

Test Case Dashboard_TC01

Test Case ID: Dashboard_TC01

Test Designed by: Kenny

Test Priority (Low/Medium/High): High

Test Designed Date: 02/Apr/20

Module Name: Dashboard.js

Test Title: Dashboard Functionality

Description: Test the dashboard's content, associated links, and other functionality.

Test Type: Functional

Related Use Cases:

Pre-conditions: User has a valid account and is on the login page

Dependencies: Login system is working

Step	Test Steps	Test Data	Expected Result
1	Navigate to the dashboard, by logging in with an account.	A valid set of account info.	Taken to the dashboard with all associated events in the correct columns.
2	Click the "Create an Event" button.		Taken to "Create an Event" page.
3	Return to the dashboard with the "Dashboard" button in the drop down in the top right corner.		Taken to the dashboard.
4	Click the "Join an Event" button.		Popup with text box, "join" button, and "cancel" button.
5	Click "Cancel" button		Taken to the dashboard. Popup is hidden.
6	Click the "Join an Event" button.	A valid invite code.	Taken to the event page associated with the invite code used.
7	Return to the dashboard with the "Dashboard" button in the drop down in the top right corner.		Taken to the dashboard. New event appears under "Joined Events".
8	Click the "Event Details" button on an event.	An event the user is associated with.	Taken to the matching event page.
9	Return to the dashboard with the "Dashboard" button in the drop down in the top right corner.		Taken to the dashboard.
10	Click the "Event Details" button on a different event.	Another event the user is associated with.	Taken to the matching event page.
11	Return to the dashboard with the "Dashboard" button in the drop down in the top right corner.		Taken to the dashboard.
10	Click the "Score Report" button on an event.	An event the user is the chair of.	Taken to the matching Score Report of the matching event.
11	Return to the dashboard with the "Dashboard" button in the drop down in the top right corner.		Taken to the dashboard.

Test Case EventDetails_TC01 - Chair

Test Case ID: EventDetails_TC01

Test Designed by: Kenny Houston

Test Priority (Low/Medium/High): High

Test Designed Date: 09/Apr/20

Module Name: EventDetails.js

Test Title: New Event Details for Chair

Description: Test all functionalities of the Event Details page from the perspective of the chair

Test Type: Functional

Related Use Cases: UC05

Pre-conditions: A newly created event was made by the chair. The chair is logged in and on the dashboard.

Dependencies: Dashboard

Step	Test Steps	Test Data	Expected Result
1	Navigate from the Dashboard to the Event Details page.		Taken to the Event Details page which shows the event's name, description, date, time, and location, all correctly. Also, there is an empty Invite Code table. A table for judges just listing the chair, who is in group "none". There is an empty table for entries.
2	Click the Edit Details button.		Taken to the Edit Event page with matching information.
3	Navigate back to the Event Details page.		No changes.
4	Click the Create a Rubric button.		Taken to Create a Rubric page.
5	Navigate back to the Event Details page.		No changes.
6	Click the Score Report button.		Taken to the appropriate (but empty) Score Report page.
7	Navigate back to the Event Details page.		No changes.
8	Click the Assigned Entries button		Taken to an empty assigned entries page.
9	Navigate back to the Event Details page.		No changes.
10	Click the Assign Groups button		Taken to the matching Assign Groups page.
11	Navigate back to the Event Details page.		No changes.
12	Click the "+" button in the Invite Codes table		A new row is added with an Invite Code. The row has a number, a code, and a Delete button.
13	Click the "+" button in the Invite Codes table again		Another row appears for a total of 2 rows. Each code is unique.
14	Click the Delete button on an Invite Code.		The corresponding row is removed, leaving the other Invite Code in the table.

Post-conditions: The Invite Code table has a single invite code in it. Otherwise, the state of the event is unchanged.

Test Case EventDetails_TC02 - Judge

Test Case ID: EventDetails_TC02

Test Designed by: Megan Phan

Test Priority (Low/Medium/High): High

Test Designed Date: 13/Apr/20

Module Name: EventDetails.js

Test Title: Event Page Judge Test

Description: Test all the functions related to judges in event page

Test Type: Functional

Related Use Cases: UC5

Pre-conditions: The judge already joined the event, is logged in, and is on the Dashboard

Dependencies: Dashboard

Step	Test Steps	Test Data	Expected Result
1	Navigate from Dashboard to Event Details		Taken to the Event Details page that shows the event info (title, datetime, location, and description). The judge can also see an Assigned Entries button
2	Click on Assigned Entries button		Taken to the assigned entries page. If the judge was already assigned some entries, they will appear there.
3	Go back to Event Details page		No change

Post-conditions:

Nothing changes

Test Case EventDetails_TC03 - Anon

Test Case ID: EventDetails_TC03

Test Designed by: Megan Phan

Test Priority (Low/Medium/High): High

Test Designed Date: 13/Apr/20

Module Name: EventDetails.js

Test Title: Event Page Anonymous Test

Description: Test all the functions related to being anonymous when access the event page

Test Type: Functional

Related Use Cases:

Pre-conditions: Having the link to the event and is not logged in.

Dependencies: the link to the event

Step	Test Steps	Test Data	Expected Result
1	Navigate to the event details by enter the link of the event		The event details page appears with the event general info like title, date, location, a message (You do not need an account to register), and a register button
2	Click on the Register button		Taken to the event register page

Post-conditions: User able to register without having an account

Test Case UC01_TC01

Test Case ID: UC01_TC01

Test Designed by: Hau Ha

Test Priority (Low/Medium/High): High

Test Designed Date: 09/Apr/20

Module Name: CreateAccount.js

Test Title: Create an Account functionality

Description: Test all functionalities of the Sign-up page

Test Type: Functional

Related Use Cases:

Pre-conditions: None

Dependencies:

Step	Test Steps	Test Data	Expected Result
1	From the Landing page, navigate to the Create an Account page by selecting the Sign-Up button		Taken to the Create an Account page which shows an empty form.
2	Select the sign-up button		The system notifies that the first name is invalid, which is empty
3	Enter an incorrect first name	!Micheal2	The system notifies that the first name is invalid.
4	Enter a correct first name	Micheal	The invalid message disappears. The system accepts the first name.
5	Enter an incorrect last name	!Scherger2	The system notifies that the last name is invalid.
6	Enter a correct last name	Scherger	The invalid message disappears. The system accepts the last name.
7	Enter an invalid email	mscherger.fakemail.com	The system notifies that the email is invalid.
8	Enter a correct email	mscherger@fakemail.com	The invalid message disappears. The system accepts the email.
9	Enter an invalid password	abcde	The system notifies that the password is invalid. The password must be between 6-20 characters
10	Enter a valid password	abcdef	The invalid message disappears. The system accepts the password.
11	Enter a different password in the confirm password space	abcdefg	The system notifies that the password does not match.
12	Enter the correct password	abcdef	The invalid message disappears. The system accepts the confirm password
13	Select the sign-up button		Taken to the Log-in page which shows an empty form. The system creates and saves a new account in the database
14	Navigate to the Create an Account page by Selecting the sign-up button		Taken to the Create an Account page which shows an empty form.
15	Select the Already have an account? Log in Link		Taken to the Log-in Page which shows an empty form

Post-conditions: A new account is saved in the database

Test Case UC02_TC01

Test Case ID: UC_TC01

Test Designed by: Action

Test Priority (Low/Medium/High): High

Test Designed Date: 05/Apr/20

Module Name: login.js

Test Title: Login Functionality

Description: Test the Evaluate login page, associated links, and other functionality.

Test Type: Functional

Related Use Cases:

Pre-conditions: User should already have created an account and must have a valid username and password.
Dependencies:

Step	Test Steps	Test Data	Expected Result
1	Navigate to login page	/login	There are an empty username and password fields. Remember Me box is unchecked. Login button is disabled.
2	Provide username to username field.	Valid username	No validation error message appears.
3	Provide password to password field.	Valid password	No validation error message appears and the Login button becomes enabled.
4	Click on Login button		Taken to the appropriate user's dashboard.
5	Logout with the dropdown in the top right corner.		Taken to the Login page, with empty fields.
6	Click on "Don't have an account? Create one" link		Taken to Create an Account page.
7	Click on the Login button in the top right corner.		Taken to Login page
8	Click on "Forgot your password? Reset it" link		Taken to Forgot Your Password page.

Post-conditions: None

Test Case UC02_TC02

Test Case ID: UC02_TC02

Test Designed by: Jordyn Dent

Test Priority (Low/Medium/High): Med

Test Designed Date: 03/Apr/2020

Module Name: login.js

Test Title: Reset an account password

Description: Test the “forgot password” feature on the login page

Test Type: Functional

Related Use Cases:

Pre-conditions: User has created an account and navigated to the login page
Dependencies:

Step	Test Steps	Test Data	Expected Result
1	Navigate to login page	Login URL	Login page loaded properly.
2	Click forgot/reset password link		Redirected to forgot password page
3	Click the email field and type in valid email for an existing user account	Valid account email	No validation error msg appears
4	Click the “send email” button		Should see “email sent” popup with its description, then should receive and email within 10-15 minutes
5	Navigate to the new email		Email should load with a reset link
6	Click the password reset link		Redirected to reset password page
7	Click the email field and type in valid user account email whose password will be changed	Valid account email	No validation error msg appears
8	Click the new password field and type in a new account password	Valid password	No validation error msg appears
9	Click the confirm password field and type in the same password as in the new account password field	Valid password from previous input field	No validation error msg appears
10	Click reset password button		Redirected to login page
11	Log in to account with new password	Valid account email and password	Redirected to dashboard

Post-conditions:

User is validated with the database and successfully logged in to the account. The account session details are logged in the database.

Test Case UC03_TC01 - Create an Event

Test Case ID: UC03_TC01

Test Designed by: Kenny Houston

Test Priority (Low/Medium/High): High

Test Designed Date: 09/Apr/20

Module Name: CreateEvent.js

Test Title: Create an Event

Description: Test all functionalities of the Create an Event page

Test Type: Functional

Related Use Cases:

Pre-conditions: A user must be logged in and on the dashboard.

Dependencies: Dashboard, Event Details

Step	Test Steps	Test Data	Expected Result
1	Click the Create an Event button on the Dashboard.		Taken to the Create an Event page which has empty fields for the event's name, description, and location. There is also a field for the date and time, initialized to the date and time the page was loaded.
2	Enter a name for the event.	"A Test Event"	The field fills with the entered information.
3	Attempt to enter a date/time value with an invalid format.	"October first"	The text field does not change.
4	Attempt to enter another date/time value with an invalid format.	"2020/12/02"	An error appears under the text field since the date time is incomplete.
5	Select the calendar icon to the right of the date/time text box.		A date/time selection gui appears
6	Select a date and time.	16/May/2020 1PM	The date/time appears in the text field with no errors.
7	Enter a location for the event.	"Tucker 137"	The field fills with the entered information.
8	Enter a description for the event.	"A test description"	The field fills with the entered information.
9	Click the Create button		Taken to the matching Event Details page, displaying the newly created event.

Post-conditions:

The user is the chair of the newly created event.

Test Case UC03_TC02 - Edit an Event

Test Case ID: UC03_TC02

Test Designed by: Kenny Houston

Test Priority (Low/Medium/High): High

Test Designed Date: 09/Apr/20

Module Name: CreateEvent.js

Test Title: Edit an Event

Description: Test all functionalities of the Edit an Event page

Test Type: Functional

Related Use Cases:

Pre-conditions: A user must be logged in and on an Event Details page for an event they created. .

Dependencies: Dashboard, Event Details

Step	Test Steps	Test Data	Expected Result
1	Click the Edit an Event button on the Event Details page.		Taken to the Edit an Event page which has fields for the event's name, date, time, location, and description, all initialized with the matching values from the previous page.
2	Change the name for the event.	*A Modified Test Event*	The field fills with the newly entered information.
3	Select the calendar icon to the right of the date/time text box.		A date/time selection gui appears
4	Select a different date and time.	20/Oct/2020 3:30PM	The new date/time appears in the text field with no errors.
5	Enter a location for the event.	*Tucker 215*	The field fills with the newly entered information.
6	Enter a description for the event.	*An altered description*	The field fills with the newly entered information.
7	Click the Save Changes button		Taken to the matching Event Details page, displaying the newly modified event information. Invite codes, judges, and entries have not changed.

Post-conditions:

All the modified data has taken the place of the old event data, while still having the same invite codes, judges, and entries.

Test Case UC04_TC01

Test Case ID: UC04_TC01

Test Designed by: Hau Ha

Test Priority (Low/Medium/High): High

Test Designed Date: 04/Apr/20

Module Name: CreateRubric.js

Test Title: Create a Rubric Functionality

Description: Test all functionalities of the Create a Rubric page

Test Type: Functional

Related Use Cases:

Pre-conditions: An event was made by the chair and no rubric has been created yet.

Dependencies:

Note: When referring to a category or subcategory with a *level* (not a *name*) of x, it is shortened to category x, or subcategory x.

Step	Test Steps	Test Data	Expected Result
1	From the Event Details page, select Create a Rubric.		Taken to the Create a Rubric page which shows a single empty category.
2	Give the empty category a name and description.	name: Cat 1 description: Desc 1	Text boxes accept the changes.
3	Select Add Category.		An empty category row appears with: Level: 2, empty Name, Max Score: 10, empty Description, and Delete and Add buttons
4	Give the empty category a name and description.	name: Cat 2 description: Desc 2	Text boxes accept the changes.
5	Select the Plus button of category 2.		An empty subcategory row appears with: Level: 2.1, empty Name, Max Score: 10, empty Description and a Delete button (no Add button). The max score row of the parent category is disabled and shows a value of 10.
6	Give the empty subcategory a name and description.	name: Subcat 2.1 description: Desc 2.1	Text boxes accept the changes.
7	Add another subcategory to category 2 and modify its Max Score value.	15	The max score of the parent category is changed to 25. Total score of 35.
8	Give the new subcategory a name and description.	name: Subcat 2.2 description: Desc 2.2	Text boxes accept the changes.
9	Modify the score of category 1	20	The total max score is changed to 45.
10	Select the Delete button of subcategory 2.1.		The subcategory row disappears. Subcategory 2.2 moves up and has its Level changed to 2.1 (but with no other modifications). The max score column of the parent category is deducted 10 and now equals 15. Total score of 35.
11	Select the Delete button of category 1.		The category row disappears. All categories and subcategories below it move up, but with no other modifications.
12	Select the Save Button	A single category with a	Taken to the Event Details page. The rubric is saved in the system. The Create a Rubric button is

		single subcategory.	replaced by an Edit Rubric button.
13	Select the Edit Rubric button		All previous categories and subcategories reappear with no changes.
14	Select the Delete All button		Every category and subcategory disappears, leaving a single empty category. Its initial Max Score is 10, and the Rubrics total score is 10.

Post-conditions:

- A complete Rubric is saved in the database (not the empty rubric which was never saved).
- The Create a Rubric button in the Event Detail page is replaced by the Edit Rubric button.

Test Case UC05_UC11_TC01

Test Case ID: UC05_UC11_TC01

Test Designed by: Action

Test Priority (Low/Medium/High): Med

Test Designed Date: 10/May/20

Module Name: EventDetail.js

Test Title: Invite Code Functionality

Description: Test the Invite Code Functionality

Test Type: Functional

Related Use Cases:

Pre-conditions: Chair needs to log in and own an event

Dependencies: UC03 Create an Event

Step	Test Steps	Test Data	Expected Result
1	click on "event detail" button on the dashboard as a chair		Taken to the event detail page
2	Click on the plus icon multiple times	random Invite codes	generates multiple invite codes
3	Click on the red bin icon		Deletes the associate invite code
4	Copy one of the invite codes and send it to the judge		Judge receives the code
5	login as a judge	/login	Successfully logs in and taken to the dashboard
6	Click on the join event button		Popup with text box, "join" button, and "cancel" button.
7	Click "Cancel" button		Taken to the dashboard. Popup is hidden.
8	Click on the join event button		Popup with text box, "join" button, and "cancel" button.
9	Enter your invite code to the text field and click join	A valid invite code	No error appears and you are taken to the event page associated with the invite code used.

Post-conditions:

The judge is now associated with the joined event.

Test Case UC07_TC01

Test Case ID: UC07_TC01

Test Designed by: Hau Ha

Test Priority (Low/Medium/High): High

Test Designed Date: 13/Apr/20

Module Name: ScoreReport.js, ExportLXSL.js

Test Title: Score Report functionality

Description: Test all functionalities of the Score Report page

Test Type: Functional

Related Use Cases:

Pre-conditions: All judges finish grading their assigned entries.

Dependencies:

Note:

Step	Test Steps	Test Data	Expected Result
1	From the Dashboard, navigate to the Score Report page by selecting the Score Report Button		Taken to the Score Report page which shows the score of every entry.
2	Select Show Score Details button of an entry		The system shows a summary for that particular entry, which includes the score of every judge, and a button with each judge to modify the score
3	Select Edit Entry button of a score		Taken to the Grading Entry page of that entry. The chair can modify the score
4	Select Submit button		Taken to the Score Report page which shows the score of every entry.
5	Select the Export to XLSL button		Download an Excel file includes the details of the score report

Post-conditions:

An Excel file is created as a summary of the Score Report

Test Case UC08_TC01

Test Case ID: UC08_TC01

Test Designed by: Action

Test Priority (Low/Medium/High): Med

Test Designed Date: 15/Apr/20

Module Name: AssignGroups.js

Test Title: Assign Groups Functionality

Description: Assigning groups

Test Type: Functional

Related Use Cases:

Pre-conditions: Chair needs to be in the event detail page. Judges and entries have joined the event. No groups have been assigned yet.
Dependencies:

Step	Test Steps	Test Data	Expected Result
1	Click the Assign Groups button on the Event Detail page.		Taken to the Assign Groups page, showing all the judges and entries from the previous page. All judges and entries have the default group of A. The chair has the default group of None.
2	Using the Slider select each of the possible group numbers from 1 to 5, leaving it at group 5 at the end.		The options in the dropdown change depending on the value of the slider: 1-A; 2-A,B; 3--A,B,C; 4-A,B,C,D; 5-A,B,C,D,E
3	Change the group of a judge to group B.		The judge has their group changed to B and the summary shows one less judge in group A, and one more judge in group B.
4	Change another judge to group C.		The judge has their group changed to C and the summary shows one less judge in group A, and one more judge in group C.
5	Change the group of an entry to group B.		The entry has their group changed to B and the summary shows one less entry in group A, and one more entry in group B.
6	Change another entry to group C.		The entry has their group changed to C and the summary shows one less entry in group A, and one more entry in group C.
7	Click on the Save button		Taken to the Event Details page, showing matching data from final Assign Groups page.
8	Click the Assign Groups button		Taken to the Assign Groups page. All group numbers match the previous page. The Number of Groups slider has a default value of 3, since the highest group used is group C.

Post-conditions:

Test Case UC12_TC01 - Assigned Entries

Test Case ID: UC12_TC01

Test Designed by: Jackie Ye

Test Priority (Low/Medium/High): High

Test Designed Date: 04/07/20

Module Name: AssignedEntries.js

Test Title: Assigned Entries page functionality

Description: Test that the page is working correctly

Test Type: Functional

Related Use Cases: N

Pre-conditions: The test CCSC event and associated data has been initialized, with judges and entries being assigned to groups
Dependencies: Grade Entry page

Step	Test Steps	Test Data	Expected Result
1	Login with jve@fakemail.com . Select "CCSC" Event Details. Then, click the Assigned Entries button.	A judge account, in group B	Taken to the Assigned Entries page, with 3 entries listed. For each entry, a title, a description, and a score, are present.
2	Click the first entry.		Taken to the Grade Entry page, showing matching title and same score from the previous page.
3	Return to the Assigned Entries page.		Taken to the Assigned Entries page, with 3 entries.
4	Click the second entry.		Taken to the Grade Entry page, showing matching title and same score from the previous page.
5	Logout and log back in with mscherger@fakemail.com . Select "CCSC" Event Details. Click the Assigned Entries button.	A chair account, in group A	Page redirected to AssignedEntries.js, with 3 Entries (different from what was shown to the judge account).

Post-conditions:

Test Case UC13_TC01

Test Case ID: UC13_TC01

Test Designed by: Jackie Ye

Test Priority (Low/Medium/High): High

Test Designed Date: 04/13/20

Module Name: GradeEntry.js

Test Title: Grading an Entry functionality test

Description: Verify the functionality of the Grade and Entry system

Test Type: Functional

Related Use Cases: UC12

Pre-conditions: User is a judge in an event with a created rubric and submitted entries.

Dependencies: Assigned Entries system is working

Step	Test Steps	Test Data	Expected Result
1	From the Assigned Entries page, select an entry, while noting the title and current score.		Redirected to the Grade an Entry page for the matching entry with matching scores.
2	change the score for each categories, leave some comments		The current page should be redirected to AssignedEntries.js
3	Check the database for the entry that had been updated		The scores and comments of the entry is successfully updated

Post-conditions:

The selected entry is now updated correctly in both database and frontend.