

Group Report

Application: College Event Website

COP4710-19 FALL 2019

Group #52

Jason Dhillon

Table of Contents

Project Description:	1
GUI:	2
ER-Model	8
Relational Data Model Script	9
SQL Examples/Results	16
Constraint Enforcement	21
Advanced Features	24
Conclusion	27

(page numbers are clickable)

Project Description:

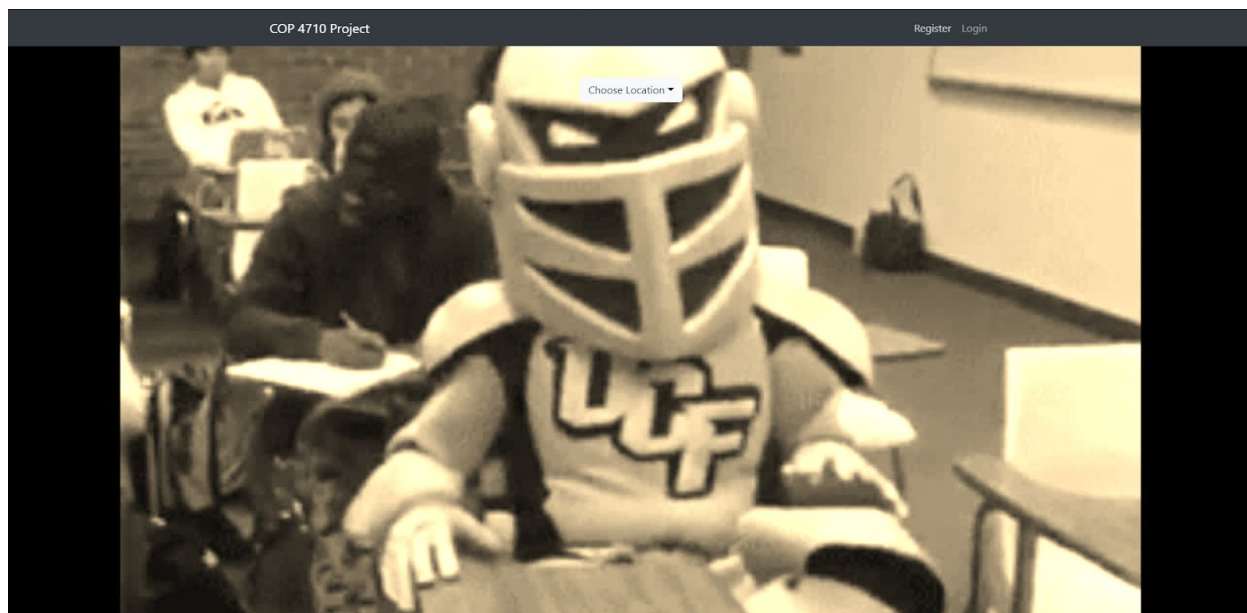
Create a web application which displays event information for student created RSO's or Registered Student Organizations. There should be three different tiers of users: students, admins, and superadmins. Students should be able to join RSO's and create them, in addition to making comments and rating the events. Admins are able to create events for their RSO's. Superadmins are able to create new universities as well as approve or deny public events from students.

Student RSO's require at least 5 students before becoming active, in which case events can then be made for the organization and displayed on the site. Events may not overlap if they occur on the same day, same time, and same location as another event. Events should include a name, date, time, description, and whether the event is public, private, or club members only.

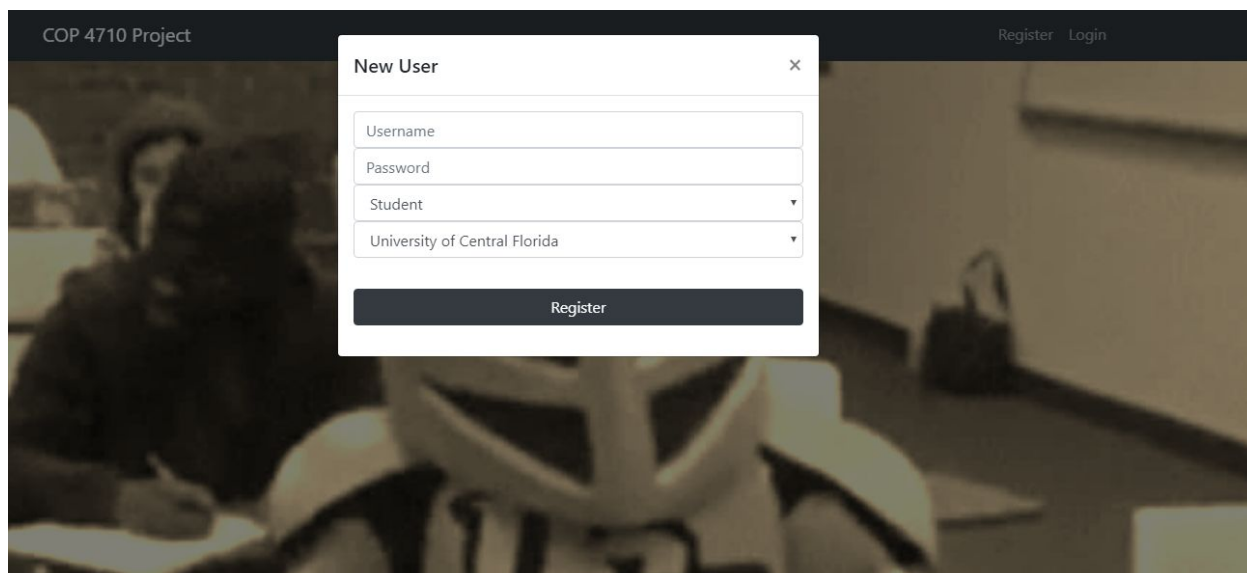
The application should be a full stack, utilizing a frontend, backend, and a database (MySQL).

GUI:

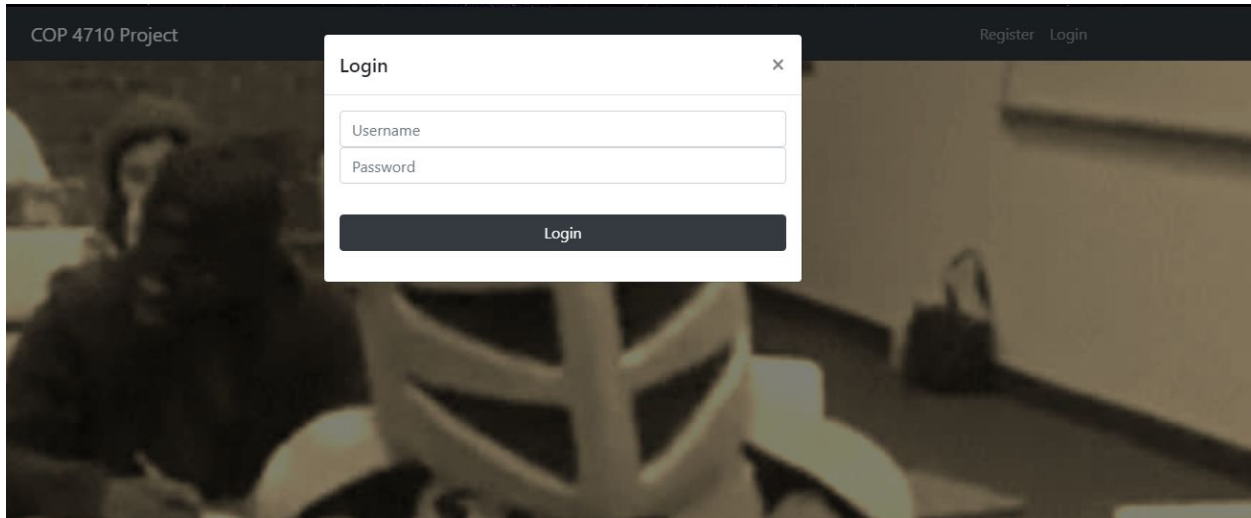
Not Logged in -> base site



Register modal



Login modal



COP 4710 Project

Register Login

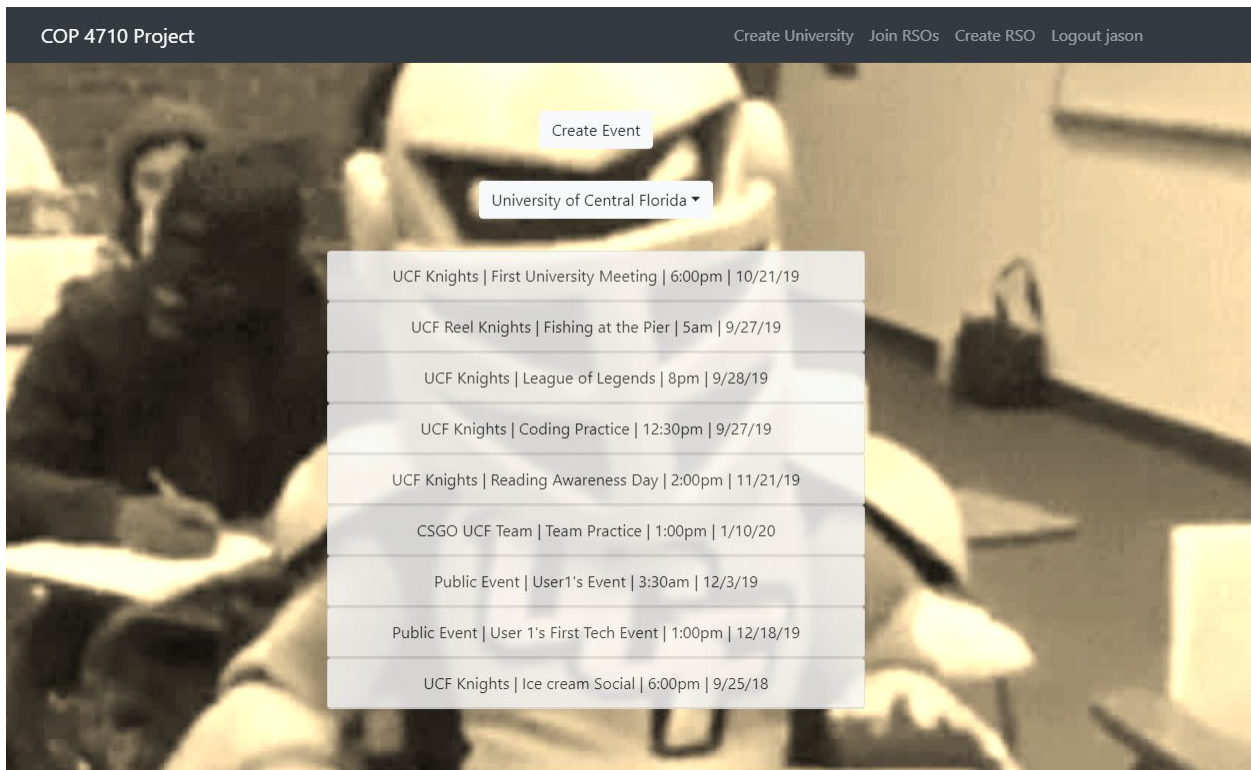
Login ×

Username

Password

Login

Logged in -> view events



COP 4710 Project

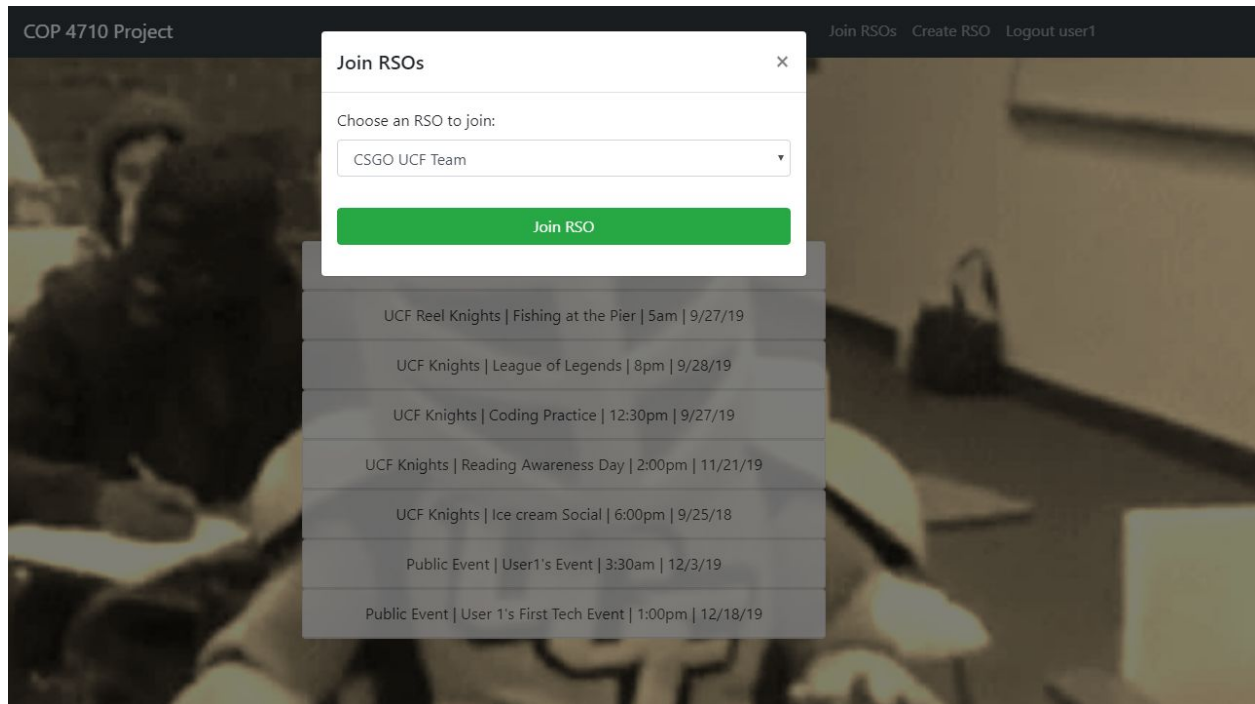
Create University Join RSOs Create RSO Logout jason

Create Event

University of Central Florida ▾

UCF Knights First University Meeting 6:00pm 10/21/19
UCF Reel Knights Fishing at the Pier 5am 9/27/19
UCF Knights League of Legends 8pm 9/28/19
UCF Knights Coding Practice 12:30pm 9/27/19
UCF Knights Reading Awareness Day 2:00pm 11/21/19
CSGO UCF Team Team Practice 1:00pm 1/10/20
Public Event User1's Event 3:30am 12/3/19
Public Event User 1's First Tech Event 1:00pm 12/18/19
UCF Knights Ice cream Social 6:00pm 9/25/18

Join RSO modal



Create RSO modal

COP 4710 Project

Create University | Join RSOs | Create RSO | Logout jason

Create New RSO

Create RSO

UCF Knights First University Meeting 6:00pm 10/21/19
UCF Reel Knights Fishing at the Pier 5am 9/27/19
UCF Knights League of Legends 8pm 9/28/19
UCF Knights Coding Practice 12:30pm 9/27/19
UCF Knights Reading Awareness Day 2:00pm 11/21/19
CSGO UCF Team Team Practice 1:00pm 1/10/20
Public Event User1's Event 3:30am 12/3/19
Public Event User 1's First Tech Event 1:00pm 12/18/19
UCF Knights Ice cream Social 6:00pm 9/25/18
(Requires Approval Public Event Tailgate Party 5:00pm 11/29/19

Create event

COP 4710 Project [Create University](#) [Join RSOs](#) [Create RSO](#) [Logout jason](#)

New Event

Choose RSO to make the event for:

UCF Knights

Event name

Meeting

Description

Time

PM

Date

Location

Phone

Email

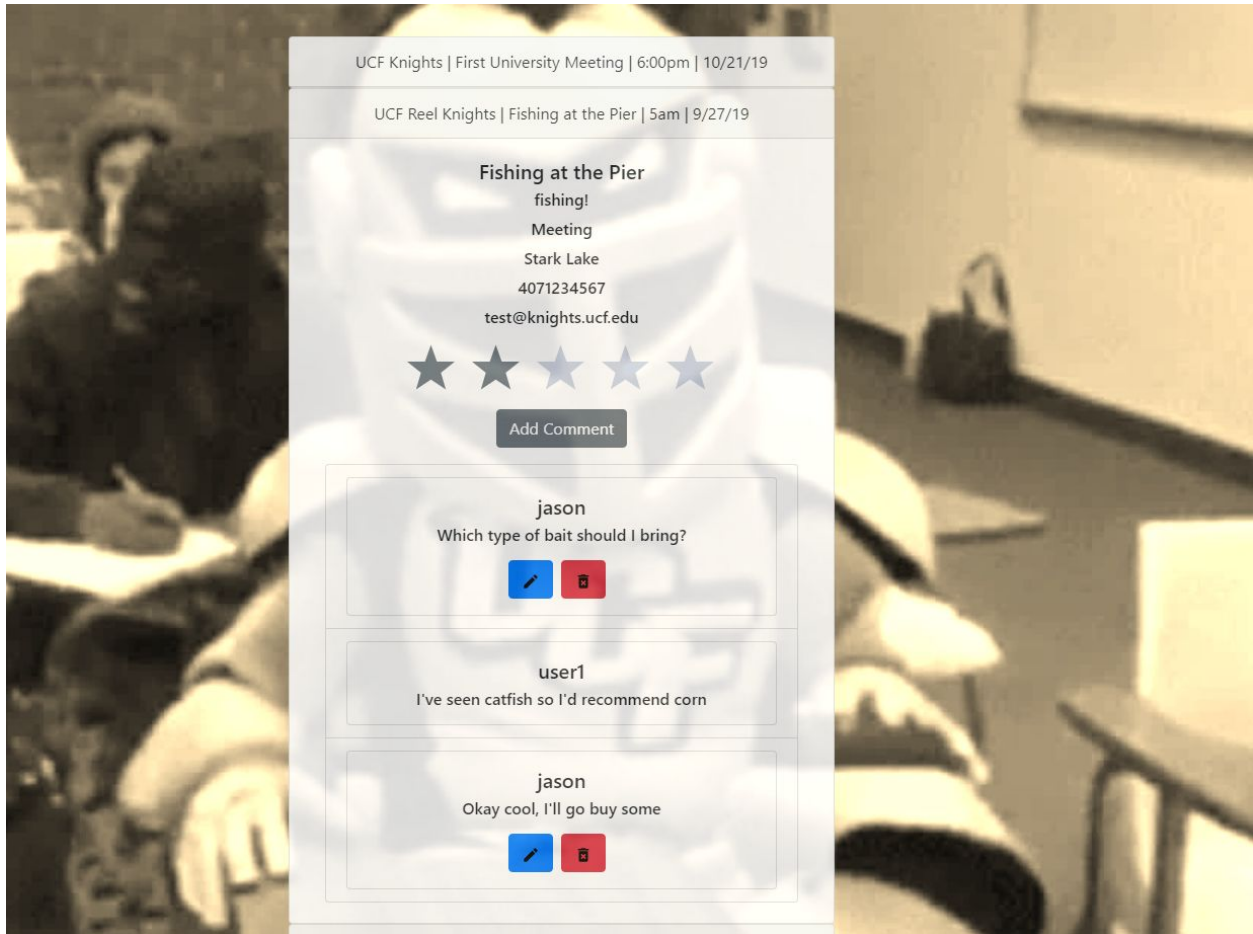
Club Members only

Create Event

UCF Knights | Ice cream Social | 6:00pm | 9/25/18

(Requires Approval | Public Event | Tailgate Party | 5:00pm | 11/29/19)

View event details/comments



UCF Knights | First University Meeting | 6:00pm | 10/21/19

UCF Reel Knights | Fishing at the Pier | 5am | 9/27/19

Fishing at the Pier
fishing!
Meeting
Stark Lake
4071234567
test@knights.ucf.edu

★ ★ ★ ★ ★

Add Comment

jason
Which type of bait should I bring?

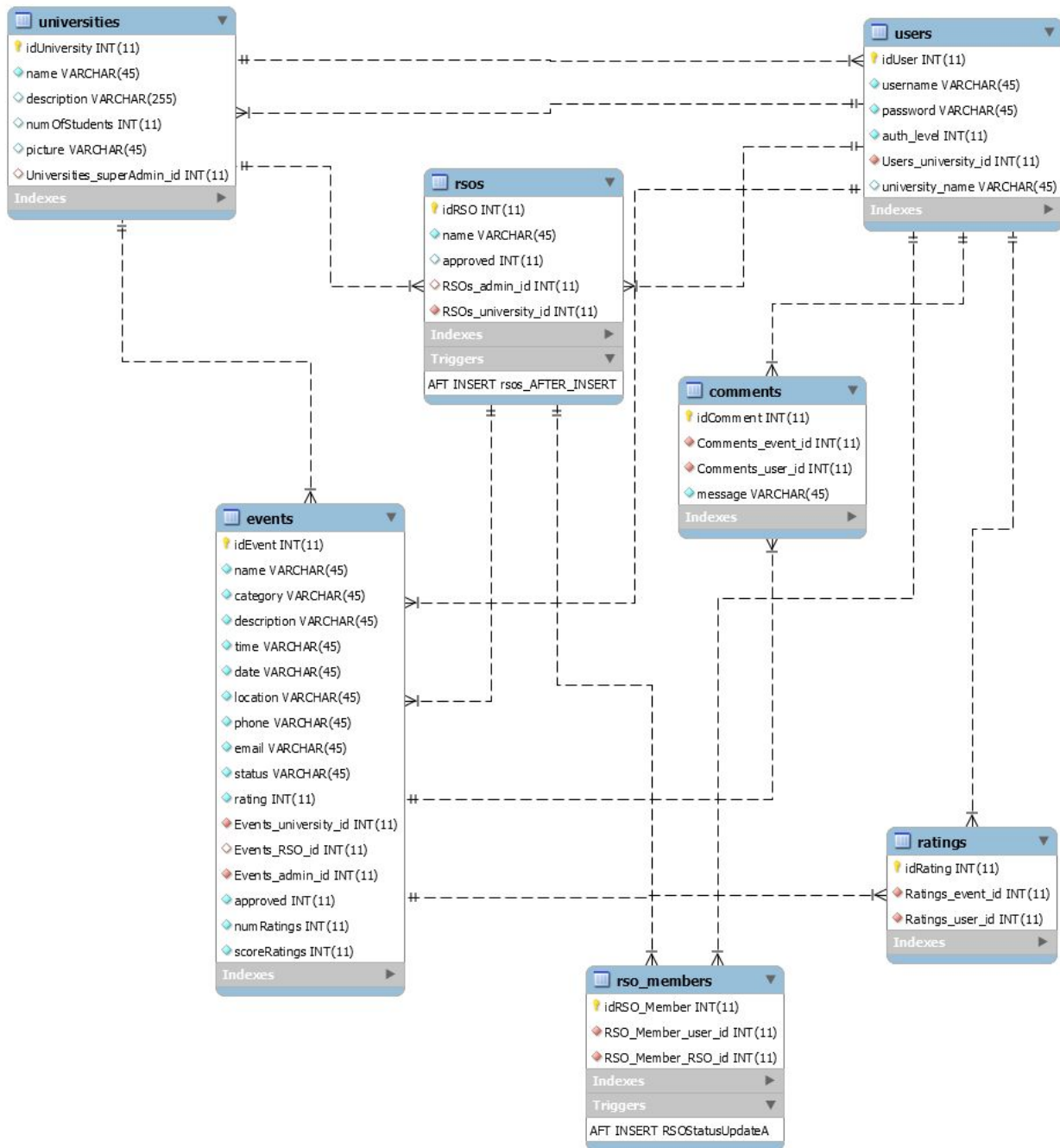
✎ 📷

user1
I've seen catfish so I'd recommend corn

jason
Okay cool, I'll go buy some

✎ 📷

ER-Model



Relational Data Model Script

```
-- MySQL Script generated by MySQL Workbench
-- Wed Nov 20 14:57:49 2019
-- Model: New Model   Version: 1.0
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_
ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----
-- Schema mydb
-- -----
-- -----
-- Schema heroku_8a8e9bdedc4afce
-- -----
```

```
-- -----
-- Schema heroku_8a8e9bdedc4afce
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `heroku_8a8e9bdedc4afce` DEFAULT CHARACTER
SET utf8mb4 COLLATE utf8mb4_0900_ai_ci ;
USE `heroku_8a8e9bdedc4afce` ;
```

```
-- -----
-- Table `heroku_8a8e9bdedc4afce`.`universities`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `heroku_8a8e9bdedc4afce`.`universities` (
  `idUniversity` INT(11) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `description` VARCHAR(255) NULL DEFAULT NULL,
  `numOfStudents` INT(11) NULL DEFAULT NULL,
```

```

`picture` VARCHAR(45) NULL DEFAULT NULL,
`Universities_superAdmin_id` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`idUniversity`),
UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,
INDEX `superAdmin_id_idx` (`Universities_superAdmin_id` ASC) VISIBLE,
CONSTRAINT `Universities_superAdmin_id`
  FOREIGN KEY (`Universities_superAdmin_id`)
  REFERENCES `heroku_8a8e9bdedc4afce`.`users` (`idUser`))
ENGINE = InnoDB
AUTO_INCREMENT = 26
DEFAULT CHARACTER SET = utf8;

```

```

-- -----
-- Table `heroku_8a8e9bdedc4afce`.`users`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `heroku_8a8e9bdedc4afce`.`users` (
  `idUser` INT(11) NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(45) NOT NULL,
  `password` VARCHAR(45) NOT NULL,
  `auth_level` INT(11) NOT NULL DEFAULT '1',
  `Users_university_id` INT(11) NOT NULL,
  `university_name` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idUser`),
  UNIQUE INDEX `username_UNIQUE` (`username` ASC) VISIBLE,
  INDEX `university_id_idx` (`Users_university_id` ASC) VISIBLE,
  CONSTRAINT `Users_university_id`
    FOREIGN KEY (`Users_university_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`universities` (`idUniversity`))
ENGINE = InnoDB
AUTO_INCREMENT = 394
DEFAULT CHARACTER SET = utf8;

```

```

-- -----
-- Table `heroku_8a8e9bdedc4afce`.`rsos`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `heroku_8a8e9bdedc4afce`.`rsos` (
  `idRSO` INT(11) NOT NULL AUTO_INCREMENT,

```

```

`name` VARCHAR(45) NOT NULL,
`approved` INT(11) NULL DEFAULT '0',
`RSOs_admin_id` INT(11) NULL DEFAULT NULL,
`RSOs_university_id` INT(11) NOT NULL,
PRIMARY KEY (`idRSO`),
UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,
INDEX `admin_id_idx` (`RSOs_admin_id` ASC) VISIBLE,
INDEX `university_id_idx` (`RSOs_university_id` ASC) VISIBLE,
CONSTRAINT `RSOs_admin_id`
  FOREIGN KEY (`RSOs_admin_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`users` (`idUser`),
CONSTRAINT `RSOs_university_id`
  FOREIGN KEY (`RSOs_university_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`universities` (`idUniversity`))
ENGINE = InnoDB
AUTO_INCREMENT = 554
DEFAULT CHARACTER SET = utf8;

```

```

-- -----
-- Table `heroku_8a8e9bdedc4afce`.`events`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `heroku_8a8e9bdedc4afce`.`events` (
  `idEvent` INT(11) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `category` VARCHAR(45) NOT NULL DEFAULT 'Meeting',
  `description` VARCHAR(45) NOT NULL,
  `time` VARCHAR(45) NOT NULL,
  `date` VARCHAR(45) NOT NULL,
  `location` VARCHAR(45) NOT NULL,
  `phone` VARCHAR(45) NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `status` VARCHAR(45) NOT NULL DEFAULT 'public',
  `rating` INT(11) NOT NULL DEFAULT '0',
  `Events_university_id` INT(11) NOT NULL,
  `Events_RSO_id` INT(11) NULL DEFAULT NULL,
  `Events_admin_id` INT(11) NOT NULL,
  `approved` INT(11) NOT NULL DEFAULT '0',
  `numRatings` INT(11) NOT NULL DEFAULT '0',

```

```

`scoreRatings` INT(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`idEvent`),
INDEX `university_id_idx` (`Events_university_id` ASC) VISIBLE,
INDEX `RSO_id_idx` (`Events_RSO_id` ASC) VISIBLE,
INDEX `admin_id_idx` (`Events_admin_id` ASC) VISIBLE,
CONSTRAINT `Events_RSO_id`
  FOREIGN KEY (`Events_RSO_id`)
  REFERENCES `heroku_8a8e9bdedc4afce`.`rsos` (`idRSO`),
CONSTRAINT `Events_admin_id`
  FOREIGN KEY (`Events_admin_id`)
  REFERENCES `heroku_8a8e9bdedc4afce`.`users` (`idUser`),
CONSTRAINT `Events_university_id`
  FOREIGN KEY (`Events_university_id`)
  REFERENCES `heroku_8a8e9bdedc4afce`.`universities` (`idUniversity`))
ENGINE = InnoDB
AUTO_INCREMENT = 531
DEFAULT CHARACTER SET = utf8;

```

```

-- -----
-- Table `heroku_8a8e9bdedc4afce`.`comments`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `heroku_8a8e9bdedc4afce`.`comments` (
  `idComment` INT(11) NOT NULL AUTO_INCREMENT,
  `Comments_event_id` INT(11) NOT NULL,
  `Comments_user_id` INT(11) NOT NULL,
  `message` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idComment`),
  INDEX `event_id_idx` (`Comments_event_id` ASC) VISIBLE,
  INDEX `user_id_idx` (`Comments_user_id` ASC) VISIBLE,
  CONSTRAINT `Comments_event_id`
    FOREIGN KEY (`Comments_event_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`events` (`idEvent`),
  CONSTRAINT `Comments_user_id`
    FOREIGN KEY (`Comments_user_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`users` (`idUser`))
ENGINE = InnoDB
AUTO_INCREMENT = 245
DEFAULT CHARACTER SET = utf8;

```

```

-----
-- Table `heroku_8a8e9bdedc4afce`.`ratings`
-----

CREATE TABLE IF NOT EXISTS `heroku_8a8e9bdedc4afce`.`ratings` (
  `idRating` INT(11) NOT NULL,
  `Ratings_event_id` INT(11) NOT NULL,
  `Ratings_user_id` INT(11) NOT NULL,
  PRIMARY KEY (`idRating`),
  INDEX `event_id_idx` (`Ratings_event_id` ASC) VISIBLE,
  INDEX `user_id_idx` (`Ratings_user_id` ASC) VISIBLE,
  CONSTRAINT `Ratings_event_id`
    FOREIGN KEY (`Ratings_event_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`events` (`idEvent`),
  CONSTRAINT `Ratings_user_id`
    FOREIGN KEY (`Ratings_user_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`users` (`idUser`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

-----

-- Table `heroku_8a8e9bdedc4afce`.`rso_members`
-----

CREATE TABLE IF NOT EXISTS `heroku_8a8e9bdedc4afce`.`rso_members` (
  `idRSO_Member` INT(11) NOT NULL AUTO_INCREMENT,
  `RSO_Member_user_id` INT(11) NOT NULL,
  `RSO_Member_RSO_id` INT(11) NOT NULL,
  PRIMARY KEY (`idRSO_Member`),
  INDEX `user_id_idx` (`RSO_Member_user_id` ASC) VISIBLE,
  INDEX `RSO_id_idx` (`RSO_Member_RSO_id` ASC) VISIBLE,
  CONSTRAINT `RSO_Member_RSO_id`
    FOREIGN KEY (`RSO_Member_RSO_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`rsos` (`idRSO`),
  CONSTRAINT `user_id`
    FOREIGN KEY (`RSO_Member_user_id`)
    REFERENCES `heroku_8a8e9bdedc4afce`.`users` (`idUser`))
ENGINE = InnoDB

```

AUTO_INCREMENT = 170

DEFAULT CHARACTER SET = utf8;

USE `heroku_8a8e9bdedc4afce`;

DELIMITER \$\$

USE `heroku_8a8e9bdedc4afce`\$\$

CREATE

DEFINER=`root`@`localhost`

TRIGGER `heroku_8a8e9bdedc4afce`.`rsos_AFTER_INSERT`

AFTER INSERT ON `heroku_8a8e9bdedc4afce`.`rsos`

FOR EACH ROW

BEGIN

INSERT INTO `heroku_8a8e9bdedc4afce`.`rso_members`

(

`RSO_Member_user_id`,

`RSO_Member_RSO_id`)

VALUES

(

NEW.RSOs_admin_id,

NEW.idRSO);

END\$\$

USE `heroku_8a8e9bdedc4afce`\$\$

CREATE

DEFINER=`root`@`localhost`

TRIGGER `heroku_8a8e9bdedc4afce`.`RSOStatusUpdateA`

AFTER INSERT ON `heroku_8a8e9bdedc4afce`.`rso_members`

FOR EACH ROW

BEGIN

IF ((SELECT COUNT(*) FROM rso_members M WHERE M.RSO_Member_RSO_id =
NEW.RSO_Member_RSO_id) > 4)

THEN

UPDATE rsos r

SET approved = 1

WHERE r.idRSO = NEW.RSO_Member_RSO_id;

END IF;

END\$\$

DELIMITER ;

SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

SQL Examples/Results

Insert a new RSO

The screenshot shows a SQL client interface with a query editor and a results grid. The query editor contains two SQL statements:

```
1 • INSERT INTO rsos (name, approved, RSOs_admin_id, RSOs_university_id) VALUES ("Test RSO", 0, 2, 2);
2 • SELECT * FROM heroku_8a8e9bde4cfce.rsos;
```

The results grid displays the output of the second query, showing a list of RSOs. The row for "Test RSO" is highlighted in yellow.

idRSO	name	approved	RSOs_admin_id	RSOs_university_id
2	UCF Knights	1	2	2
22	UCF Reel Knights	1	342	2
32	USF Football	1	292	12
552	CSGO UCF Team	0	2	2
553	Public Event	0	2	2
554	Test RSO	0	2	2
NULL	NULL	NULL	NULL	NULL

Join RSO

SQL File 5* x

Limit to 1000 rows

```

1 • INSERT INTO rso_members (Rso_Member_user_id, Rso_Member_RSO_id) VALUES ( 342 , 554 );
2 • SELECT * FROM rso_members

```

Result Grid

	idRso_Member	Rso_Member_user_id	Rso_Member_RSO_id
▶	2	2	2
	12	342	22
	22	292	32
	122	2	22
	132	322	2
	142	322	22
	159	2	552
	160	383	22
	161	2	553
	162	384	22
	163	384	2
	164	383	2
	165	385	2
	166	390	32
	167	391	32
	168	389	32
	169	393	32
	170	2	554
	171	342	554
*	NULL	NULL	NULL

rso_members 2 x

Create an event

```

1 • INSERT INTO events (name, category, description, time, date, location, phone, email, status, rating, Events_university_id, Events_RSO_id, Events_admin_id, approved)
2 • VALUES ("Test Event", "Meeting", "Test event", "5:00pm", "11/20/19", "Library", "407-000-0000", "test@email.com", "private", 2, 2, 2, 1);
3
4 • SELECT * FROM events;

```

idEvent	name	category	description	time	date	location	phone	email	status	rating	Events_university_id	Events_RSO_id	Events_admin_id	approved	numRatings	scoreRatings
2	First University Meeting	Meeting	First Meeting for all university members!	6:00pm	10/21/19	28.0024° N, 81.2001° W	407-000-0000	test@knights.ucf.edu	public	4	2	2	2	1	10	39
32	South Florida Bulls Football Game	Meeting	Football	4:00pm	11/12/19	USF	417-414-6328	test@bulls.ucf.edu	Public	2	12	32	292	1	4	6
42	South Florida Bulls Football Game 2	Meeting	Football	4:00pm	11/19/19	USF	417-414-6328	test@bulls.ucf.edu	Public	3	12	32	292	1	2	6
52	South Florida Bulls Football Game 3	Meeting	Football	6:00pm	11/26/19	USF	417-414-6328	test@bulls.ucf.edu	Public	1	12	32	292	1	1	1
182	Fishing at the Pier	Meeting	fishing!	5am	9/27/19	Stark Lake	4071234567	test@knights.ucf.edu	2	2	22	342	1	3	7	
272	League of Legends	Meeting	Let's Play League	8pm	9/28/19	165.1	23	21	rso	0	2	2	2	1	0	0
452	Coding Practice	Social	Come practice your coding skills	12:30pm	9/27/19	HEC	4077778888	Guba@knights.ucf.edu	private	2	2	2	2	1	2	4
509	Reading Awareness Day	Guest Speaker	Take a moment and swing by to our event!	2:00pm	11/21/19	Library	407-234-1243	jason@knights.ucf.edu	public	4	2	2	2	1	1	4
510	Team Practice	Meeting	First meeting of the semester!	1:00pm	1/10/20	Library 1st Floor	407-845-8547	jason@knights.ucf.edu	rso	0	2	552	2	1	0	0
519	User Ex Event	Tech Talks	I want to hold a tech talk!	3:30am	12/3/19	HEC 210	407-556-5417	user1@knights.ucf.edu	Public	2	2	553	383	1	2	4
520	User Ex First Tech Event	Tech Talks	I'm discussing new technology!	1:00pm	12/16/19	HEC 211	407-593-4462	user1@knights.ucf.edu	Public	2	2	553	383	1	4	9
523	USF Meet and Greet	Social	Meet new friends!	1:00pm	11/23/19	Library 2nd Floor	407-458-8855	user@bulls.ucf.edu	public	1	12	553	292	1	1	1
524	341231232	Meeting	313132...	321332...	331332	321	2313213	323	Public	0	12	553	389	1	0	0
526	46	Meeting	4654	6545pm	4	65465	4	5645	rso	0	12	553	292	1	0	0
529	Talgate Party	Social	Come support the knights!	5:00pm	11/25/19	Spectrum Stadium	407-111-1234	user1@knights.ucf.edu	Public	0	2	553	383	0	0	0
530	Ice cream Social	Social	Come on out and meet new people!	6:00pm	9/25/18	My house	407-000-0000	jason@knights.ucf.edu	rso	0	2	2	2	1	0	0
531	Test Event	Meeting	Test event	5:00pm	11/20/19	Library	407-000-0000	test@email.com	private	0	2	2	2	1	0	0

Create comment

```

1 • INSERT INTO comments (Comments_event_id, Comments_user_id, message) VALUES ( 531 , 2, "This is a test comment!");
2
3 • SELECT * FROM comments;

```

idComment	Comments_event_id	Comments_user_id	message
92	272	2	Hello !
222	272	342	Test Comment
232	2	342	Test Comment
242	182	2	Which type of bait should I bring?
243	182	383	I've seen catfish so I'd recommend corn
244	182	2	Okay cool, I'll go buy some
245	531	2	This is a test comment!
NULL	NULL	NULL	NULL

Modify comment

The screenshot shows a database management interface with a top toolbar containing icons for file operations, execution, and navigation. The main window displays two SQL statements:

```
1 • UPDATE comments SET message = "Hello!" WHERE idComment = 245;
2
3 • SELECT * FROM comments;
```

Below the SQL editor is a "Result Grid" section. It includes a "Filter Rows" input field and buttons for "Edit", "Export/Import", and "Wrap Cell Content". The grid displays the following data:

	idComment	Comments_event_id	Comments_user_id	message
▶	92	272	2	Hello !
	222	272	342	Test Comment
	232	2	342	Test Comment
	242	182	2	Which type of bait should I bring?
	243	182	383	I've seen catfish so I'd recommend corn
	244	182	2	Okay cool, I'll go buy some
	245	531	2	Hello!
*	NULL	NULL	NULL	NULL

Delete comment

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
1 • DELETE FROM comments WHERE idComment = 245;
2
3 • SELECT * FROM comments;
```

The result grid displays the following data:

	idComment	Comments_event_id	Comments_user_id	message
▶	92	272	2	Hello !
	222	272	342	Test Comment
	232	2	342	Test Comment
	242	182	2	Which type of bait should I bring?
	243	182	383	I've seen catfish so I'd recommend corn
	244	182	2	Okay cool, I'll go buy some
	NULL	NULL	NULL	NULL

Display events for user (public, private, and RSO)

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 • SELECT idEvent, rso.name, rso.idRso, events.name AS eventName, events.approved, Events_university_id, events.status, category, description, time, events.date, location, phone, email, rating, numRatings, scoreRatings, RSO_Member_user_id FROM events
2 INNER JOIN rso_members ON events.Events_RSO_id = rso_members.RSO_member_RSO_id AND RSO_Member_user_id = 2
3
4
5
6
7 UNION SELECT idEvent, rso.name, rso.idRso, events.name AS eventName, events.approved, Events_university_id, events.status, category, description, time, events.date, location, phone, email, rating, numRatings, scoreRatings,
8 RSO_Member_user_id FROM events, rso
9 WHERE events.Events_RSO_id = rso.idRso AND (events.Events_university_id = 2 AND events.approved = 1 AND events.status = "public") GROUP BY idEvent
10
```

The result grid displays the following data:

	idEvent	name	idRso	eventName	approved	Events_university_id	status	category	description	time	date	location	phone	email	rating	numRatings	scoreRatings	RSO_Member_user_id
▶	2	UCF Knights	2	First University Meeting	1	2	public	Meeting	First Meeting for all university members!	6:00pm	10/21/19	28.4024°N, 81.2001°W	407-000-0000	test@knights.mal.edu	4	9	38	2
	182	UCF Knights	22	Fishing at the Pier	1	2	public	Meeting	fishing!	5am	9/27/19	Stark Lake	407-234-9667	test@knights.ucf.edu	2	3	7	2
	272	UCF Knights	2	League of Legends	1	2	rso	Meeting	Let's Play League	8pm	9/28/19	3551	23	21	0	0	0	2
	452	UCF Knights	2	Coding Practice	1	2	private	Social	Come practice your coding skills	12:30pm	9/27/19	HEC	407-7778088	Gabe@knights.ucf.edu	3	1	3	2
	509	UCF Knights	2	Reading Awareness Day	1	2	public	Guest Speaker	Take a moment and swing by to our event!	2:00pm	11/21/19	Library	407-234-1243	jason@knights.ucf.edu	4	1	4	2
	510	CSGO UCF Team	552	Team Practice	1	2	rso	Meeting	First meeting of the semester!	1:00pm	1/10/20	Library 1st Floor	407-845-8547	jason@knights.ucf.edu	0	0	0	2
	519	Public Event	553	User's Event	1	2	Public	Tech Talks	I want to hold a tech talk!	3:00pm	12/9/19	HEC 210	407-555-5417	user1@knights.ucf.edu	3	1	3	2
	520	Public Event	553	User's First Tech Event	1	2	Public	Tech Talks	I'm discussing new technology!	1:00pm	12/18/19	HEC 211	407-593-4482	user1@knights.ucf.edu	0	0	0	2

Constraint Enforcement

Overlapping events:

COP 4710 Project Create University Join RSOs Create RSO Logout jason

New Event

This event time is already taken, please choose a different time of day

Choose RSO to make the event for:

UCF Knights

Ice cream Social

Social

Come on out and meet new people!

6:00

PM

9/25/18

My house

407-000-0000

jason@knights.ucf.edu

Club Members only

Create Event

Admin who is not the admin of a different RSO creating an event:

COP 4710 Project

Join RSOs Create RSO Logout user

New Event

Choose RSO to make the event for:

- Public Event (Pending approval of SuperAdmin) (Unapprove)
- USF Football (Unapproved)
- Public Event (Pending approval of SuperAdmin) (Unapproved)
- Meeting

Description

Time

PM

Date

Location

Phone

Email

Club Members only

Create Event

- Admins can only see the RSO's for which they are an admin for, no way for this error to occur

Student joining an RSO with 4 members (going from unapproved to approved status)

	idRSO	name	approved	RSOs_admin_id	RSOs_university_id
	2	UCF Knights	1	2	2
	22	UCF Reel Knights	1	342	2
	32	USF Football	0	292	12
	552	CSGO UCF Team	0	2	2
▶	553	Public Event	0	2	2
*	NULL	NULL	NULL	NULL	NULL

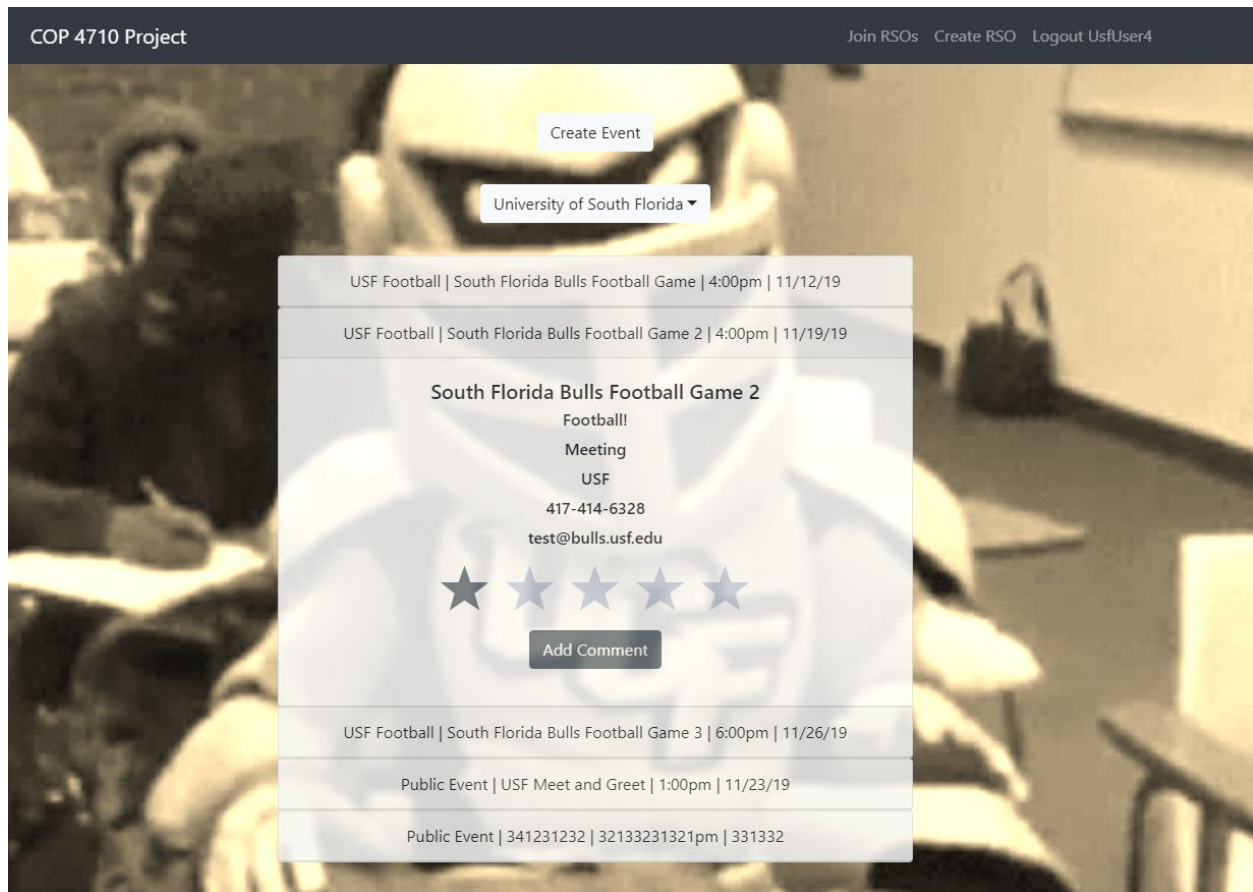
- Note we are adding the student to USF Football #32, unapproved

	idRSO_Member	RSO_Member_RSO_id	RSO_Member_user_id	approved
▶	2	2	2	1
	132	2	322	1
	163	2	384	1
	164	2	383	1
	165	2	385	1
	12	22	342	1
	122	22	2	1
	142	22	322	1
	160	22	383	1
	162	22	384	1
	22	32	292	0
	166	32	390	0
	167	32	391	0
	168	32	389	0
	159	552	2	0
	161	553	2	0

	idRSO_Member	RSO_Member_RSO_id	RSO_Member_user_id	approved
▶	2	2	2	1
	132	2	322	1
	163	2	384	1
	164	2	383	1
	165	2	385	1
	12	22	342	1
	122	22	2	1
	142	22	322	1
	160	22	383	1
	162	22	384	1
	22	32	292	1
	166	32	390	1
	167	32	391	1
	168	32	389	1
	169	32	393	1
	159	552	2	0
	161	553	2	0

Advanced Features

- I have used JSON Web Tokens to keep the user logged in for an hour before expiring. This allows users to close and reopen the site without having to login again.
- Rating System:



COP 4710 Project

Join RSOs Create RSO Logout UsfUser4

Create Event

University of South Florida

USF Football | South Florida Bulls Football Game | 4:00pm | 11/12/19

USF Football | South Florida Bulls Football Game 2 | 4:00pm | 11/19/19

South Florida Bulls Football Game 2

Football!

Meeting

USF

417-414-6328

test@bulls.usf.edu

★ ★ ★ ★ ★

5 Stars

Add Comment

USF Football | South Florida Bulls Football Game 3 | 6:00pm | 11/26/19

Public Event | USF Meet and Greet | 1:00pm | 11/23/19

Public Event | 341231232 | 32133231321pm | 331332

COP 4710 Project

Join RSOs Create RSO Logout UsfUser4

Create Event

University of South Florida

USF Football | South Florida Bulls Football Game | 4:00pm | 11/12/19

USF Football | South Florida Bulls Football Game 2 | 4:00pm | 11/19/19

South Florida Bulls Football Game 2

Football!

Meeting

USF

417-414-6328

test@bulls.usf.edu

★ ★ ★ ★ ★

Add Comment

USF Football | South Florida Bulls Football Game 3 | 6:00pm | 11/26/19

Public Event | USF Meet and Greet | 1:00pm | 11/23/19

Public Event | 341231232 | 32133231321pm | 331332

Conclusion

Overall, all requested features have been implemented using a full stack. Users are able to create events, rate them, and add comments. User's from different schools have restricted access to foreign universities, and can only view the public events. One feature that would be beneficial is to delete events after they have passed, or implementing manual deletion/editing to existing events.

The main and really only problem that occurred when developing this application was the database system I originally used did not offer support for triggers. This caused me to change the provider I was using from ClearDB to a localhost server of MySQL. Otherwise, I already had some background in creating fullstack websites from other classes, and was able to chip away at this project fairly quickly.