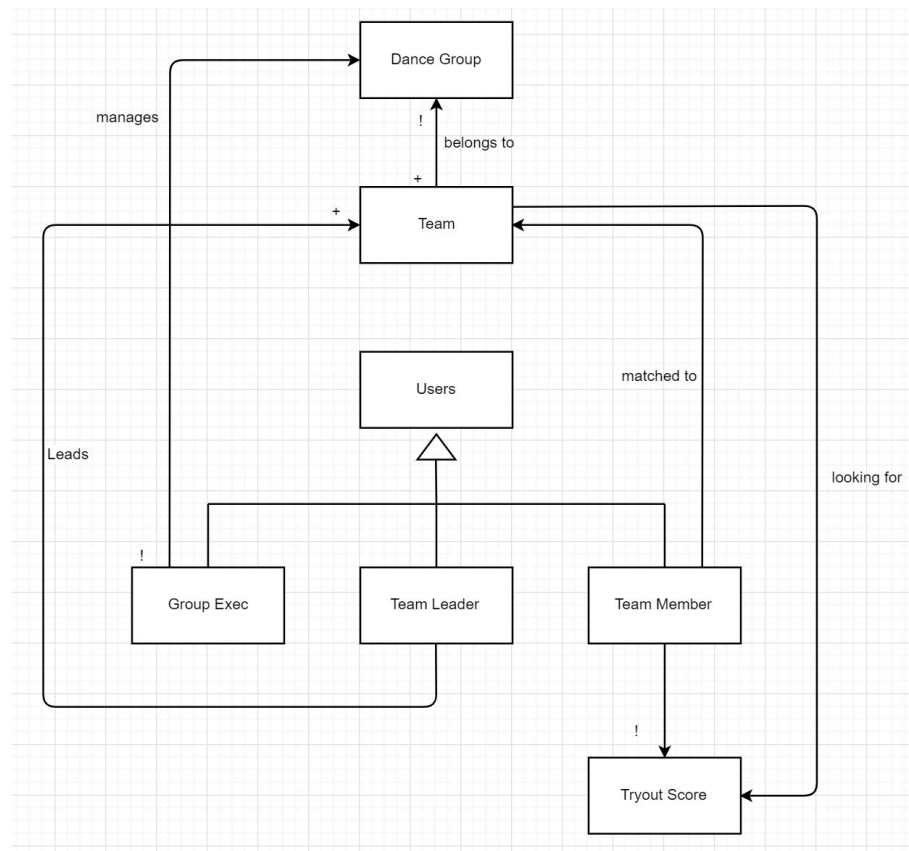
System actions:

- displayTeamAnnouncement(team: Team, announcement: String)
 - Effects
 - Display announcement to team's announcement page
- displayGroupAnnouncement(group: Group, announcement: String)
 - Effects
 - Display announcement to group's announcement page
- Operation Principle:

Users that belong to group exec can post announcements to the group's announcement page. Users that are a team leader can post announcements to the team's announcement page. All team members can view the announcements on both the group and team announcement page.

Rostering

- Purpose: Automatically match new users to teams based on multiple parameters
- Structure:



- Textual Constraint
 - Group exec, team leader, and team member subsets are mutually exclusive

- Behavior:

- User actions:*

- signUp(user: User, group: Group)
 - Requires
 - user has not already signed up
 - Effects
 - Saves the user's preferences for teams they want to join

- System actions:*

- match(user: User, userPrefs: UserPrefs)
 - Requires:
 - user is not on a team
 - Effects
 - Utilizes matching algorithm to place user on team using userPrefs

- Operation Principle:

User can signUp for a group if they have not previously signed up. User will enter in preferences for teams they want to be on. After all users have signed up, the system will use a matching algorithm to optimally assign each user to a team based on the user's preferences.

Data Model

Author: Lesian


```
CREATE TABLE users (  
    id int primary key,  
    name varchar(20) unique,  
    password varchar(30)  
)
```

```
CREATE TABLE groups(  
    id int primary key,  
    name varchar(20) unique,  
)
```

```
CREATE TABLE teams(  
    id int primary,  
    name varchar(20),  
    teamGroup varchar(20) references groups(name),  
    int maxCapacity  
)
```

```
CREATE TABLE groupAdmins(  
    admin varchar(20) references users(name),  
    group varchar(20) references groups(name),  
    Primary key (admin,group)  
)
```

```
CREATE TABLE teamLeads(  
    lead varchar(20) references users(name),  
    team varchar(20) references team(name),  
    primary key (lead, team)  
)
```

```
CREATE TABLE teamMembers(  
    member varchar(20) references users(name),  
    team varchar(20) references teams(name),  
    joinDate datetime  
)
```

```
CREATE TABLE teamAnnouncements(  
    team varchar(20) references team(name),  
    announcement text,  
    announcer references teamLeads(lead)  
)
```

```
CREATE TABLE groupAnnouncements(  
    group varchar(20) references group(name),
```

```
announcement text,  
announcer references groupAdmins(admin)  
)
```

```
CREATE TABLE userTeamRanking(  
    user varchar(20) references users(name),  
    team varchar(20) references team(name),  
    int teamRankByUser,  
    int userRankByTeam,  
    primary key (user, team)  
)
```

Security Concerns

Author: Faaya Fulas

- Key security requirements:
 - The system will have a password protected login page to make sure that users identify and authenticate themselves before accessing restricted resources.
 - Some system features will be visible only to authorized users with special privileges (Group Execs and Team Leads). For example, the 'Create Teams' feature will only be visible to a user with a Group Exec privilege.
- Mitigations to protect the system against standard web attacks:
 - User input supplied via forms will be sanitized to prevent XSS and SQL injection attacks.
 - All forms will include CSRF tokens to prevent CSRF attacks.
 - The system will only store hashed passwords to prevent unauthorized logins even if the 'users' table in the database is compromised.
 - Data integrity will be maintained across tables in the database via appropriate foreign key constraints.

Wireframes

Author: Yolanda Zhou

Users initially start on the sign up / sign in page (presumably after a generic home page with information about the app)

UWU

Sign In

Sign Up

Sign In

Username

Password

Sign Up

Upon signing up, users have the option to join or create a group

GROUPS YOU'RE IN

You aren't in any groups! Click here to [create](#) or [join](#) one.

ANNOUNCEMENTS

No announcements to display.

The rough page for creating a student group would look like this:

CREATE A STUDENT GROUP

Name:

Type: ▼

Create Teams:

+

Sign Up: ▼

Create

After creating the group, there will be a page to manage it (create new teams, manage the team leaders, manage the members of the group, invite team leaders, change the status of the student group sign up phase).

If the user chooses to join a student group, they will see a screen like this:

SEARCH FOR A STUDENT GROUP

dance



RESULTS

MIT Asian Dance Team

Teams: 21

Members: 184

MIT Dancetroupe

Teams: 20

Members: 165

Join

After joining the student group, if the student group is in its sign up phase, the user will be able to pref to pref dances.

MIT Asian Dance Team

TEAM PREFERENCES

1 Rollercoaster

2 Empathy

3 My Yacht

Submit

Similarly, team leaders can pref members.

MIT Asian Dance Team

MEMBER PREFERENCES

Rank your top 30 members

1 Yolanda Zhou

2 Victor Cheng

3 Faaya Abate

Search for a member

Submit

After these preferences are done, the admin of the student group will run stable matching or some other kind of matching process from students to groups. Then, the app functions more as an announcements app to display relevant information to users.

This is what a typical member's view might look like:

TEAMS YOU'RE IN

Empathy

My Yacht

Rollercoaster

ANNOUNCEMENTS

FILTER BY TEAM

- ☒ Empathy
- ☒ Rollercoaster
- ☒ My Yacht

Uncheck all

FILTER BY TAG

- ☒ Important
- ☒ Video
- ☒ Formations
- ☒ Master Post
- ☒ Extras

Uncheck all

HIDE FILTERS

Posted 1 hour ago by Charleen in Empathy

REHEARSAL MOVED FROM MACGREGOR TO MCCORMICK

Important

Posted 1 hour ago by Sophia in Rollercoaster

Reminder that we have rehearsal tomorrow!



You've seen all new announcements

Posted 1 day ago by Ara in My Yacht

Here's the rehearsal video from today



Video

Posted 2 days ago by Ara in My Yacht

Here's the rehearsal video from yesterday



An team leader will get a similar page but with the option to view the teams they manage.

TEAMS YOU MANAGE

Rollercoaster

TEAMS YOU'RE IN

Empathy

My Yacht

ALL ANNOUNCEMENTS

FILTER BY TEAM

- ☒ Empathy
- ☒ Rollercoaster
- ☒ My Yacht

Uncheck all

FILTER BY TAG

- ☒ Important
- ☒ Video
- ☒ Formations
- ☒ Master Post
- ☒ Extras

Uncheck all

HIDE FILTERS

Posted 1 hour ago by Charleen in Empathy

REHEARSAL MOVED FROM MACGREGOR TO MCCORMICK

Important

Posted 1 hour ago by Sophia in Rollercoaster

Reminder that we have rehearsal tomorrow!



You've seen all new announcements

Posted 1 day ago by Ara in My Yacht

Here's the rehearsal video from today



Video

Posted 2 days ago by Ara in My Yacht

Here's the rehearsal video from yesterday



Lastly, the team leader can go on their team page to post announcements or access other settings such as the team roster to view or change

UWU

Dashboard

Sophia ▾

Rollercoaster

SETTINGS / MANAGE

POST AN ANNOUNCEMENT

Enter announcement...

Add Video

Add Tags

Post

ANNOUNCEMENTS

FILTER BY TAG

☒ Important

☒ Master Post

☒ Video

☒ Extras

☒ Formations

Uncheck all

HIDE FILTERS

✓

You've seen all new announcements

Posted 1 hour ago by Sophia in Rollercoaster

Reminder that we have rehearsal tomorrow!

Posted 1 week ago by Sophia in Rollercoaster

Here's the rehearsal video from today

11/3 Rehearsal - a

Design Commentary

Author: Faaya Fulas

- During the conceptual design phase for the Announcements concept, we experimented with the idea of having a single set for both group and team announcements. However, we realized that we will need to differentiate between announcements targeted to the entire student group and announcements relevant to specific teams to avoid spamming users with team announcements irrelevant to them . Ultimately, we decided to have seperate sets for each announcement types.
- During the conceptual design phase for the Rostering concept, we had extensive discussions about how the system would automatically match teams with unassigned users. We initially came up with the idea of having a 'Qualifications' set that would contain a list of attributes the user 'has' and the team 'requires'. This proved to be problematic during the schema design phase since the list of attributes that the Qualifications table should contain would be different depending on the type of Dance Group.

Ultimately, we came up with the idea of Ranking - each user will be labeled with a single score assigned to them by the Team Leads based on any number of criteria they deem relevant. This design choice not only resolved the issue of having inconsistent attributes in the Qualifications table , it also made the matching process more straightforward since users can be effectively sorted and filtered based on their ranking scores.