**CS165 Project Requirements**

1. **Short Description**

In a world like ours-- where you can reach everything in just a click-- organization of matter is a must have. Although social media is a lot better than writing all things with pen and paper, it also makes it confusing and hard for people to find certain things in the internet because social media is very broad.

The aim of this event sharing app is to provide a specific place for people to huddle, share, get and update information about current events around the corner of our university. Apparently, a state university like UP is a home for thousands of students. Every week, there are numbers of events from different colleges, new film showing, and different seminars. Can you imagine how many sign-up google forms we fill up every time there’s an event we want to go to? As a student with lots of required events and prerequisite events to attend to, it’s tiring to write same informations over and over again (Student number, name, contact information, etc) and it is also very hard for the event organizer to handle the contact details of everyone. Another purpose of this project is to help the event organizations in organizing their event by providing simple database with just one tap on their handy android phones.

1. **List of features with description**
2. Create Account

This feature allows anyone to create an account. The user can either choose to create a regular account or an organization account however, only one organization account is allowed for each organization. The regular account is basically a personal account whereas an organization account is like an admin account of the organization. If the user wants to follow an event from a particular organization or attend a particular event, he must create a regular account.

1. Account activation

A flag with default value 0, is set for every account. If the user confirmed his account using his email, the flag will be set to 1; it means his account is now active. Otherwise, it is still not active; hence, not useable.

1. Search Event

This feature allows anyone to browse the website and search for events. User can search events using the **name** of event itself or by using the **tags** of the events.

1. View Events

Aside from being able to see all events, this feature allows the users to click on the event and view its details. Important information such as date, place, person to contact, and other details are included.

1. Log-In

This feature is used by organization and other users. There are two ways you can log-in: create account for new users and normal log-in for the organization and old users.

1. Create Event

This feature allows the organization account to create a page of their event for viewing, searching and attending visitors.

1. Modify Event

This feature allows the organization account to edit information about the event.

1. Delete Event

This feature allows the organization account to delete an event they made.

1. List of Attendees

This feature allows the regular account users to click “Going” on any event they want to attend. The organization account users can get information about the list of attendees.

1. **DDL Statements**

DROP SCHEMA IF EXISTS eventsharing;

CREATE SCHEMA IF NOT EXISTS eventsharing DEFAULT CHARACTER SET latin1 ;

USE eventsharing;

DROP TABLE IF EXISTS eventsharing.GoingStats;

DROP TABLE IF EXISTS eventsharing.RegularAccounts;

DROP TABLE IF EXISTS eventsharing.Organizations;

DROP TABLE IF EXISTS eventsharing.Events;

DROP TABLE IF EXISTS eventsharing.Subscriptions;

-- REG ACCOUNT

CREATE TABLE eventsharing.RegularAccounts

(AccountID INT NOT NULL AUTO\_INCREMENT,

UserName VARCHAR(16) NOT NULL,

LastName VARCHAR(30) NULL,

FirstName VARCHAR(30) NULL,

MiddleName VARCHAR(5) NULL,

Email VARCHAR(30) NULL,

Password VARCHAR(20) NULL,

Phone VARCHAR(24) NULL,

PRIMARY KEY (AccountID)

) ENGINE = InnoDB DEFAULT CHARSET = latin1;

-- ORG ACCOUNT

CREATE TABLE eventsharing.Organizations

(OrgID INT NOT NULL AUTO\_INCREMENT,

OrgUserName VARCHAR(16) NOT NULL,

OrgFullName VARCHAR(50) NOT NULL,

Email VARCHAR(30) NULL,

Password VARCHAR(20) NULL,

OrgDesc LONGTEXT NULL,

PRIMARY KEY (OrgID)

) ENGINE = InnoDB DEFAULT CHARSET = latin1;

-- EVENT PAGE

CREATE TABLE eventsharing.Events

(EventID INT NOT NULL AUTO\_INCREMENT,

EventName VARCHAR(50) NOT NULL,

OrgID INT NOT NULL,

Tag VARCHAR(20) NULL,

Description LONGTEXT NULL,

Picture LONGBLOB NULL,

Venue VARCHAR(150) NULL,

Eventtime time NULL,

Eventdate date NULL,

PRIMARY KEY (EventID),

CONSTRAINT fk\_events\_event1

FOREIGN KEY (OrgID)

REFERENCES eventsharing.Organizations(OrgID)

) ENGINE = InnoDB DEFAULT CHARSET = latin1;

-- Going Stats

CREATE TABLE eventsharing.GoingStats

(AccountID INT NOT NULL,

EventID INT NOT NULL,

PRIMARY KEY (AccountID, EventID),

CONSTRAINT fk\_goingstats\_account

FOREIGN KEY (AccountID)

REFERENCES eventsharing.RegularAccounts(AccountID),

CONSTRAINT fk\_goingstats\_event

FOREIGN KEY (EventID)

REFERENCES eventsharing.Events(EventID)

) ENGINE = InnoDB DEFAULT CHARSET = latin1;

-- Subscribe

CREATE TABLE eventsharing.Subscriptions

(AccountID INT NOT NULL,

OrgID INT NOT NULL,

PRIMARY KEY (AccountID, OrgID),

CONSTRAINT fk\_subscriptions\_account

FOREIGN KEY (AccountID)

REFERENCES eventsharing.RegularAccounts(AccountID),

CONSTRAINT fk\_subscriptions\_org

FOREIGN KEY (OrgID)

REFERENCES eventsharing.Organizations(OrgID)

) ENGINE = InnoDB DEFAULT CHARSET = latin1;

-- Indexing

CREATE INDEX eventname\_index ON eventsharing.Events(EventName)

-- Values --

-- Regular Accounts

INSERT INTO RegularAccounts(UserName, LastName, FirstName, MiddleName, Email, Password, Phone) VALUES

('cams', 'Bacister', 'Camille Grace', 'M', 'bcamillegrace@gmail.com', 'a1b2c3d4e5', '09227150692');

INSERT INTO RegularAccounts(UserName, LastName, FirstName, MiddleName, Email, Password, Phone) VALUES

('mika', 'Lenon', 'Mikaela Jun', 'M', 'mika.lenon@gmail.com', 'f1g2h3i4j5', '09987654321');

INSERT INTO RegularAccounts(UserName, LastName, FirstName, MiddleName, Email, Password, Phone) VALUES

('pats', 'Caparoso', 'Patricia Ann', 'G', 'tricia.caparoso@gmail.com', 'a5b4c3d2e1', '0912345876');

-- Org Accounts

INSERT INTO Organizations(OrgUserName, OrgFullName, Email, Password, OrgDesc) VALUES

('TRICAMI', 'TRICAMI LOOOONG LOOONG', 'tricamiemail@gmail.com', 'gandanamin', 'desc dis hehe');

INSERT INTO Organizations(OrgUserName, OrgFullName, Email, Password, OrgDesc) VALUES

('SECOND', 'LongSecondOrg', 'secondemail@gmail.com', 'secondgandalang', 'second desc dis');

-- Events

INSERT INTO Events(EventName, OrgID, Description, Venue, Eventtime, Eventdate) VALUES

('FREE PAKAIN', 1, 'Unlimited Food, drinks, dessert', 'Department of Computer Science', '12:00 am', '2015-12-25');

INSERT INTO Events(EventName, OrgID, Description, Venue, Eventtime, Eventdate) VALUES

('2FREE PAKAIN', 2, '2Unlimited Food, drinks, dessert', '2Department of Computer Science', '12:00 am', '2015-12-28');

-- Going

INSERT INTO GoingStats(AccountID, EventID) VALUES(1,2);

-- Subscriptions

INSERT INTO Subscriptions (AccountID, OrgID) VALUES(1,2);

**MOCK UPS**

****

**SEARCH EVENTS on click**

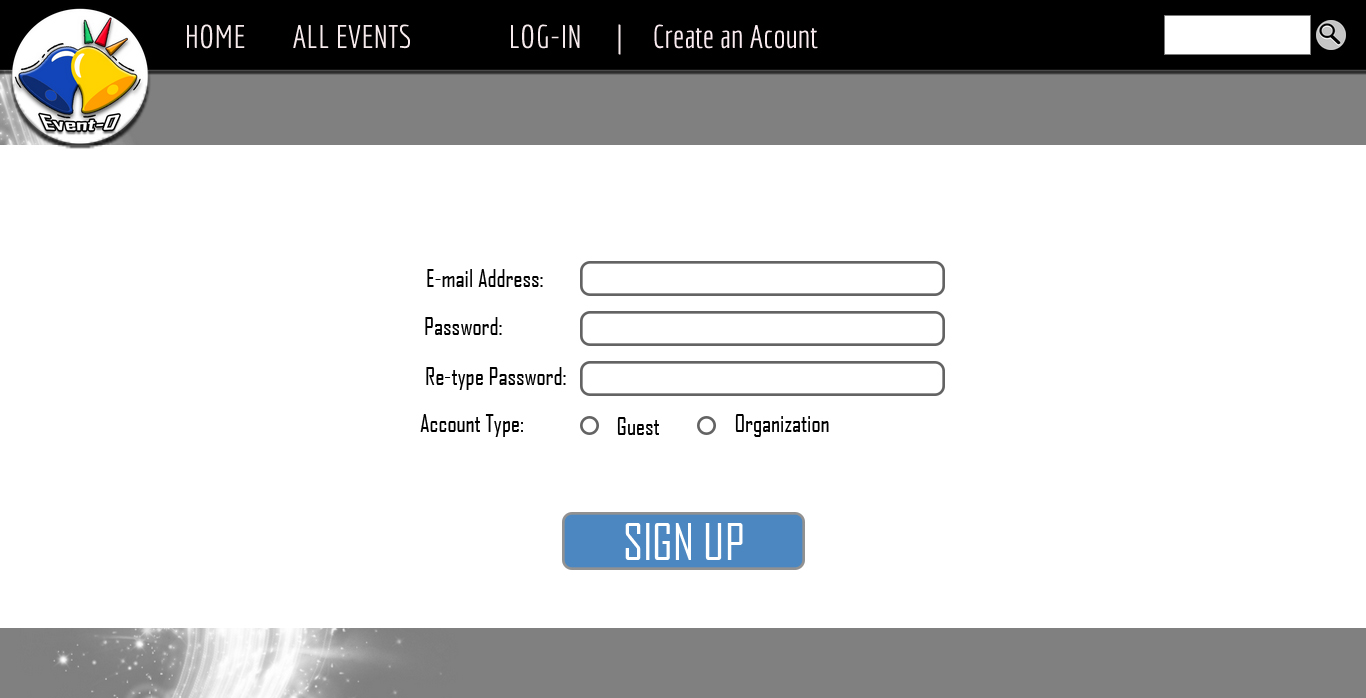
The user can search anywhere/any screen by clicking the search button on the upper right most part of any screen

SELECT \* FROM Events WHERE EventTitle = <EventTitle>;

**ACCOUNT ACTIVATION**

*Internal – upon log-in,the system will check first if the user’s account is active or not. Meaning, it determines if the user can proceed in using his account or not.*

SELECT \* FROM RegularAccounts and Organizations WHERE Activated = 1



**CREATE ACCOUNT REGULAR ACCOUNT on click**

*Lets the user create an account with this statement*

INSERT INTO RegularAccounts(UserName, LastName, FirstName, MiddleName, Email, Password, Phone)

VALUES (<Username>, <Lastname>, <Firstname>, <Middlename>, <Email>, <Password>, <Phone>);