**Introduction**

This test plan addresses input verification and session management for the Scuba School website. It will address functions available to users of both kinds (guests and dedicated) as well as the admin. The main goal of the testing process is to ensure that the system does not enter an unacceptable state via faulty input, and that the site behaves properly to separate guest content from dedicated user content.

**Items to be Tested**

Acceptance Tests: All direct site functions will be placed under scrutiny. Any page in which the user places direct input to the system will be tested for improper inputs to ensure that they do not negatively affect the system. Every page will be tested to make sure that it displays properly and provides proper access (or lack thereof) depending on whether or not the user is logged in.

Integration Tests: All functions which access the database will be tested to ensure that changes to the database are immediately and properly reflected on all pages of the website.

**Items not to be Tested**

The SQL database itself will not be given Unit Tests. It is assumed that queries succeed as long as the input is valid. Race conditions between multiple pages (i.e. if a course is removed at the same moment a user registers for it) will not be tested due to infeasibility with available resources.

**Approach**

Individual input fields will be tested for different sets of inputs. In most cases, errors caused by faulty input are independent of input in other fields of the same form. In cases where this is not true, conjoining erroneous inputs will also be tested. The system will also be tested after changing the state of the system, such as by adding new courses, to ensure that changes are reflected correctly in all site functions.

Testing will be performed on a page-by-page basis.

Home Page – home.php

There is no user input on this page. However, the following requirements must be tested.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should not provide a link to the homepage, as the user is already there | None | The bottom of the page provides no links. |  |

About Us Page – about.php

There is no user input on this page. However, the following requirements must be tested.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should provide a link to the home page | None | The bottom of the page provides a link to home.php |  |

List Page – list.php

There is no user input on this page. However, the following requirements must be tested.

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should provide a link to the home page | None | The bottom of the page provides a link to home.php |  |
| If the table is empty, the page should display no data. | ScubaCourse table has no entries | Table is empty |  |
| If there is data to be displayed, all classes must be displayed | ScubaCourse table has entries | All entries are properly displayed |  |
| If a student registers for a course, the capacity should be increased by one | Directly following the completion of schedule.php functions | The capacity for the indicated course is one higher than before. Capacities of other courses are not affected. |  |
| If the admin changes the data for a course, the new data must be immediately reflected in the list | Directly following the completion of admin.php functions | The course metadata is modified to the input values. |  |

Registration Page – register.php, createaccount.php

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in  Register.php page | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in  Register.php page | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in  Register.php page | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in  Register.php page | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should provide a link to the home page | Register.php page  Or createaccount.php page | The bottom of the page provides a link to home.php |  |
| If the user is logged in, the page should not offer any services | User is logged in  Register.php page  Or createaccount.php page | The page must not display input fields, perform any action, or make any changes. The page must display an error message. |  |
| If the user accesses createaccount.php directly, the page should not perform any action | User is not logged in  Createaccount.php run without posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user inputs a username that is taken, the account should not be created | User is not logged in  Createaccount.php run with posted data with conflicting primary key | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If any input field is empty, data is not processed | User is not logged in, creataccount.php run with posted data that has one or more empty fields. | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If either the username or password input is less than four characters, the account should not be created. | User is not logged in, createaccount.php run with posted data of insufficient length (<4 chars) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If any input is greater than twenty characters, the account should not be created. | User is not logged in, createaccount.php run with posted data too long (<20 chars) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If any input contains nonalphanumeric characters, the account should not be created. | User is not logged in, createaccount.php run with posted data of incorrect input (nonalphanumeric chars) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user inputs alphanumeric data with a length between 4 and 20 characters (for username, password) or between 1 and 20 characters (firstname, lastname) to register.php, it should be posted to createaccount.php and continue normally | User is not logged in  Createaccount.php run with valid posted data. | Account created with specified username and password – data viewable on admin CP and in database. |  |

Login Page – login.php, checklogin.php

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in  login.php page | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in  login.php page | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in  login.php page | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in  login.php page | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should provide a link to the home page | login.php page Or checklogin.php page | The bottom of the page provides a link to home.php |  |
| If the user is logged in, the page should not offer any services | User is logged in  login.php page | The page must not display input fields, perform any action, or make any changes. The page must display an error message. |  |
| If the user accesses checklogin.php directly, the page should not perform any action | User is not logged in  checklogin.php run without posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user inputs a username and password that do not match a database entry for the account, the service should fail. | User is not logged in  Createaccount.php run with posted data that does not match the database entry | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user inputs a username and password that do not exist at all in the database, the service should fail. | User is not logged in  Createaccount.php run with posted data that does not exist in the database | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user inputs data to login.php corresponding to an existing username and password, it should be posted to checklogin.php and continue normally | User is not logged in  checklogin.php run with valid posted data. | User logged in and session created. Session variables set with user account data. |  |

Log Out Page – logout.php

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should provide a link to the home page | None | The bottom of the page provides a link to home.php |  |
| If the user is not logged in, the page should not offer any services | User is not logged in | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user is logged in, the session should be terminated and all session variables discarded | User is logged in | Session is terminated. User is logged out. |  |

Course Registration Page – schedule.php, coursesignup.php

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in  schedule.php page | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in  schedule.php page | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in  schedule.php page | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in  schedule.php page | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should provide a link to the home page | schedule.php page Or coursesignup.php page | The bottom of the page provides a link to home.php |  |
| If the user is not logged in, the page should not offer any services | User is not logged in  schedule.php page Or coursesignup.php page | The page must not display input fields, perform any action, or make any changes. The page must display an error message. |  |
| If the user accesses coursesignup.php directly, the page should not perform any action | User is logged in  coursesignup.php run without posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the data input to schedule.php is empty, the page should not perform any action | User is logged in  Coursesignup.php run with empty posted data. | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the course ID input to schedule.php is not a number, the page should not perform any action | User is logged in  Coursesignup.php run with nonnumeric posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the course ID input to schedule.php does not correspond to an existing course, the page should not perform any action | User is logged in  Coursesignup.php run with incorrect posted data (class ID nonmatching) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user is already registered for the course specified by the course ID input to schedule.php, the page should not perform any action. | User is logged in  Coursesignup.php run with posted data that causes a primary key conflict | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the user does not have sufficient money to afford the course, the page should not perform any action. | User is logged in  Coursesignup.php run with correct posted data, but insufficient funds | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If course ID input corresponds to an existing course, and the user’s account has sufficient funds to afford the course, class should be registered normally | User is logged in  Coursesignup.php run with correct posted data and sufficient funds | User’s account is deducted by the price of the course, reflected immediately in both session variables and in the database. User is given an entry in the TakesCourse table for the course, reflected immediately on the list.php page regarding capacity. User views the class successfully on their user control panel. |  |

User Control Panel – userpanel.php, changename.php, money.php

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If the user is a guest, the page should welcome them as a guest, and grant a method to create an account. | User is not logged in  userpanel.php page | The top right of the page must provide a link to register.php |  |
| If the user is logged in, the page should welcome them with their name, and grant a method to terminate their session. | User is logged in  userpanel.php page | The top right of the page must display the user’s name, and provide a link to logout.php |  |
| If the user is a guest, the top of the page should only provide links to features accessible by guests. | User is not logged in  userpanel.php page | The top of the page must provide links to login.php, list.php, and about.php |  |
| If the user is logged in, the top of the page should provide links only to features accessible by logged in users. | User is logged in  userpanel.php page | The top of the page must provide links to userpanel.php, list.php, schedule.php, and about.php |  |
| The bottom of the page should provide a link to the home page | userpanel.php page or changename.php page or money.php page | The bottom of the page provides a link to home.php |  |
| If the user is not logged in, the page should not offer any services | User is not logged in  userpanel.php page | The page must not display input fields, perform any action, or make any changes. The page must display an error message. |  |
| If the user accesses changename.php or money.php directly, the page should not perform any action | User is logged in  changename.php or money.php run without posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If any data input to userpanel.php is empty, the resulting page should not perform any action | User is logged in  input run with empty posted data. | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If input to changename.php is nonalphanumeric, service should fail. | User is logged in  Changename.php run with nonalphanumeric posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If input to changename.php is greater than 20 characters, service should fail. | User is logged in  Changename.php run with posted data too long (<20 chars) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If input to money.php is nonnumerical, service should fail | User is logged in  money.php run with nonnumerical posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If input to money.php is nonpositive, service should fail | User is logged in  Money.php run with nonpositive number as posted data | The page must not perform any action, or make any changes. The page must display an error message. |  |
| Courses registered for should be viewable in the user control panel. | User is logged in  Directly following services of schedule.php | The class registered for, and no other class, should immediately appear in the user control panel. |  |
| If input to changename.php is alphanumeric and between 1 and 20 characters, name is changed. | User is logged in  Changename.php run with valid posted data | The user’s new desired name is reflected immediately in the session variables, database, and admin control panel |  |
| If input to money.php is numeric and greater than 0, money is changed. | User is logged in  Money.php run with valid posted data | The user’s new money total is reflected immediately in session variables, database, and admin control panel. |  |

Admin Control Panel – admin.php, adminlist.php, adminlist2.php, adminadd.php, admindrop.php

|  |  |  |  |
| --- | --- | --- | --- |
| Description | State | Expected Outcome | Observed Outcome |
| If adminlist.php, adminlist2.php, adminadd.php, or admindrop.php are accessed directly, service should fail. | Auxiliary page accessed directly | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If any input to any field is empty, service should fail | One or more input fields left blank for any page service | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If the admin key is incorrect for any input, the service should fail. | Admin key input to any page does not correspond to proper value (note: admin key is case-insensitive) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If classID input to adminlist2.php is nonnumeric, service should fail. | Adminlist2.php run with nonnumeric classID input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If name input to adminadd.php is greater than 30 characters, service should fail. | Adminadd.php run with name input of > 30 characters | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If location input to adminadd.php is greater than 30 characters, service should fail. | Adminadd.php run with location input of > 30 characters | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If description input to adminadd.php is greater than 100 characters, service should fail. | Adminadd.php run with description input of > 100 characters | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If month input to adminadd.php is nonnumeric, service should fail. | Adminadd.php run with nonnumeric month input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If month input to adminadd.php is numeric and less than 1, or greater than 12, service should fail. | Adminadd.php run with month input outside acceptable bounds (1~12) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If month input to adminadd.php is numeric and noninteger, service should fail | adminadd.php run with month input not a whole number | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If day input to adminadd.php is nonnumeric, service should fail. | Adminadd.php run with nonnumeric day input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If day input to adminadd.php is numeric and less than 1, or greater than 31, service should fail. | Adminadd.php run with day input outside acceptable bounds (1~30) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If day input to adminadd.php is numeric and noninteger, service should fail | Adminadd.php run with day input not a whole number | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If year input to adminadd.php is nonnumeric, service should fail. | Adminadd.php run with nonnumeric year input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If year input to adminadd.php is less than 2000 and greater than 2100, service should fail. | Adminadd.php run with year input outside acceptable bounds (2000~2100) | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If year input to adminadd.php is numeric and noninteger, service should fail. | Adminadd.php run with year input not a whole number | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If capacity input to adminadd.php is nonnumeric, service should fail. | Adminadd.php run with nonnumeric capacity input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If capacity input to adminadd.php is numeric and less than 1, service should fail. | Adminadd.php run with nonpositive capacity input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If capacity input to adminadd.php is numeric and noninteger, service should fail. | Adminadd.php run with capacity input not a whole number. | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If price input to adminadd.php is nonnumeric, service should fail. | Adminadd.php run with nonnumeric price input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If price input to adminadd.php is numeric and less than 1, service should fail. | Adminadd.php run with nonpositive price input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If price input to adminadd.php is numeric and noninteger, service should fail. | Adminadd.php run with price input not a whole number. | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If class ID input to admindrop.php is nonnumeric, service should fail. | Admindrop.php run with class ID input not a number | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If classID input to admindrop.php does not correspond to an existing class, service should fail. | Admindrop.php run with noncorresponding class Id input | The page must not perform any action, or make any changes. The page must display an error message. |  |
| If admin key input to adminlist.php is correct, service should proceed | Adminlist.php run with correct admin key input | The page must display a table containing data of all site users in a clear and readable format. |  |
| If a new user registers, their personal data should be immediately reflected in adminlist.php | Directly after the services of createaccount.php | The user who just registered should have their data reflected in the table on the adminlist.php page. |  |
| If adminkey input to adminlist2.php is correct and class ID input corresponds to an existing class, service should proceed | Adminlist2.php run with correct admin key input and class ID corresponding to an existing class | The page must display class metadata along with a table containing data of all site users registered for the corresponding class in a clear and readable format. |  |
| If a user signs up for a course, this should be immediately reflected in adminlist2.php | Directly after the services of coursesignup.php | The user who just signed up for a course should have their data reflected in the adminlist2.php page for that course, and not in any other course. |  |
| If adminkey input to adminadd.php is correct, and all class information is valid, service should proceed | Adminadd.php run with correct admin key input, name and location fields both less than 30 characters, description field less than 100 characters, month field a positive integer between 1 and 12, day field a positive integer between 1 and 31, year field a positive integer between 2000 and 2100, capacity and price fields both positive integers | The page must add the course to the database. The class ID should be generated automatically using an auto incrementer. |  |
| If a new course is created, it should be immediately reflected in list.php | Directly after the services of adminadd.php | The course must be viewable by a user on the list.php page |  |
| If a new course is created, it should be immediately accessible in adminlist2.php | Directly after the services of adminadd.php | The course must be viewable by the admin on the adminlist2.php page |  |
| If a new course is created, it should be immediately available for signups | Directly after the services of adminadd.php | A user must be able to successfully sign up for the course which was just added. |  |
| If adminkey input to admindrop.php is correct, and class ID corresponds to an existing course, services should proceed | Admindrop.php run with correct admin key input and class ID corresponding to an existing class | The page should remove the specified course from the database, and no other course. |  |
| If a course is deleted, it should be immediately reflected in list.php | Directly after the services of admindrop.php | The course must no longer be viewable by a user on the list.php page |  |
| If a course is deleted, it should no longer be accessible in adminlist2.php | Directly after the services of admindrop.php | The course must no longer be viewable by the admin on the adminlist2.php page. |  |
| If a course is deleted, it should no longer be available for registration | Directly after the services of admindrop.php | A user must not be able to successfully sign up for the course which was just deleted. |  |