**UNIT TESTING SKELETON**

**User tests**

* User
  + View site
    - Go to site, verify that UI loads
  + View current calendar
    - View site, verify that calendar is current and populated
  + Connect to server
    - View site, verify data populates from server.
  + Facility rental request
    - Send request, verify request is sent to ?
  + Ban User
    - Ban user, verify user can't log in
  + Edit event
    - Change event, verify with set information
  + Remove event

**Event tests**

* Event
  + Donate
  + Promote
  + Download

**School tests**

* Facilities
  + Check approval status

**Social Media**

* Facebook/Twitter
  + Export function
  + Is API connected
  + Send data correctly
  + Send data via correct user
  + Token/connection testing

**Payment**

* Payment processing
* Payment confirmation
* Token/connection testing

**Database**

* Can web site connect to database/server
* Data requests
* Adding/updating/deleting from table
* Get/set table data methods

**Verification**

* User role management
* Permissions

**Testing Methods**

Homepage:

1. Does it show the user’s information (location based) events?
2. Is the information about all the event that are featured and near the user have all the data?

Sub-Pages:

1. Is there a way to know if you are logged in or not?
2. Is there a way to go homepage from the

Profile:

1. Can you find information about the user about the user when looking at the page?
2. Can you create an event if your user is allowed?
3. Can you report people? ??
4. Does it show if the proper information when the user is looking for their own profile?

Event:

1. If logged in does it show that you can attend the event?
2. Does it show all the event’s information [Event Name, Event Host, Event Description]?
3. Is there a way to share to popular social media sites?
4. Can the login user edit the event, if it was the same host?

Login:

1. Can you enter wrong data to username and password, and still login?
2. Does it prompt when username and password is not working?
3. Does it ask to find password if the user tried multiple times?

Header Menu:

1. Does the menu show all the Searching (Event, Profile), Login
2. Does it change when logged in and logged out.

User Verification:

Users having appropriate Permissions

Event Creation

Database Testing

* adding/updating/deleting from table
* fetching data when requested by the user

Getter and Setter methods

“Testing mode”

**Integration Tests:**

**General:**

* Ensure every part of the website is reachable by clicking.
* Test login information should carry over links, at least until somebody logs off.

**Database Integration:**

* Test displayed information matches that in database.
* Test login system works.
* Test validity of queries sent on UI events.
* Text fields should be stripped of any extraneous symbols to prevent SQL injection.
* Test automatic updates with Database should be automatically updated when events are

**Payment System Integration:**

* Ensure public transaction details are saved in database.
* Validate address and zip code of transaction form
* Test ability of site administrators to view payment information (which should be saved in some graphical interface not through the database)

**Social Media Integration:**

* Save information linking site account with social media accounts on database and allow users to change that information.

**Event System:**

* Test that joining an event updates information on the user’s profile
* Test automatic update the front page with recent events and swapping in and out of events periodically.
* Test events show, dissapear, and update location on calendar when being update

**Regression Testing:**

When testing changes, we should make sure first and foremost the integrity of the database is held since it holds or will hold information that is not recreatable unless we backup the information. Almost everything on the site relies on it as well so we have to check all of the places which we query the database or read display information from it. We also have to check scheduled updates else we may end up getting random problems which cannot be easily foreseeable unless focused upon.

Next we would test the systems that are related to the thing being changed. Keep an old copy, and test operations on the old and new copy to see if there are any differences. After testing everything that used the old changes, do a quick test on unrelated but important systems on the site are still working as intended.

**System Testing:**  
System testing would be done by reenacting how every user in our use case would use our web application, and making sure the requirements are met.

**When imitating the user, we would:**

* Go to site, verify that UI loads and we get a greeting with our username being part of the message to verify the database has been connected to.
* We can also mistype the password and try account creation to test the database and error checking capabilities.
* View current calendar for events and perform a couple of searches for different kinds of events
* Add a few events to our list of events we’ll be attending
* Attempt to rent out some school location. Verify we the payment procedure works appropriately and that we get some kind of verification email.
* Verify we can cancel our payment within the allotted timespan.
* Use the site’s integrated social media advertising to spread the word about our event.
* Get our mom to join our event so we can see the attendee list populating
* Kick our mom so we know we have control over the attendee list
* Update our event information on our event’s site page
* While we do everything we will also be looking at the table inside our database on another screen to see things are appropriately automatically updating.

**When imitating a school admin:**

* Facility rental request
* Send request, verify request is sent to ?
* Ban User
* Ban user, verify user can't log in
* Edit event
* Change event, verify with set information
* Remove event

**When Imitating an admin:**

**Unit Test:** Component level tests Slide (11)

**Regression testing** is the process of testing changes to computer programs to make sure that the older programming still works with the new changes (SLIDE 18)

**Integration testing:** Testing that the parts work together Slide (14)

**System Testing:** Testing the whole system from user’s perspective, and test all requirements.

All integrated applications are tested (slide 20)