

Acceptance tests

Acme-Easy 2.0

Provide the title and version of the project to which this document applies.

Indexing data	
Designer	Group 32
Tester	Group 32
Notes	These acceptance test are made to group 22

Group 32:

Javier Carmona Ferri
Juan Carlos Pérez García
Julio Carmona Ferri
Raúl Cayetano Díaz León

Functional requirement <001>

Description

A user who is not authenticated must be able to register to the system as an author.

Access

From the main view, is possible to click in the link that takes us to the register view.

Checks

Check <#001A>	
Description	Register with the fields empty
Expected	We expect the System to catch and show us the errors.
Outcome	After keeping the fields empty, we click on the button “Save” and of course the System doesn’t save the new author.
Notes	

Check <#001B>	
Description	Register with a birth date in the future
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the birth date in the future and the System doesn’t allow save it.
Notes	

Check <#001C>	
Description	Register with an invalid credit card number
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have filled the credit card number with an invalid number so the System doesn’t allow save it.
Notes	

Check <#001D>	
Description	Register with the expiration date of the credit card in the past.
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the expiration date in the past so the System doesn't allow save it.
Notes	The System registers with errors if you introduce the expiration date of the credit card in the past. For example : 01/2015 (MM/YYYY)

Check <#001E>	
Description	Running of the "Cancel" button
Expected	We expect the System come back to the before view.
Outcome	After clicking in this button, the System show us the correct view.
Notes	

Check <#001F>	
Description	Register with correct data
Expected	We expect the System to save our user account.
Outcome	After filling up all the fields, the System has perfectly saved the new author.
Notes	

Functional requirement <002>

Description

A user who is not authenticated must be able to register to the system as an organiser.

Access

From the main view, is possible to click in the link that takes us to the register view.

Checks

Check <#002A>	
Description	Register with the fields empty
Expected	We expect the System to catch and show us the errors.
Outcome	After keeping the fields empty, we click on the button "Save" and of course the System doesn't save the new organiser.
Notes	

Check <#002B>	
Description	Register with a birth date in the future
Expected	We expect the System to catch and show us the errors.
Outcome	After filling all the fields, we have established the birth date in the future and the System doesn't allow save it.
Notes	

Check <#002C>	
Description	Register with an invalid credit card number
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have filled the credit card number with an invalid number so the System doesn't allow save it.
Notes	

Check <#002D>	
Description	Register with the expiration date of the credit card in the past.
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the expiration date in the past so the System doesn't allow save it.
Notes	The System registers with errors if you introduce the expiration date of the credit card in the past. For example : 01/2015 (MM/YYYY)

Check <#002E>	
Description	Running of the "Cancel" button
Expected	We expect the System come back to the before view.
Outcome	After clicking in this button, the System show us the correct view.
Notes	

Check <#002F>	
Description	Register with correct data
Expected	We expect the System to save our user account.
Outcome	After filling up all the fields, the System has perfectly saved the new organiser.
Notes	

Functional requirement <003>

Description

A user who is not authenticated must be able to list the contest that the system stores.

Access

From the main view, is possible to click in the link that takes us to the contest's list.

Checks

Check <#003A>	
Description	List the contest stored in the System
Expected	We expect the System show us the contest's list.
Outcome	The contest's list is correctly shown.
Notes	

Functional requirement <004>

Description

A user who is not authenticated must be able to see the list of essays that have been published in a contest and see their details.

Access

From the main view, is possible to click in the link that takes us to the contest's list, where we click in the link that takes us to their respective essays. In this view, we can click again in the link that will show us the details of each essay selected.

Checks

Check <#004A>	
Description	Listing the essays of a contest and their details
Expected	We expect the System show us the list of essays and we can surf to their details.
Outcome	The contests list is correctly shown, but only show us the own essays, so is possible to access to the details.
Notes	Authors must being separated to the existing essays. Also it must show us the own essays.

Functional requirement <005>

Description

A user who is authenticated as an author must be able to do the same as a user who is not authenticated, except for registering to the system.

Access

From the main view, is possible to click in the link that takes us to the contest's list, where we can access to the respective essay. We can show the details of each essay.

Checks

Check <#005A>	
Description	Listing the contest of the System
Expected	We expect the System show us the contest's list.
Outcome	The general contests list isn't shown, only the list with the contest of the own autor.
Notes	

Check <#005B>	
Description	Listing the essays of a contest and their details
Expected	We expect the System show us the list of essays and we can surf to their details.
Outcome	The contests list is correctly shown and is possible to access to the details.
Notes	It must show us all the published essays of a contest, not all the published essays.

Functional requirement <006>

Description

A user who is authenticated as an author must be able to submit an essay, which he or she must have written, to a contest.

Access

From the main view, is possible to access to the creation essay.

Checks

Check <#006A>	
Description	Register with the fields empty
Expected	We expect the System to catch and show us the errors.
Outcome	After keeping the fields empty, we click on the button “Save” and of course the System doesn’t save the new essay.
Notes	

Check <#006B>	
Description	Running of the “Cancel” button
Expected	We expect the System come back to the before view.
Outcome	After clicking in this button, the System show us the correct view.
Notes	

Check <#006C>	
Description	Register with correct data
Expected	We expect the System to save our essay.
Outcome	After filling all the fields, the System has perfectly saved the new essay.
Notes	

Functional requirement <007>

Description

A user who is authenticated as an author must be able to edit an essay as long as the deadline of the corresponding contest has not expired.

Access

From the list of essays, we can enter in the edition view.

Checks

Check <#007A>	
Description	Editing with all the fields in blank
Expected	We expect the System to catch and show us the errors.
Outcome	After keeping the fields empty, we click on the button “Save” and of course the System doesn’t save the new essay.
Notes	

Check <#007B>	
Description	Checking that an essay which contest has an expired deadline can’t be edited
Expected	We expect the System doesn’t allow us to edit the essay.
Outcome	The essays whose contests with a deadline that hasn’t expired yet, will be accompany by an “Edit” link.
Notes	

Check <#007C>	
Description	Register with correct data
Expected	We expect the System to save our essay.
Outcome	After filling all the fields, the System has perfectly saved the new essay.
Notes	

Functional requirement <008>

Description

A user who is authenticated as an author must be able to list the essays that he or she’s submitted to a given contest

Access

From the list of essays, we can enter in the edition view.

Checks

Check <#008A>	
Description	Listing the submitted essays
Expected	We expect the System list the respective essays.
Outcome	The System show us correctly the essays that an user has submitted to a contest.
Notes	

Functional requirement <009>

Description

A user who is authenticated as an author must be able to list the essays that he or she’s submitted to a contest including the information regarding the public sessions in which they must be read, if applicable.

Access

From the list of contest, we can enter in the submitted contest view thanks to a link.

Checks

Check <#009A>	
Description	Listing the related public sessions to an essay
Expected	We expect the System list the related public sessions.
Outcome	The System show us correctly the essays that an user has submitted to a contest.
Notes	We can't see the public sessions in the details of the essays.

Functional requirement <010>

Description

A user who is authenticated as an organiser must be able to do the same as a user who is not authenticated, except for registering to the system.

Access

From the main view, is possible to get the contest's list, where we can access to the related essays. We can show the details of the essays.

Checks

Check <#010A>	
Description	Listing the contest of the System
Expected	We expect the System show us the contests list.
Outcome	It doesn't show us the general contest's list, only the specific list of the organiser.
Notes	

Check <#010B>	
Description	Listing the essays of a contest and see their details
Expected	We expect the System show us the list of essays and we can surf to their details.
Outcome	The contests list is correctly shown and is possible to access to the details.
Notes	It must show us all the published essays of a contest, not all the published essays.

Functional requirement <011>

Description

A user who is authenticated as an organiser must be able to create and edit contests. Deadlines and holding dates may be delayed when editing a contest, but they can only be set ahead the original dates.

Access

From the main view, is possible to get the creation contest view.

Checks

Check <#011A>	
Description	Register with the fields empty
Expected	We expect the System to catch and show us the errors.
Outcome	After keeping the fields empty, we click on the button “Save” and of course the System doesn’t save the new contest.
Notes	

Check <#011B>	
Description	Register with the holding date in the past
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the holding date in the past and the System can’t to save it.
Notes	

Check <#011C>	
Description	Register with the deadline in the past
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the deadline in the past and the System can’t save it.
Notes	

Check <#011D>	
Description	Register with a holding date before to the deadline
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the holding date before to the deadline and the System can’t save it.
Notes	

Check <#011E>	
Description	Running of the “Cancel” button
Expected	We expect the System come back to the before view.
Outcome	After clicking in this button, the System show us the correct view.
Notes	

Check <#011F>	
Description	Register with correct data
Expected	We expect the System to save our contest.
Outcome	After filling up all the fields, the System has perfectly saved the new contest.
Notes	

Check <#011G>	
Description	Editing with all the fields empty
Expected	We expect the System to catch and show us the errors.
Outcome	After keeping the fields empty, we click on the button “Save” and of course the System doesn’t save the new contest.
Notes	

Check <#011H>	
Description	Editing an existing contest with the holding date before to the actual date
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the holding date before to the actual date, and the System catch the errors and doesn’t save it.
Notes	

Check <#011I>	
Description	Editing an existing contest with the deadline before to the actual date
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established the deadline before to the actual date, and the System catch the errors and doesn’t save it.
Notes	

Check <#011J>	
Description	The System can’t allow to edit an essay which contest has been ending
Expected	We expect the System not to allow us to edit the essay.
Outcome	In the list of essays, if the contest has already expired, then it will appear a message “Expired” instead of the link of the edition list.
Notes	

Check <#011K>	
Description	Running of the “Cancel” button (Back to my contest)
Expected	We expect the System come back to the before view.
Outcome	After clicking in this button, the System show us the correct view.
Notes	

Functional requirement <012>

Description

A user who is authenticated as an organiser must be able to delete one of the contests he or she organises, as long as no author has already submitted any essays to it.

Access

From the main view, is possible to get the creation contest view.

Checks

Check <#012A>	
Description	Deleting of a empty contest
Expected	We expect the System allows to delete the contest.
Outcome	The System delete correctly the contest.
Notes	

Check <#012B>	
Description	Deleting of a none empty contest
Expected	We expect the System not allow to delete the contest.
Outcome	The System shows us a message saying that it can't be deleted the selected contest due to that contest has already essays.
Notes	

Functional requirement <013>

Description

A user who is authenticated as an organiser must be able to register the results of contests. The results are written as a piece of text in plain natural language. The results can be registered as soon as the holding date has expired.

Access

From the contest's list view of an organiser we can Access to the interface to add results.

Checks

Check <#013A>	
Description	Adding result to a expired contest
Expected	We expect the System allows us to add a result.
Outcome	The System save correctly the result introduced by the user for a selected contest.
Notes	

Check <#013B>	
Description	Adding the result to a contest that hasn't expired yet
Expected	We expect the System not allow us to add a result to the contest.
Outcome	The System only allow us to add result to expired contest, if they haven't no result.
Notes	

Functional requirement <014>

Description

A user who is authenticated as an organiser must be able to list the contests that he or she's created.

Access

From the main view, is possible to get the contest's .

Checks

Check <#014A>	
Description	Listing the corresponding contest of our organiser
Expected	We expect the System return us the contests list.
Outcome	The System returns the list.
Notes	

Functional requirement <015>

Description

A user who is authenticated as an organiser must be able to decide on whether an essay must be published or not. The decision can be made only after the hold date of the corresponding contest has expired.

Access

From the main view, is possible to get the view of contest with all the essays that can be published. From there, we can access to the essays that we want to publish. Clicking in a button we will check it as published.

Checks

Check <#015A>	
Description	Publishing an essay
Expected	We expect the System allows us to publish an essay.
Outcome	The System allow to publish an essay.
Notes	It can't be prove because of we can't know the organiser of each essay.

Functional requirement <016>

Description

A user who is authenticated as an organiser must be able to add another organiser to the contests that he or she organises

Access

From the main view, is possible to get the contest view, then we can access to a view for adding an organize.

Checks

Check <#016A>	
Description	Adding an organiser to a contest
Expected	We expect the System allows to add an organiser.
Outcome	The System allows to add an organiser, so when the organiser log in the System he will see that he is the owner of that contest.
Notes	

Check <#016B>	
Description	Adding himself as an additional organiser for a contest
Expected	We expect the System catch the error and not allows us to save the contest.
Outcome	The System not allows us to join ourself, because of the authenticated organiser isn't in the list.
Notes	

Check <#016C>	
Description	Running of the "Cancel" button
Expected	We expect the System come back to the before view.
Outcome	We can't prove it because of there isn't any "Cancel" button.
Notes	We can't prove it because of there isn't any "Cancel" button.

Functional requirement <017>

Description

A user who is authenticated as an administrator must be able to do the same as a user who is not authenticated, except for registering to the system.

Access

From the main view, is possible to get the contest's list view.

Checks

Check <#017A>	
Description	Listing all the contests stored in the System
Expected	We expect the System show us the contests list.
Outcome	It show us the list with all the contest of the System.
Notes	

Functional requirement <018>

Description

A user who is authenticated as an administrator must be able to register a new administrator of the system.

Access

From the main menú, is possible to register a new administrator.

Checks

Check <#018A>	
Description	Register with all the fields empty
Expected	We expect the System to catch and show us the errors.
Outcome	After keeping all the fields empty, we try to save it and obviously the System doesn't save the author.
Notes	

Check <#018B>	
Description	Register with an invalid email
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established an invalid email and the System didn't save it.
Notes	

Check <#018C>	
Description	Register with an invalid phone number
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established an invalid phone number and the System didn't save it.
Notes	

Check <#018D>	
Description	Register with an invalid home page
Expected	We expect the System to catch and show us the errors.
Outcome	After filling up all the fields, we have established an invalid home page and the System didn't save it.
Notes	The System didn't show us the error and saves it, this shouldn't happen. You must put the constraint URL.

Check <#018F>	
Description	Register with correct data
Expected	We expect the System to save our administrator.
Outcome	After filling all the fields, the System has perfectly saved the new administrator.
Notes	

Functional requirement <019>

Description

A user who is authenticated as an administrator must be able to see the list of the authors and organisers and see their details.

Access

From the main menu, is possible to get the list of authors and organisers also the details of each one.

Checks

Check <#019A>	
Description	Listing the authors and see their details
Expected	We expect the System show the author's list and see their details.
Outcome	It shows us the author's list of the System and their details.
Notes	

Check <#019B>	
Description	Listing the organiser and see their details
Expected	We exoect the System show us the organisers list and see their details.
Outcome	It shows us the organisers list of the System and their details.
Notes	

Functional requirement <020>

Description

A user who is authenticated as an administrator must be able to display the dashboard.

Access

From the main, is possible to get the Dashboard for the actual administrator logged.

Checks

Check <#020A>	
Description	Contests list ordered by descending number of submitted essays
Expected	We expect the System shows us the contests list ordered by descending number of submitted essays.
Outcome	It shows us the contests list with descendent order.
Notes	It shows us the contests list ordered by the number of submitted essays, but it doesn't show the number of these essays ,so we can't know neither if the list is really ordering nor we can order it.

Check <#020B>	
Description	The authors list who have more submitted essays
Expected	We expect the System shows us the authors list who have more submitted essays.
Outcome	It shows us the authors list who have more submitted essays.
Notes	It shows the authors list but not the number of submitted essays.

Check <#020C>	
Description	The authors list who have more published essays.
Expected	We expect the System shows us the authors list who have more published essays.
Outcome	It shows us the authors list who have more published essays.
Notes	It shows the authors list but not the number of published essays.

Check <#020D>	
Description	The authors list who have less published essays.
Expected	We expect the System shows us the authors list who have less published essays.
Outcome	It shows us the authors list who have less published essays.
Notes	It shows the authors list but not the number of published essays.

Check <#020E>	
Description	The average of the essays sent by an author
Expected	We expect the System shows us the average of the number of essays sent by an author.
Outcome	It shows us the average of the number of essays sent by an author.
Notes	

Check <#020F>	
Description	The average of the number of contest organized by an organiser
Expected	We expect the System shows us the average of the number of contest organized by an organiser.
Outcome	It shows the average of the number of contest organized by an organiser.
Notes	

Check <#020G>	
Description	The contest list that took place the last month
Expected	It's expected that the system display the contest list that took place the last month.
Outcome	The system displays the contest list that took place the last month.
Notes	

Check <#020H>	
Description	The list of contests with their public sessions, in descending order of number of essays that have been submitted.
Expected	It's expected that the system display the list of contests with their public sessions, in descending order of number of essays that have been submitted.
Outcome	The system displays the list of contests with their public sessions, in descending order of number of essays that have been submitted.
Notes	

Check <#020I>	
Description	The list of organisers who have to be chairmen, including the number of public sessions to chair.
Expected	It's expected that the system display the list of organisers who have to be chairmen, including the number of public sessions to chair.
Outcome	The system displays the list of organisers who have to be chairmen, including the number of public sessions to chair.
Notes	

Check <#020J>	
Description	The public sessions with an assigned chairman in descending order of the session capacity.
Expected	It's expected that the system display the public sessions with an assigned chairman in descending order of the session capacity.
Outcome	The system displays the public sessions with an assigned chairman in descending order of the session capacity.
Notes	

Additional checks

In this section we are going to give a few tips which could improve the original application.

Check <#021>	
Description	Telephone numbers
Expected	It's expected the application to allow us to introduce any type of number, international ones included, for example: +34
Outcome	The application must allow us to introduce any type of number.
Notes	We recommend they use a text field without restrictions.

Check <#022>	
Description	Localhost problems
Expected	Due to the use of localhost sometimes cancel buttons do not work.
Outcome	Due to the use of localhost sometimes cancel buttons do not work.
Notes	Due to the use of localhost sometimes cancel buttons do not work.

Check <#023>	
Description	Use of hours and minutes in dates
Expected	Use of hours and minutes in some dates.
Outcome	You should use of hours and minutes in some dates.
Notes	You should use of hours and minutes in some dates, for example, in HoldingDate.

Check <#024>	
Description	App icon change
Expected	The same app icon in all the views.
Outcome	You should use the same app icon in all the views.
Notes	Sometimes the app icon changes between some users or actions.

Check <#025>	
Description	Contest misspelled
Expected	Correct the translations when creating a new contest.
Outcome	You should correct the translation when creating a new contest.
Notes	

Check <#026>

Description	App hacking
Expected	The app should not allow users to modify the behavior with html tags.
Outcome	The app should not allow users to modify the behavior with html tags.
Notes	For example, you should not be allowed to use <h1>name</h1> as your user name.

The screenshot shows a web browser window with the URL `localhost/contest/organiser/my-contests.do`. The page header includes the logo for 'Sample Co., Inc.' and navigation links: CONTEST, ESSAY, PUBLIC SESSION, PROFILE (ORGANISER1), and PRIVACY AND COOKIES. Below the header, the page title is 'Contest list'. A message states '5 items found, displaying all items.' followed by a table of contests. A red arrow points to the row for 'hashjgdk'.

Name	Description	Holding Date	Deadline Date			Display
Contest 1	description1	2015-04-04	2015-04-03	Public Seasons	Add organiser	<input type="button" value="Display"/>
Contest 2	description2	2015-05-05	2015-05-04	Public Seasons	Add organiser	<input type="button" value="Display"/>
Contest 3	description3	2012-07-07	2012-07-06	Public Seasons	Add organiser	<input type="button" value="Display"/>
Contest 4	description4	2015-06-06	2015-06-05	Public Seasons	Add organiser	<input type="button" value="Display"/>
hashjgdk	hashjgdk	2015-07-17	2015-06-17	Public Seasons	Add organiser	<input type="button" value="Display"/>

Below the table is a 'Create' button. The footer contains the copyright notice 'Copyright © 2015 AcmeEssay, Inc.' and a taskbar at the bottom showing the Start button, taskbar icons, and the system clock at 11:46 PM.