

1. (Verified) User

a. Create Event (Pending Approval)

- i. Pre: Event does not exist
- ii. Post: Event is now pending approval from admin; inserted into SQL database
- iii. Queries:
 1. select U_ID from users where USERNAME = '\$username';
 2. insert into events (NAME, ORGANIZER, LOCATION, DESCRIPTION, MEDIAURL, STARTDATE, ENDDATE, LINK, APPROVED, U_ID) values ('\$name','\$group','\$location','\$description','\$mediaurl','\$startdate','\$enddate','\$eventlink', 0, '\$U_ID');

b. Login

- i. Pre: User is not logged in and is creating an event via the “Add Event” form
- ii. Post: User was able to successfully create an event, or was prevented
- iii. Queries:
 1. select U_ID, PRIV from users where USERNAME = '\$username';

c. Reset Password

- i. Pre: User does not remember his password
- ii. Post: Newly generated password is emailed to the user
- iii. Queries:
 1. update users set PASSWORD = '\$newpassword' where USERNAME = '\$username';

2. Guest/Unverified User

a. View Events

- i. Pre: Hasn't seen events yet
- ii. Post: Sees events and possibly decides to go to one
- iii. Queries:
 1. select * from `events` order by STARTDATE asc;

b. Filter Events

- i. Pre: Sees all events
- ii. Post: Only sees events that match the search
- iii. Queries:
 1. (select * from `events` where `name` AND `NAME` like '\$string') union (select * from `events` where `ORGANIZER` like '\$string') union select * from `events` where `LOCATION` like '\$string') union (select * from `events` where `DESCRIPTION` like '\$string') ORDER BY STARTDATE ASC;

c. Register as an “unverified user”

- i. Pre: Isn't registered to become a user and can only view events
- ii. Post: Now can create events pending admin approval on his privileges
- iii. Queries:
 1. insert into users (FIRTNAME, LASTNAME, USERNAME, EMAIL, PASSWORD, PRIV) values ('\$firstname', '\$lastname', '\$username', '\$email', '\$password', 'unverified');

- d. *Get Info About Events*
 - i. Pre: Doesn't know description of an event
 - ii. Post: Clicks on "info" button for that event and finds out more about it
- 3. **Admin**
 - a. *Approve Events*
 - i. Pre: Sees an unapproved event on the admin panel from a registered user
 - ii. Post: The event is now listed on the website
 - iii. Queries:
 - 1. Update events set APPROVED = 1 where E_ID = '\$id';
 - b. *Create Events (Without Approval)*
 - i. Pre: Event isn't listed on the website
 - ii. Post: Event is now listed on the website
 - iii. Queries:
 - 1. select U_ID from users where USERNAME = '\$username';
 - 2. insert into events (NAME, ORGANIZER, LOCATION, DESCRIPTION, MEDIAURL, STARTDATE, ENDDATE, LINK, APPROVED, U_ID) values ('\$name','\$group','\$location','\$description','\$mediaurl','\$startdate','\$enddate', '\$eventlink', 1, '\$U_ID');
 - c. *Remove Events*
 - i. Pre: The event is listed on the website
 - ii. Post: The event has now been deleted from the database
 - iii. Queries:
 - 1. delete from events where \$E_ID = '\$id';
 - d. *Change User's Privileges*
 - i. Pre: The user is listed as having PRIV = 'unverified'
 - ii. Post: The user is listed as having PRIV = 'user' and can now create events pending admin approval
 - iii. Queries:
 - 1. update users set PRIV = 'user' where \$U_ID = '\$id';