Acceptance tests   
Acme Rookies v2.0

In this document we provide all acceptance tests required in D04 – Acceptance testing lesson. For the Acme Hacker-Rank and Acme-Rookies projects.

|  |  |
| --- | --- |
| ****Development team**** | |
| **Identifier** | G24 |
| **Members** | Parra, Juan;  Bermudo, Miguel;  Aguilar, Gonzalo;  Macías, Antonio;  Giraldez, Rafael |
| **Testing team** | |
| **Identifier** | G20 |
| **Members** | Candelario, Luis;  Carrasco, Antonio;  Gil, Luis;  Márquez, Francisco;  Morales, Sergio;  Moreno, Juan Manuel |
| **Indexing data** | |
| **Test designers** | Parra, Juan;  Bermudo, Miguel; |
| **Testers** | Candelario, Luis;  Morales Moreno, Sergio; |
| **Notes** |  |

Table of contents

[Use case UC7.1: Register to the system as a Company or a Hacker 4](#_Toc8578679)

[Use case UC7.2: List positions and navigate to the companies 10](#_Toc8578680)

[Use case UC7.3: List companies and navigate to the positions 12](#_Toc8578681)

[Use case UC7.4: Search for a position 13](#_Toc8578682)

[Use case UC8.2: Edit personal data 15](#_Toc8578683)

[Use case UC9.1: Manage the positions 22](#_Toc8578684)

[Use case UC9.2: Manage their problems 31](#_Toc8578685)

[Use case UC9.3: Manage the applications to their positions 36](#_Toc8578686)

[Use case UC10.1: Manage his or her applications 40](#_Toc8578687)

[Use case UC11.1: Create new administrators 44](#_Toc8578688)

[Use case UC11.2: Display a dashboard (Level C) 47](#_Toc8578689)

[Use case UC14: Customize system at run time 48](#_Toc8578690)

[Use case UC17.1 Manage his or her curricula 49](#_Toc8578691)

[Use case UC17.2 Manage his or her finder 61](#_Toc8578692)

[Use case UC18.1: Display a dashboard (Level B) 64](#_Toc8578693)

[Use case UC23.1 Manage his or her social profiles 66](#_Toc8578694)

[Use case UC23.2 Manage his or her messages 69](#_Toc8578695)

[Use case UC24.1 Broadcast a notification message 73](#_Toc8578696)

[Use case UC24.2 Launch the spammer or not-spammers process 74](#_Toc8578697)

[Use case UC24.3 Ban an actor with the spammer flag 75](#_Toc8578698)

[Use case UC24.4 Unban an actor with the spammer flag 76](#_Toc8578699)

[Additional tests 78](#_Toc8578700)

[Acme-Rookies v2.0 81](#_Toc8578701)

[Use case UC3: Self-assign a position to audit it & Manage his or her audits 81](#_Toc8578702)

[Use case UC4.1: Run a procedure to notify the existing users of the rebranding 85](#_Toc8578703)

[Use case UC4.2: Create user accounts for new auditors. 86](#_Toc8578704)

[Use case UC4.3: Launch a process to compute an audit score for every Company. 89](#_Toc8578705)

[Use case UC4.4 Display a dashboard (Level C) 90](#_Toc8578706)

[Use case UC9.1: Browse Providers & Navigate to their items. 92](#_Toc8578707)

[Use case UC9.2 Browse the list of items and navigate to their providers 93](#_Toc8578708)

[Use case UC9.3: Register to the system as a provider 94](#_Toc8578709)

[Use case UC10.1 Manage his or her catalogue of items 98](#_Toc8578710)

[Use case UC11.1 Display a dashboard (Level B) 102](#_Toc8578711)

[Use case UC13.1 Manage his or her sponsorships 103](#_Toc8578712)

[Use case UC14.1 Display a dashboard (Level A) 107](#_Toc8578713)

[Additional tests 108](#_Toc8578714)

# Use case UC7.1: Register to the system as a Company or a Hacker

Description

An unauthenticated user wants to register to the system either as a company or as a Hacker, he selects which of the two suits him/her better, then selects that option in the dropdown menu on the register tab and finally proceeds to fill in the form with the correct data.

Access

1. Main menu -> register -> register a new hacker
2. Main menu -> register -> register a new company

Tests

|  |  |
| --- | --- |
| **Test <#1> Register Hacker** | |
| **Description** | The form must be filled in with the following data:  holder: “TEST”, Credit Card Make: “VISA/AMEX…”, Credit Card Number:”1234123412341234”, CVV:”123”, Expiration month:”1”, Expiration Year: “2020”, Username:”hackertest”, Password & Repeat password: “hackertest”, Name: “test”, Surname: “test test”, VAT%:”ASD12341234”, Photo: ”http://www.photo.com/test.jpg”, Email:”email@email.com”, Address:”test”, phone:”612345678”; |
| **Expected** | After clicking on the Submit button the system will take you back to the landing page and now you can login as a new hacker with the credentials hackertest/hackertest. |
| **Outcome** | Result: Correct  The form was filled with the provided data.  2019-05-07 (2)  After pressing submit, I was redirected to the main page. Then, I clicked on login and filled the form with “hackertest/hackertest” credentials and it logged with success. |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#1.1> Register Hacker** | |
| **Description** | The form must be empty but for email: ”test” and photo: ”test” |
| **Expected** | After clicking on the Submit button the system will display in the register view the following errors:  holder: “Must not be blank”, Credit Card Make: “Must not be blank”, Credit Card Number: ”Size must be from 16 to 16 & Does not match pattern”, CVV: ”cannot be null”, Expiration month & Expiration Year: “this is an expired date”, Username:” Must not be empty”, Password & Repeat password: “Must not be empty”, Name: “Must not be empty”, Surname: “Must not be empty”, VAT%:”Does not match patter”, Photo: ”Invalid URL”, Email:” Invalid email”, phone: ”Does not match pattern”; |
| **Outcome** | Result: Correct  The form was filled with the provided data and this was the outcome after pressing the “Submit” button:  2019-05-07 (4) |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2> Register Company** | |
| **Description** | The form must be filled in with the following data:  holder: “TEST”, Credit Card Make: “VISA/AMEX…”, Credit Card Number:”2345234523452345”, CVV:”123”, Expiration month:”1”, Expiration Year: “2020”, Username:”companytest”, Password & Repeat password: “companytest”, Name: “test”, Surname: “test test”, VAT%:”ASD12341234”, Photo: ”http://www.photo.com/test.jpg”, Email:”email@email.com”, Address: ”test”, phone:”612345678”, Commercial Name: “test”; |
| **Expected** | After clicking on the Submit button the system will take you back to the landing page and now you can login as a new company with the credentials companytest / companytest. |
| **Outcome** | Result: Correct  The form was filled with the provided data:  2019-05-07 (6)  After pressing the “Submit” button, I was redirected to the main page.  I clicked on login and filled the form with “companytest/companytest” credentials and it logged with success. |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2.1> Register Company** | |
| **Description** | The form must be empty but for email: ”test” and photo: ”test” |
| **Expected** | After clicking on the Submit button the system will display in the register view the following errors:  holder: “Must not be blank”, Credit Card Make: “Must not be blank”, Credit Card Number: ”Size must be from 16 to 16 & Does not match pattern”, CVV: ”cannot be null”, Expiration month & Expiration Year: “this is an expired date”, Username:” Must not be empty”, Password & Repeat password: “Must not be empty”, Name: “Must not be empty”, Surname: “Must not be empty”, VAT%:”Does not match patter”, Photo: ”Invalid URL”, Email:” Invalid email”, phone: ”Does not match pattern” commercial name: “Must not be blank”; |
| **Outcome** | Result: Correct  The form was filled with the provided data and this was the outcome after pressing the “Submit” button:  2019-05-07 (8) |
| **Notes** |  |

# 

|  |  |
| --- | --- |
| **Test <#2.2> Register Company** | |
| **Description** | The form must be filled in with the following data:  holder: “TEST”, Credit Card Make: “VISA/AMEX…”, Credit Card Number:”2345234523452345”, CVV:”123”, Expiration month:”1”, Expiration Year: “2020”, Username:”companytest”, Password: “companytest”, Repeat password: “companyerror”, Name: “test”, Surname: “test test”, VAT%:”ASD12341234”, Photo: ”http://www.photo.com/test.jpg”, Email:”email@email.com”, Address: ”test”, phone:”612345678”, Commercial Name: “test”; |
| **Expected** | After clicking on the Submit button the system will display in the register view a message to notify that the passwords don’t match |
| **Outcome** | Result: Incorrect  The form was filled with the provided data and this was the outcome after pressing the “Submit” button:    After this, I logged in to the system with “companytest/companytest” credentials with success. |
| **Notes** |  |

# Use case UC7.2: List positions and navigate to the companies

Description

An unauthenticated user wants to List the available positions and navigate to the companies that posted those positions.

Access

1. Main menu -> List of positions
2. Main menu -> List of positions -> Company1 (company’s username)

Tests

|  |  |
| --- | --- |
| **Test <#1> List of Positions** | |
| **Description** | Clicking the List of Positions |
| **Expected** | It returns the view with a table which displays all the positions on the system that are on final mode. |
| **Outcome** | Result: Correct  After pressing the “List of position” button on the menu bar this was the outcome:  asdf |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2> Navigate to the companies** | |
| **Description** | Clicking the name of the company’s username you want to navigate to. |
| **Expected** | It returns a view with the personal information supplied by the company and the social profiles it has. |
| **Outcome** | Result: Correct  After pressing the “company1” button on the first row this was the outcome:  2019-05-07 (12)  After pressing the “company2” button on the first row this was the outcome:  2019-05-07 (13) |
| **Notes** |  |

# Use case UC7.3: List companies and navigate to the positions

Description

An unauthenticated user wants to List the companies and navigate to the positions they published.

Access

1. Main menu -> List of companies
2. Main menu -> List of companies -> Positions

Tests

|  |  |
| --- | --- |
| **Test <#1> List of Companies** | |
| **Description** | Clicking the List of Companies |
| **Expected** | It returns the view with a table which displays all the companies on the system. |
| **Outcome** | Result: Correct  After pressing the “List of companies” button on the menu bar this was the outcome:  2019-05-07 (14) |
| **Notes** |  |

# Use case UC7.4: Search for a position

Description

An unauthenticated user wants to List the positions and then filter by a keyword that

must be contained in its title, its description, its profile, its skills, its technologies, or the name of the corresponding company.

Access

1. Main menu -> List of positions

Tests

|  |  |
| --- | --- |
| **Test <#1> keyword** | |
| **Description** | Clicking the List of Companies, Keyword: ”resurrect” |
| **Expected** | It returns to the same view but it should be only one entry on the table, the one corresponding to “RESU-9691” ticker. |
| **Outcome** | Result: Correct  The form was filled with the provided data and this was the outcome after pressing the “Submit” button:  2019-05-07 (16) |
| **Notes** |  |

# 

|  |  |
| --- | --- |
| **Test <#2> keyword ticker** | |
| **Description** | Clicking the List of Companies, Keyword: ”9691” |
| **Expected** | It returns to the same view but it should be only one entry on the table, the one corresponding to “RESU-9691” ticker. |
| **Outcome** | Result: Incorrect  The form was filled with the provided data and this was the outcome after pressing the “Search” button: |
| **Notes** |  |

# Use case UC8.2: Edit personal data

Description

An authenticated user wants to edit his or her personal data, to do so he or she must navigate to his/her profile and edit the personal data, credit card data or username and password

Access

1. Login-> Main menu -> profile -> Edit Personal Data
2. Login-> Main menu -> profile -> Edit CreditCard Data
3. Login-> Main menu -> profile -> Edit UserAccount Data

Tests

|  |  |
| --- | --- |
| **Test <#1> Edit Personal Data** | |
| **Description** | The form must be filled in with the following data:  Name: “test0”, Surname: “test0 test0”, VAT%:”ASD12341234”, Photo: ”http://www.photo.com/test.jpg”, Email:”email@email.com”, Address: ”test”, phone:”612345678”; |
| **Expected** | After clicking on the Submit button the system will take you back to the profile with the data you changed visible. |
| **Outcome** | Result: Correct  The form was filled with the provided data and this was the outcome after pressing the “Save” button: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#1.1> Edit Personal Data** | |
| **Description** | The form must be empty |
| **Expected** | After clicking on the Submit button the system will display in the register view the following errors:  Name: “Must not be empty”, Surname: “Must not be empty”, VAT%:”Does not match patter”, Photo: ”Must not Be empty”, Email: “ must not be empty”, phone: ”Does not match pattern”; |
| **Outcome** | Result: Correct  The form was filled with the provided data and this was the outcome after pressing the “Save” button: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#1.2> Edit Personal Data** | |
| **Description** | The form must be filled in with the following data:  Name: “test0”, Surname: “test0 test0”, VAT%:”ASD12341234”, Photo: ”photo”, Email:”email”, Address: ”test”, phone:”612345678”; |
| **Expected** | After clicking on the Submit button the system will display in the register view an error message to notify that the photo and the email doesn’t have the correct format |
| **Outcome** | Result: Incorrect  The form was filled with the provided data, the system saved the personal data and this was the outcome after pressing the “Save” button: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2> Edit CreditCard Data** | |
| **Description** | The form must be filled in with the following data:  holder: “TEST0”, Credit Card Make: “VISA/AMEX…”, Credit Card Number:”2345234523452345”, CVV:”123”, Expiration month:”1”, Expiration Year: “2020”; |
| **Expected** | After clicking on the Submit button the system will take you back to the profile with the data you changed visible. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2.1> Edit CreditCard Data** | |
| **Description** | The form must be empty, Expiration Month: “1”, Expiration Month: “2019”; |
| **Expected** | After clicking on the Submit button the system will display in the register view the following errors:  holder: “Must not be blank”, Credit Card Make: “Must not be blank”, Credit Card Number: ”Size must be from 16 to 16 & Does not match pattern”, CVV: ”cannot be null”, Expiration month & Expiration Year: “this is an expired date” ; |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** | The pattern of the expiration year contains a typo: “and”. |

|  |  |
| --- | --- |
| **Test <#3> Edit UserAccount Data** | |
| **Description** | The form must be filled in with the following data:  Username:”companytest1”, Password & Repeat password: “companytest1” |
| **Expected** | After clicking on the Submit button the system will ask for confirmation and then take you back to the landing page and now you can login as a new company with the credentials companytest1 / companytest1. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing submit I was redirected to the main page. Then I logged into the system with “companytest1/ companytest1” credentials and it logged with success. |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#3.1> Edit UserAccount Data** | |
| **Description** | The form must be empty, Password: ”pass1” Repeat password: “pass2” |
| **Expected** | After clicking on the Submit button the system will display in the register view the following errors:  Username:” Must not be empty”, Password & Repeat password: “passwords do not match” |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

# Use case UC9.1: Manage the positions

Description

1. A Company wants to list their positions, so they login and click on list of positions, and list my positions
2. A company wants to create a new position, from step number 1 click “create new position”
3. A company wants to show a position, from step number 1 click show on the position they want to show
4. A company wants to update a position, from step number 1 click edit on the position they want to edit.
5. A company wants to delete a position, from step number 1 click on delete on the position they want to delete

Access

1. Login -> Main menu -> List of positions -> List my positions
2. Login -> Main menu -> List of positions -> List my positions -> Create new position
3. Login -> Main menu -> List of positions -> List my positions -> Show
4. Login -> Main menu -> List of positions -> List my positions -> Edit
5. Login -> Main menu -> List of positions -> List my positions -> Delete

Tests

|  |  |
| --- | --- |
| **Test <#1> List the positions** | |
| **Description** | Login then click on list my positions from the list of positions dropdown menu |
| **Expected** | It returns a view with all the positions that belong to a company. |
| **Outcome** | Result: Correct  After pressing the “List my position” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2> Create a position** | |
| **Description** | Click on Create a new position then fill out the form with the following data:  Title: ”Test”, Description: ”Test”, Deadline: “31/12/2019 00:00”, Profile: ”test” , Skills: “test”, Technologies “test, test”, Salary:”1000” |
| **Expected** | It returns a view with all the positions that belong to a company including the one you just created with the options to show edit and delete it. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2.1> Create a position** | |
| **Description** | Click on Create a new position, send the form empty |
| **Expected** | It returns the following errors:  Title: ”Must not be Blank”, Description: ”Must not be Blank”, Deadline: “Cannot be null”, Profile:” Must not be Blank” , Skills: “Must not be Blank”, Technologies “Must not be Blank”, Salary: ”cannot be null”  And a commit error |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#3> Show a position** | |
| **Description** | Click on show on the position you want to display |
| **Expected** | It returns a view with all the information relative to a position. |
| **Outcome** | Result: Correct  This was the outcome after clicking on the “Show” link: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#4> Edit a position** | |
| **Description** | Click on edit on the position you want to edit then fill out the form with the following data:  Title: ”Test1”, Description: ”Test1”, Deadline: “31/12/2020 00:00”, Profile: ”test1” , Skills: “test1”, Technologies “test1, test1”, Salary:”1500”, is final: “unckecked”; |
| **Expected** | It returns a view with all the positions, it should contain the updated information on the position you edited and all the options avaliable |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#4.1> Edit a position** | |
| **Description** | Click on edit on the position you want to edit then fill out the form with the following data:  Title: ”Test2”, Description: ”Test2”, Deadline: “31/12/2020 00:00”, Profile: ”test2” , Skills: “test2”, Technologies “test2, test2”, Salary:”2500”, is final: “checked”; |
| **Expected** | Case1, the position has 2 problems linked to it:  It returns a view with all the positions, it should contain the updated information on the position you edited and ONLY show available.  Case2, the position doesn’t have 2 problems linked to it:  It returns an error that reads: You can't publish this position, it needs to have at least 2 problems associated. |
| **Outcome** | Result both cases: Correct  Case1: The first position of the list only allows to show it    Case2: The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

\*To see how to add problems to a position see use case 9.2

|  |  |
| --- | --- |
| **Test <#4.2> Edit a position** | |
| **Description** | Click on edit on the position you want to edit, send the form empty |
| **Expected** | It returns the view with a commit error. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#4.3> Edit a position** | |
| **Description** | Login as company1, edit position with name “Position 3 programmacks”, copy the link, and as company2 try to modify the position and save it |
| **Expected** | It returns the view with a commit error. |
| **Outcome** | Result: Incorrect  The form was filled with arbitrary data and after pressing the “Save” button the company1 position saved and this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#5> Delete a position** | |
| **Description** | Click on delete on the position you want to delete |
| **Expected** | It returns the view with all the positions, but the one you just chose to delete. |
| **Outcome** | Result: Correct  This was the outcome after clicking on “Delete” link of a position: |
| **Notes** |  |

# Use case UC9.2: Manage their problems

Description

1. A company wishes to list their problems; they login into the system and finally clicks on “List of problems” on the menu.
2. A company wishes to show their problems; they login into the system, clicks on “List of problems” on the menu, and finally clicks on the “Show” link of the problem to show.
3. A company wishes to create a problem; they login into the system, clicks on “List of problems” on the menu, then clicks on the “Create problem” link, fills the formulary, and finally clicks on the “Save” button.
4. A company wishes to edit a problem; they login into the system, clicks on “List of problems” on the menu, then clicks on the “Edit” link on the problem to edit, fills de formulary, and finally clicks on “Save” button.
5. A company wishes to delete a problem; they login into the system, clicks on “List of problems” on the menu, and finally clicks on the “Delete” link on the problem to delete.

Access

1. Main Menu -> List of problems
2. Main Menu -> List of problems -> Show
3. Main Menu -> List of problems -> Create -> Fills form -> Save button
4. Main Menu -> List of problems -> Edit -> Fills form -> Save button
5. Main Menu -> List of problems -> Delete

Test

|  |  |
| --- | --- |
| **Test <#001> List problems** | |
| **Description** | Click on the “List of problems” link on the main menu |
| **Expected** | It returns the view with a table which displays all the problems that the company has |
| **Outcome** | Result: Correct  After pressing the “List of problems” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#002> Show problem** | |
| **Description** | Click on the “List of problems” link on the main menu, and then on the “Show” link displayed on the row of the corresponding problem to display |
| **Expected** | It returns the view with the information of the corresponding problem and a table with the related positions |
| **Outcome** | Result: Correct  This was the outcome after clicking on the “Show” link of a problem from the list: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#003> Create problem** | |
| **Description** | It returns the form that must be filled with the following data: title: ‘Test’, Hint: ‘test’, Statement: ’test’, attachments: ‘http://www.test.com’, positions: ‘Position 1 programacks” |
| **Expected** | After clicking on the Save button, it returns to the list where the created problem appears |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#004> Edit problem** | |
| **Description** | It returns the form that must be filled changing with the following data: title: ‘Test2’ |
| **Expected** | After clicking on the Save button, it returns to the list where the edited problem appears with the new title ‘Test2’ |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#004.1> Edit problem** | |
| **Description** | It returns the form that must be filled changing with the following data: title: empty, statement: empty |
| **Expected** | After clicking on the Save button, it returns to the form showing the “Must not be blank” error on the fields of title and statement and “Must not be empty” error in Attatchments. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#004.2> Edit problem** | |
| **Description** | Login as company1, edit problem with name “Title problem 1”, copy the link, and as company2 try to modify the problem and save it |
| **Expected** | It returns the view with a commit error. |
| **Outcome** | Result: Incorrect  The form was filled with arbitrary data and after pressing the “Save” button the company1 problem saved and this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#005> Delete problem** | |
| **Description** | Click on the delete link of the corresponding row of the problem to test |
| **Expected** | It returns the list of problems without the problem deleted |
| **Outcome** | Result: Correct  This was the outcome after delete the position with title “Title Problem 8”: |
| **Notes** |  |

# Use case UC9.3: Manage the applications to their positions

Description

1. A company wishes to list the applications to their positions; they login into the system and finally clicks on “List of applications” on the menu.
2. A company wishes to show the applications to their positions; they login into the system, clicks on “List of applications” on the menu, and finally clicks on the “Show” link of the application to show.
3. A company wishes to update an application to their problem; they login into the system, clicks on “List of applications” on the menu, then clicks on the “Edit” link of the application to update, choose the new status, and finally clicks on the “Save” button.

Access

1. Main Menu -> List of positions
2. Main Menu -> List of positions -> Show
3. Main Menu -> List of positions -> Edit -> Fills form -> Save button

Test

|  |  |
| --- | --- |
| **Test <#001> List applications** | |
| **Description** | Click on the “List of applications” link on the main menu |
| **Expected** | It returns the view with a table which displays all the applications to their positions that the company has, grouped by status |
| **Outcome** | Result: Correct  After pressing the “List of applications” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#002> Show application** | |
| **Description** | Click on the “List of applications” link on the main menu, and then on the “Show” link displayed on the row of the corresponding application to display |
| **Expected** | It returns the view with the information of the corresponding application, and the information of the position and problem related |
| **Outcome** | Result: Correct  This was the outcome after clicking on the “Show” link of one application from the list: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#003> Update application** | |
| **Description** | It returns the form that must be filled with the following data: status: ‘ACCEPTED’ |
| **Expected** | After clicking on the Save button, it returns to the list where the updated application appears |
| **Outcome** | Result: Correct  Before updating the “application” on the table this was the outcome:    After, the form was filled with provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

# 

|  |  |
| --- | --- |
| **Test <#003.1> Update application** | |
| **Description** | Login as company2, edit the application with status “SUBMITTED”, copy the link, and as company1 try to modify the application and save it |
| **Expected** | It returns the view with a commit error |
| **Outcome** | Result: Incorrect  The form was filled with arbitrary data and after pressing the “Save” button the company1 position saved and this was the outcome:    If we list again the company 2 applications, we can see that the application status has changed by company 1: |
| **Notes** |  |

# Use case UC10.1: Manage his or her applications

Description

1. A hacker wishes to list his or her applications grouped by status; he or she login into the system and finally clicks on “List of applications” on the menu.
2. A hacker wishes to show his or her applications; he or she login into the system, clicks on “List of applications” on the menu, and finally clicks on the “Show” link of the application to show.
3. A hacker wishes to create a new application; he or she login into the system, clicks on “List of positions” on the menu, then clicks on the “Apply” link on the corresponding position that he or she wants to apply.
4. A hacker wishes to update his or her application; he or she login into the system, clicks on “List of applications” on the menu, then clicks on the “Edit” link of the application to update, fills the form, and finally clicks on the “Save” button.

Access

1. Main Menu -> List of applications
2. Main Menu -> List of applications -> Show
3. Main Menu -> List of positions -> Apply
4. Main Menu -> List of positions -> Edit -> Fills form -> Save button

Test

|  |  |
| --- | --- |
| **Test <#001> List applications** | |
| **Description** | Click on the “List of applications” link on the main menu |
| **Expected** | It returns the view with a table which displays all the applications that the hacker has, grouped by status |
| **Outcome** | Result: Correct  After pressing the “List of applications” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#002> Show application** | |
| **Description** | Click on the “List of applications” link on the main menu, and then on the “Show” link displayed on the row of the corresponding application to display |
| **Expected** | It returns the view with the information of the corresponding application, and the information of the position and problem related |
| **Outcome** | Result: Correct  This was the outcome after clicking on the “Show” link of an application from the list: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#003> Create application** | |
| **Description** | Click on the “List of positions” link on the main menu, and then on the “Apply” link displayed on the row of the corresponding position to apply |
| **Expected** | It returns the list of applications of the hacker, with the new that we have just created, with status “PENDING” |
| **Outcome** | Result: Correct  After clicking on the “apply” link of an application from the table this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#004> Update application** | |
| **Description** | It returns the form that must be filled with the following data: answer: ‘Test’, code link: ’http://www.test.com’, problem: ’Test’, curricula: ‘Curricula 1’  It is needed a published problem so the application can be updated |
| **Expected** | After clicking on the Save button, it returns to the list where the updated application appears |
| **Outcome** | Result: Correct  The form was filled with provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#004.1> Update application** | |
| **Description** | It returns the form that must be filled with the following data: answer: empty’, code link: empty, problem: empty, curricula: empty  It is needed a published problem so the application can be updated |
| **Expected** | After clicking on the Save button, it returns to the form with an error message on all the fields that needs to be filled |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

# Use case UC11.1: Create new administrators

Description

An administrator wishes to create a new administrator account; he or she login into the system, clicks on “Register a new admin” on the menu, fills the form and click on “Save” button

Access

Main Menu -> Register a new admin -> Fills the form -> Save

Test

|  |  |
| --- | --- |
| **Test <#001> Create new administrator** | |
| **Description** | Click on the “Register a new admin” link on the main menu, it returns the form that must be filled with the following data: holder: ‘test’ , make: ‘VISA’, number: ‘1231231231231231’, cvv: ‘339’, expiration month: ‘1’, expiration year: ‘2020’, username: ‘admintest’, password: ‘admintest’, repeat password: ‘admintest’, name: ‘admintest’, surnames: ‘admintest’, vat: ‘ESN80082090’, photo: ‘http://www.test.com’, email: ‘test@test.com’, address: ‘test’, phone: ‘666666666’ |
| **Expected** | After clicking on the “Submit” button, it returns to the welcome page |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Submit” button this was the outcome:    After this, I logged in to the system with “admintest/ admintest” credentials and this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#002> Create new administrator** | |
| **Description** | Click on the “Register a new admin” link on the main menu, it returns the form that must be filled with the following data: all the fields empty |
| **Expected** | After clicking on the “Submit” button, it returns to the form which displays all the error messages on the fields that we left empty |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Submit” button this was the outcome: |
| **Notes** |  |

# Use case UC11.2: Display a dashboard (Level C)

Description

An administrator wishes to display a dashboard with information of the system; he or she login into the system, and clicks on “Dashboard” on the “System” section of the menu.

Access

Main Menu -> System -> Dashboard

Test

|  |  |
| --- | --- |
| **Test <#001> Display dashboard** | |
| **Description** | Click on the “Dashboard” link on the “System” section of the main menu |
| **Expected** | It returns to a page with all the information displayed with tables |
| **Outcome** | Result: Correct  After pressing the “Dashboard” button on the menu bar all level c queries and its results appeared and this was the outcome:    All level c queries and its results appear. |
| **Notes** |  |

# Use case UC14: Customize system at run time

Description

An administrator wishes to be easy to customize the system at run time; he or she login into the system, clicks on “Configuration” on the “System” section of the menu, fills the form, and clicks on the “Save” button.

Access

Main Menu -> System -> Configuration -> Fills the form -> Save button

Test

|  |  |
| --- | --- |
| **Test <#001> Customize system** | |
| **Description** | Click on the “Configuration” link on the “System” section of the main menu, and fills the form with the information that he or she wants to customize |
| **Expected** | It returns to the welcome page so the admin can see the changes |
| **Outcome** | Result: Correct  The form was filled with arbitrary data after pressing submit, this was the outcome:    Also, we can see that the configuration saved properly: |
| **Notes** |  |

# Use case UC17.1 Manage his or her curricula

Description

1. A hacker wishes to list his or her curriculas; he or she login into the system and finally clicks on “My Curriculas” option in the menu.
2. A hacker wishes to show their curriculas; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list and finally curricula is shown.
3. A hacker wishes to show their curricula personal data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list and finally his or her curricula personal data is shown.
4. A hacker wishes to show their curricula position data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; displays a certain position data from the position data list and finally it is shown.
5. A hacker wishes to show their curricula education data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; displays a certain education data from the education data list and finally it is shown.
6. A hacker wishes to show their curricula miscellaneous data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; displays a certain miscellaneous data from the miscellaneous data list and finally it is shown.
7. A hacker wishes to create a curricula; he or she login into the system; clicks on “My Curriculas”, click on the ‘Create a new curricula’ button, fills the form, and clicks on “Save” button.
8. A hacker wishes to create a curricula position data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; clicks on ‘New position data’ button; fills in a form with curricula position data and finally he or she hits the ‘Save’ button.
9. A hacker wishes to create a curricula education data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; clicks on ‘New education data’ button; he or she fills in a form with curricula education data and finally he or she hits the ‘Save’ button.
10. A hacker wishes to create a curricula miscellaneous data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; clicks on ‘New miscellaneous data’ button; he or she fills in a form with curricula miscellaneous data and finally he or she hits the ‘Save’ button.
11. A hacker wishes to update a curricula personal data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; clicks on the ‘Edit’ link on the Position Data table; he or she fills in a form with new curricula personal data and finally he or she hits the ‘Save’ button.
12. A hacker wishes to update a curricula position data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; clicks on ‘Edit’ link of a position data item from position data list; he or she fills in a form with curricula position data and finally he or she hits the ‘Save’ button.
13. A hacker wishes to update a curricula education data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; clicks on ‘Edit’ link of an education data item from education data list; he or she fills in a form with curricula education data and finally he or she hits the ‘Save’ button.
14. A hacker wishes to update a curricula miscellaneous data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list; clicks on ‘Edit’ link of an miscellaneous data item from miscellaneous data list; he or she fills in a form with curricula miscellaneous data and finally he or she hits the ‘Save’ button.
15. A hacker wishes to delete a curricula position data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list and finally he or she clicks on ‘Delete’ link of a position data item from position data list.
16. A hacker wishes to delete a curricula education data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list and finally he or she clicks on ‘Delete’ link of an education data item from education data list.
17. A hacker wishes to delete a curricula miscellaneous data; he or she login into the system; clicks on “My Curriculas”; displays a certain curricula from the list and finally he or she clicks on ‘Delete’ link of a miscellaneous data item from miscellaneous data list.

Access

1. Login > Main Menu > My curriculas
2. Login > Main Menu > My curriculas > Show
3. Login > Main Menu > My curriculas > Show
4. Login > Main Menu > My curriculas > Show
5. Login > Main Menu > My curriculas > Show
6. Login > Main Menu > My curriculas > Show
7. Login > Main Menu > My curriculas > Create new curricula > Fills the form > Save
8. Login > Main Menu > My curriculas > Show > New position data > Fills the form > Save
9. Login > Main Menu > My curriculas > Show > Create education data > Fills the form > Save
10. Login > Main Menu > My curriculas > Show > Create miscellaneous data > Fills the form > Save
11. Login > Main Menu > My curriculas > Show > Edit (Personal data list) > Fills the form > Save
12. Login > Main Menu > My curriculas > Show > Edit (Position data list) > Fills the form > Save
13. Login > Main Menu > My curriculas > Show > Edit (Education data list) > Fills the form > Save
14. Login > Main Menu > My curriculas > Show > Edit (Miscellaneous data list) > Fills the form > Save
15. Login > Main Menu > My curriculas > Display (Curricula list) > Delete (Position data list)
16. Login > Main Menu > My curriculas > Display (Curricula list) > Delete (Education data list)
17. Login > Main Menu > My curriculas > Display (Curricula list) > Delete (Miscellaneous data list)

Tests

|  |  |
| --- | --- |
| **Test 001 – List curriculas** | |
| **Description** | The hacker’s curriculas must be listed. |
| **Expected** | The system must return the hacker curricula list. |
| **Outcome** | Result: Correct  After pressing the “My curriculas” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 002 – Show curricula** | |
| **Description** | A hacker’s curricula must be shown. |
| **Expected** | The system must return a hacker curricula full details view which includes its datas in tables |
| **Outcome** | Result: Correct  After clicking on “Show Curricula” link of one curricula from the list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 003 – Show curricula** | |
| **Description** | A hacker’s curricula personal data must be shown. |
| **Expected** | The system must display all curricula personal data details. |
| **Outcome** | Result: Correct  After clicking on “Show Curricula” link of one curricula from the list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 004 – Show curricula** | |
| **Description** | A hacker’s curricula position data must be shown. |
| **Expected** | The system must display all curricula position data details. |
| **Outcome** | Result: Correct  After clicking on “Show Curricula” link of one curricula from the list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 005 – Show curricula** | |
| **Description** | A hacker’s curricula education data must be shown. |
| **Expected** | The system must display all curricula education data details. |
| **Outcome** | Result: Correct  After clicking on “Show Curricula” link of one curricula from the list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 006 – Show curricula** | |
| **Description** | A hacker’s curricula miscellaneous data must be shown. |
| **Expected** | The system must display all curricula miscellaneous data details. |
| **Outcome** | Result: Correct  After clicking on “Show Curricula” link of one curricula from the list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 007 – Create curricula** | |
| **Description** | A hacker’s curricula must be created. |
| **Expected** | The system must list all hacker’s curriculas and the new curricula must appear on it. |
| **Outcome** | Result: Correct  After pressing the “Create a new Curricula” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 008 – Create curricula** | |
| **Description** | The form must be filled in with the following data: curricula’s name: ‘Test’, fullName:’Test’, statement: ‘I´m the best hacker in Europe’, Phone: ‘648456555’, GitHub profile: ‘https://www.github.com/profile123224’, LinkedIn profile: ‘https://www.linkedin.com/profile123224’. |
| **Expected** | The system must save the entity properly and return the curricula display view showing the new personal data. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 009 – Create position data** | |
| **Description** | The form must be filled in with the following data: Title: ‘Test’, Description: ‘Test’, Phone: ‘648456555’, Start date: ‘05/08/2013 12:00’, End date: ‘05/08/2018 12:00’. |
| **Expected** | The system must save the entity properly and return the curricula display view showing the new position data. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 010 – Create education data** | |
| **Description** | The form must be filled in with the following data: Degree: ‘Test’, Institution: ‘Test, Mark: ‘9’, Start date: ‘05/08/2013 12:00’, End date: ‘05/08/2018 12:00’. |
| **Expected** | The system must save the entity properly, return the curricula display view and the new education data must appear on education data list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 011 – Create miscellaneous data** | |
| **Description** | The form must be filled in with the following data: Text: ‘Test’, attachments: ‘http://www.test.com’. |
| **Expected** | The system must save the entity properly, return the curricula display view and the new miscellaneous data must appear on miscellaneous data list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 011.1 – Create miscellaneous data** | |
| **Description** | As hacker1, on the curricula with name “Curricula 1”, creating a new miscellaneous data, copy the link, logout and login as hacker2, access to the link and trying to create and save the data |
| **Expected** | It returns the view with a commit error. |
| **Outcome** | Result: Incorrect  The form was filled with the provided data and after pressing the “Save” button, the system saved the entity this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 012 – Create position data** | |
| **Description** | The form must be filled in with the following data: Title: ‘ ‘, Description: ‘Test’, Phone: ‘648456555’, Start date: ‘05/08/2013 12:00’, End date: ‘05/08/2018 12:00’ |
| **Expected** | The system must not save the entity properly and return the position data view showing the ‘Must not be blank’ message error in title. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 013 – Create education data** | |
| **Description** | The form must be filled in with the following data: Degree: ‘ ‘, Institution: ‘Test, Mark: ‘9’, Start date: ‘05/08/2013 12:00’, End date: ‘05/08/2018 12:00’. |
| **Expected** | The system must not save the entity properly and return the education data view showing the ‘Must not be blank’ message error in degree. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 014 – Create miscellaneous data** | |
| **Description** | The form must be filled in with the following data: Text: ‘’, attachments: ‘http://www.test.com’. |
| **Expected** | The system must not save the entity properly and return the miscellaneous data view showing the ‘Must not be blank’ message error in text. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 015 – Delete position data** | |
| **Description** | A hacker curricula position data must be deleted. |
| **Expected** | The system must return the curricula display view and the selected position data must not appear on position data list. |
| **Outcome** | Result: Correct  After clicking on “Delete link” of one position data from the list this was the outcome |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 016 – Delete education data** | |
| **Description** | A hacker curricula education data must be deleted. |
| **Expected** | The system must return the curricula display view and the selected education data must not appear on education data list. |
| **Outcome** | Result: Correct  After clicking on “Delete link” of one education data from the list this was the outcome |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 017 – Delete miscellaneous data** | |
| **Description** | A hacker’s curricula miscellaneous data must be deleted. |
| **Expected** | The system must return the curricula display view and the selected miscellaneous data must not appear on miscellaneous data list. |
| **Outcome** | Result: Correct  After clicking on “Delete link” of one miscellaneous data from the list this was the outcome: |
| **Notes** |  |

# Use case UC17.2 Manage his or her finder

Description

1. A hacker wishes to update his or her search criteria and lists its contents; he or she login into the system; clicks on “Finder”; he or she fills the finder form and finally he or she hits the ‘Search’ button.
2. A hacker wishes to clear its finder; he or she login into the system; clicks on “Finder” and finally he or she hits the ”Clear” button.

Access

1. Login > Main Menu > Finder > Search
2. Login > Main Menu > Finder > Clear

Tests

|  |  |
| --- | --- |
| **Test 021** | |
| **Description** | The form must be filled in with the following data: Key word: ‘resurrect’, Deadline:’’, Minimum salary:’’, Maximum salary:’’. |
| **Expected** | The system must return a position list with only two position (RESU-9691 & RESU-9692) due to search criteria provided, also the view should display a message about when were the results cached. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Search” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | The form must be filled in with the following data: Key word: ‘test, Deadline:’’, Minimum salary:’’, Maximum salary:’’. |
| **Expected** | The system must return an empty position list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Search” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 023** | |
| **Description** | The button ‘Clear’ must be hit |
| **Expected** | The system must return a position list with all available positions, there must be 6. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Search” button this was the outcome: |
| **Notes** |  |

# 

|  |  |
| --- | --- |
| **Test 024** | |
| **Description** | The form must be filled in with the following data: Key word: ‘’, Deadline:’’, Minimum salary:’’, Maximum salary:’600’. |
| **Expected** | The system must return a position list with only one position (PROG-3902) due to search criteria provided, also the view should display a message about when were the results cached. |
| **Outcome** | Result: Incorrect  The form was filled with the provided data and after pressing the “Search” button this was the outcome:    The param “Maximum salary are not detected” |
| **Notes** |  |

# Use case UC18.1: Display a dashboard (Level B)

Description

An administrator wishes to display a dashboard with information of the system; he or she login into the system, and clicks on “Dashboard” on the “System” section of the menu.

Access

Main Menu -> System -> Dashboard

Test

|  |  |
| --- | --- |
| **Test <#001> Display dashboard** | |
| **Description** | Click on the “Dashboard” link on the “System” section of the main menu |
| **Expected** | It returns to a page with all the information displayed with tables |
| **Outcome** | Result: Correct  After clicking on the “Dashboard” button on the menu bar all level b queries and its results appeared and this was the outcome: |
| **Notes** |  |

# Use case UC23.1 Manage his or her social profiles

Description

1. An authenticated user wishes to display his or her social profiles; he or she login into the system; clicks on ‘Profile (Username)’; then on ‘profile’ and all of his or her info is shown including the social profiles.
2. An authenticated user wishes to create a social profile; he or she login into the system; clicks on ‘Profile (Username)’; then on ‘profile’; clicks on ‘Create’ right under the Social profile title; he or she fills the form and finally hits the ‘Save’ button.
3. An authenticated user wishes to update a social profile; he or she login into the system; clicks on ‘Profile (Username)’; then on ‘profile’; clicks on ‘Edit’ link of the social profile list; he or she fills the form and finally hits the ‘Save’ button.
4. An authenticated user wishes to delete a social profile; he or she login into the system; clicks on ‘Profile (Username)’; then on ‘profile’; clicks on ‘’delete” link of the social profile list.

Access

1. Login > Main Menu > Profile > Profile
2. Login > Main Menu > Profile > Profile > Create social profile
3. Login > Main Menu > Profile > Profile > Edit (Social profile list)
4. Login > Main Menu > Profile > Profile > Delete (Social profile list)

Tests

|  |  |
| --- | --- |
| **Test 1** | |
| **Description** | An authenticated user social profiles must be listed in his/her profile. |
| **Expected** | The system must return an authenticated user social profiles list |
| **Outcome** | Result: Correct  After pressing the “Profile” button, this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 2** | |
| **Description** | The form must be filled in with the following data: Nick: ‘Tester43’, Social network: ‘Twitter’, Link to profile: ‘https://twitter.com/tester43’ |
| **Expected** | The system must save the entity properly, return the social profile list view and the new social profile must appear on the list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 2.1** | |
| **Description** | The form must be filled in with the following data: Nick: ‘Tester43’, Social network: ‘Twitter’, Link to profile: ‘twitter’ |
| **Expected** | The system must not save the entity and return the social profile edit view showing the ‘Invalid URL’ message error |
| **Outcome** | Result: Incorrect  The form was filled with the provided data and after pressing the “Save” button the system saved the entity so this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 3** | |
| **Description** | The form must be filled in with the following data: Nick: ‘test2’, Social network: ‘Instagram’, Link to profile: ‘https://www.instagram.com |
| **Expected** | The system must save the entity properly, return the social profile list view and the updated social profile must appear on the list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 4** | |
| **Description** | An authenticated user social profiles must be deleted |
| **Expected** | The system must return the social profile list view and the social profile must not appear on the list. |
| **Outcome** | Result: Correct  After clicking on the “Delete” link of one social profile from the list this was the outcome: |
| **Notes** |  |

# Use case UC23.2 Manage his or her messages

Description

1. An authenticated user wishes to send a message; he or she login into the system; clicks on ‘Profile (Username)’; clicks on ‘Mailbox’; clicks on ‘Create a new message; fills the form and finally he or she hits the ‘Send’ button.
2. An authenticated user wishes to list his or her messages; he or she login into the system; clicks on ‘Profile (Username)’; clicks on ‘Mailbox’
3. An authenticated user wishes to list his or her messages grouped by tag; he or she login into the system; clicks on ‘Profile (Username)’; clicks on ‘Mailbox’ fills out the form to list by tag and clicks search.
4. An authenticated user wishes to delete a message; he or she login into the system; clicks on ‘Profile (Username)’; clicks on ‘Mailbox’; clicks on ‘Delete’ link of the message list and the message now, shows a tag ‘DELETED’, if he repeats the same again the message is now removed from the system.

Access

1. Login > Main Menu > Profile > Mailbox > Create a new message
2. Login > Main Menu > Profile > Mailbox
3. Login > Main Menu > Profile > Mailbox > Search
4. Login > Main Menu > Profile > Mailbox > Delete (Sent/Received message list)

Tests

|  |  |
| --- | --- |
| **Test 030** | |
| **Description** | The form must be filled in with the following data: Recipients: ‘’\*  Subject: ’’, Message Body: ’test’, Tags: ‘’ |
| **Expected** | The system must return a view with the message you just created. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save Message” button, this was the outcome: |
| **Notes** |  |

\*\*\* you can use as many usernames as you wish, there is a list with all the users of the system, you can pick whichever you desire, but you must always include your own to be able to send the message to yourself and see it in the list view.

|  |  |
| --- | --- |
| **Test 029** | |
| **Description** | An authenticated user messages must be listed |
| **Expected** | The system must return an authenticated user message list |
| **Outcome** | Result: Correct  After clicking on the “Mailbox” button on the main menu this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 030** | |
| **Description** | The form must be filled in with the following data: Tag: ‘’TAG” |
| **Expected** | The system must list all messages with the tag “TAG” |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Search” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 031** | |
| **Description** | An authenticated user message must be deleted |
| **Expected** | The system must return an authenticated user message list and message that has a tag ‘DELETED’, if done again the message must disappear. |
| **Outcome** | Result: Correct  After clicking on the “Delete Message” link of a message from the list this was the outcome:    When I clicked again on the “Delete Message” link again this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test 032** | |
| **Description** | The form must be filled in with the following data: Recipients: ‘company2’,  Subject:’’, Messages Body: ’test’ , Tags: ‘’ |
| **Expected** | The system must return to the edition view with a message error on the subject field, that it cannot be empty |
| **Outcome** | Result: Incorrect  The form was filled with the provided data and after pressing the “Save” button the system saved the entity. |
| **Notes** |  |

# Use case UC24.1 Broadcast a notification message

Description

An administrator wishes to send a broadcast notification; he or she login into the system; clicks on “System”; clicks on “Broadcast message”; he or she fills in a form with message and finally he or she hits the ‘Save’ button.

Access

Login > Main Menu > System > Broadcast message

Tests

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | The form must be filled in with the following data: Subject: ‘New terms and conditions’, Body: ‘We changed our conditions and terms’. |
| **Expected** | The system must save the entity property and return to the message list view that contains the new message entity in sent message list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save Message” button this was the outcome:  2019-05-07 (19) |
| **Notes** |  |

# Use case UC24.2 Launch the spammer or not-spammers process

Description

An administrator wishes to launch a process to flags the actors of the system as spammers or not-spammers; he or she login into the system; clicks on “List all actors”; clicks on “Compute spammers” and finally this process is launched.

Access

Login > Main Menu > System > List all actors > Evaluate spam

Tests

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | A hacker sends a message with the following data: Subject: ‘sex, Body: ‘I had sex last week’ holidays’.  An admin launches a process to evaluate spam. |
| **Expected** | The system displays all of the actors of the system, the hacker that sent that message must have an option active to be banned. |
| **Outcome** | Result: Correct  The form was filled with the provided data and then, logged as Admin I pressed the “compute spammers” button and this was the outcome:  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (22).png |
| **Notes** |  |

# Use case UC24.3 Ban an actor with the spammer flag

Description

An administrator wishes to ban an actor with a spammer flag; he or she login into the system; clicks on “System”; clicks on “List all actors” on “Compute spammers” and finally he or she clicks on “Ban” for an actor.

Access

Login > Main Menu > Administrator > List all actors > ban

Tests

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | An admin bans a spammer actor. |
| **Expected** | The system must return a spammer list which includes the previous spammer actor with an ‘Unban’ tag. |
| **Outcome** | Result: Correct  After the previous test, hacker1 was a suspicious actor and after pressing the ban button this was the outcome:  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (23).png  Also, hacker1 was unable to use the system functionalities. |
| **Notes** |  |

# Use case UC24.4 Unban an actor with the spammer flag

Description

An administrator wishes to unban an actor with a spammer flag; he or she login into the system; clicks on “Administrator”; clicks on “List all actors” and finally he or she clicks on “Unban” for an actor.

Access

Login > Main Menu > Administrator > Evaluate spam

Tests

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | An admin unbans a spammer actor. |
| **Expected** | The system must return a spammer list which includes the previous spammer actor with ‘Ban’ tag. |
| **Outcome** | Result: Correct  After the previous test, hacker1 was banned and clicking the “unban” link this was the outcome:  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (24).png  Also, hacker1 was able to login into the system and used its functionalities. |
| **Notes** |  |

# Additional tests

|  |  |
| --- | --- |
| **Test <#999>** | |
| **Description** | Change the language of the system, by clicking on the en | es just under the menu. |
| **Expected** | The language the system is displayed on changes. |
| **Outcome** | Result: Correct  After clicking on “es” language link the outcome was the following:  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (25).png  After clicking on “en” language link the outcome was the following:  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (27).png |
| **Notes** |  |

Login > see all actors > show(username)

|  |  |
| --- | --- |
| **Test <#999>** | |
| **Description** | The profile of an actor must be shown, if he/she is a spammer it will be shown |
| **Expected** | The system shows the profile with a Spammer attribute and true or false. |
| **Outcome** | Result: Correct  After pressing on hacker1 profile the outcome was the following:  :  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (28).png |
| **Notes** | //TODO |

Login > System > Configuration

|  |  |
| --- | --- |
| **Test <#999>** | |
| **Description** | The administrator configuration must be displayed |
| **Expected** | The system must return a full detailed configuration view. The default list of spam words includes “sex”, “viagra”, “cialis”, “one million”, “you’ve been selected”, “Nigeria”, and their corresponding Spanish translations. |
| **Outcome** | Result: Correct  After pressing on “Configuration” button the outcome was the following:  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (29).png  After changing the language, the system shows the words in Spanish as well and the outcome was the following:  C:\Users\Elerian\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2019-05-07 (30).png |
| **Notes** |  |

# Acme-Rookies v2.0

# Use case UC3: Self-assign a position to audit it & Manage his or her audits

Description

1 A user authenticated as an auditor wants to self-assign a position to audit it.

2.1 Create a new audit

2.2 List their audits

2.3 Display an audit

2.4 Update an audit

2.5 Delete an audit that’s not in final mode.

Access

1. Login ->Main menu -> Audit -> Create audit (choose a position to assign)
2. Login ->Main menu -> Audit -> List Audits
3. Login ->Main menu -> Audit -> List Audits -> Show (audit to be displayed).
4. Login ->Main menu -> Audit -> List Audits -> Edit (audit to be edited).
5. Login ->Main menu -> Audit -> List Audits -> Delete (audit to be deleted)

Tests

|  |  |
| --- | --- |
| **Test <#1> Create audit & Self-Assign position** | |
| **Description** | The form must be filled with the following data:  Text: “Test”; Score “5.0”; Position: “Position 2 Programacks”; |
| **Expected** | After clicking the save button the system will take you to the list of your audits where you can see the new audit you created with the date of the moment when it was saved. |
| **Outcome** | Result: Incorrect  The form was filled with the provided data, the audit moment was not the date of the moment when it was saved and this was the outcome after pressing the “Save” button: |
| **Notes** | We considered that se assignement of a position was done when an audit was done, hence the combination in the same use case. |

|  |  |
| --- | --- |
| **Test <#1.1> Create audit** | |
| **Description** | The form must be empty but for Score that must be: “Invalid” |
| **Expected** | They system will return the following errors:  Text: “ Must not be blank” ; Score: “Failed to convert property value of type java.lang.String to required type java.lang.Double for property score;..” Position: “Cannot be null”; |
| **Outcome** | Result: Correct  The form was filled with the provided data and this was the outcome after pressing the “Save” button: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2> List audits** | |
| **Description** | Click on List Audits |
| **Expected** | The system will return a list with all the audits that belong to the actor. |
| **Outcome** | Result: Correct  After pressing the “List audits” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#3> Show audit** | |
| **Description** | Click on show |
| **Expected** | They system will return a view where all of the audits data will be displayed. |
| **Outcome** | Result: Correct  After clicking on the “Show” link of one audit from the audit list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#4> Edit audit** | |
| **Description** | The form must be filled with the following data:  Text: “Test1” ; Score “6.0”; Final: “Unchecked” ;Position: “Position 2 Programacks”; |
| **Expected** | After clicking the save button the system will take you to the list of your audits where you can see the audit you modified. |
| **Outcome** | Result: Incorrect  The form was filled with the provided data, the audit moment was not the date of the moment when it was saved and this was the outcome after pressing the “Save” button: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#4.1> Edit audit** | |
| **Description** | The form must be filled with the following data:  Text: “Test2” ; Score “8.0”; Final: “checked” ;Position: “Position 2 Programacks”; |
| **Expected** | After clicking the save button the system will take you to the list of your audits where you can see the audit you modified and the commands to edit/delete it should’ve disappeared. |
| **Outcome** | Result: Incorrect  The form was filled with the provided data, the audit moment was not the date of the moment when it was saved and this was the outcome after pressing the “Save” button: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#4.2> Edit audit** | |
| **Description** | The form must be empty |
| **Expected** | They system will return the following errors:  Text: “ Must not be blank” ; Score: “Cannot be null” Position: “Cannot be null”; |
| **Outcome** | Result: Correct  The form was filled with the provided data and this was the outcome after pressing the “Save” button: |
| **Notes** |  |

# Use case UC4.1: Run a procedure to notify the existing users of the rebranding

Description

An administrator wants to notify the users about the rebranding to the currently registered users

Access

1. Login -> Main menu -> Profile(username) -> Mailbox -> Send the users the notification about the update

Tests

|  |  |
| --- | --- |
| **Test <#1> Notification** | |
| **Description** | Clicks on Send…Update |
| **Expected** | The following message will be sent to all users, and the option to send the message must disappear. |
| **Outcome** | Result: Correct  After clicking on the “Send the users the nofication about the update” link this was the outcome: |
| **Notes** |  |

# Use case UC4.2: Create user accounts for new auditors.

Description

An administrator wants to Create new user accounts for other administrators.

Access

1. Login -> Main menu -> Register -> Register as an auditor

Tests

|  |  |
| --- | --- |
| **Test <#1> Register an auditor** | |
| **Description** | The form must be filled with the following data:  holder: “TEST”, Credit Card Make: “VISA”, Credit Card Number:”1234123412341234”, CVV:”123”, Expiration month:”1”, Expiration Year: “2020”, Username: “auditortest”, Password & Repeat password: “auditortest”, Name: “test”, Surname: “test”, VAT%: “ASD12341234”, Photo: “http://www.photo.com/test.jpg”, Email:”email@email.com”, Address: “test”, phone: “612345678”; |
| **Expected** | The system will return you to the welcome page, so you can logout and login with the credentials auditortest/auditortest with the functions of an auditor |
| **Outcome** | Result: Correct  The auditor register form was filled with the provided data.    After pressing submit, I was redirected to the main page. Then, I clicked on login and filled the form with “auditortest/auditortest” credentials and it logged with success. Finally, this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#1.1> Register an auditor** | |
| **Description** | The form must be empty |
| **Expected** | The system will return the following errors holder: “Must not be blank”, Credit Card Make: “Must not be blank”, Credit Card Number:”Size must be from 16 to 16, does not match pattern”, CVV: “ Cannot be null”, Expiration month & Expiration Year: “this is an expired date”, Username: “Must not be empty”, Password & Repeat password: “Must not be empty”, Name: “Must not be empty”, Surname: “Must not be empty”, VAT%: “Does not match pattern”, Photo: “Must not be empty”, Email:” Must not be empty”, phone: “Does not match pattern”; |
| **Outcome** | Result: Correct  The system returns the auditor register form, it was filled with the provided data and this was the outcome after pressing the “Save” button : |
| **Notes** |  |

# Use case UC4.3: Launch a process to compute an audit score for every Company.

Description

An administrator wants to Launch a process to compute de audit scores for every company.

Access

1. Login -> Main menu -> Compute -> Compute Score

Tests

|  |  |
| --- | --- |
| **Test <#1> Compute Score** | |
| **Description** | Click on Compute Score |
| **Expected** | The system will display a message that reads: “**SCORE HAS BEEN COMPUTED, CHECK THE ACTOR LIST FOR CHANGES”**, And the Scores for the audits will have been computed, NIL will be displayed if the value is null. |
| **Outcome** | Result: Correct  After clicking on the “compute score” link this was the outcome:    Then, I checked the company scores and these were the outcomes: |
| **Notes** | If you just populated the database, the scores will be 0.61 for company1,  0.42 for company2 and NIL for company3. |

# Use case UC4.4 Display a dashboard (Level C)

Description

An administrator wishes to display his or her dashboard; he or she login into the system, clicks on ‘System’ and finally clicks on ‘Dashboard’.

Access

Login > Main Menu > System > Dashboard

Tests

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | The administrator dashboard must be displayed. |
| **Expected** | The system must return a list where all level c queries results are shown. |
| **Outcome** | Result: Correct  After pressing the “Dashboard” button on the menu bar this was the outcome: |
| **Notes** |  |

# 

# Use case UC9.1: Browse Providers & Navigate to their items.

Description

An unauthenticated user wants to see all the providers for the system and navigate to their items.

Access

1. Main menu -> List providers

2. Main menu -> List providers -> List items (of the provider you want to show).

Tests

|  |  |
| --- | --- |
| **Test <#1> List Providers** | |
| **Description** | Click on List Providers. |
| **Expected** | A table with all the providers of the system will be shown |
| **Outcome** | Result: Correct  After pressing the “List providers” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#2> Navigate to their items** | |
| **Description** | Click on List Items |
| **Expected** | A list with the items belonging to the provider you chose must be displayed. |
| **Outcome** | Result: Incorrect  After clicking on the “List items” link of one provider from the provider list this was the outcome. When I navigated to other provider items, the items shown were the same: |
| **Notes** |  |

# Use case UC9.2 Browse the list of items and navigate to their providers

Description

An unauthenticated user wishes to browse the items and navigate to their providers; he or she clicks on ‘List items’ on the main menu, then clicks on ‘username’ on the column Provider of the list to access to the provider profile.

Access

Main Menu > List items > username (Item list)

Test

|  |  |
| --- | --- |
| **Test <#001> List items** | |
| **Description** | The items must be listed, and there must be a way to see the provider’s profile. |
| **Expected** | The system must return an item list with its corresponding provider, at clicking on the usernames should redirect to their profiles. |
| **Outcome** | Result: Correct  After pressing the “List items” button on the menu bar this was the outcome:    Then, I clicked on the provider username link of one item from the item list and this was the outcome: |
| **Notes** |  |

# Use case UC9.3: Register to the system as a provider

Description

An unauthenticated user wishes to register to the system as a provider; he or she clicks on ‘Register’, clicks on ‘Register as provider’, fills the form and finally hits the ‘Save’ button.

Access

Main menu > Register > Register as provider

Tests

|  |  |
| --- | --- |
| **Test <#1>** | |
| **Description** | The form must be filled with the following data:  holder: “TEST”, Credit Card Make: “VISA”, Credit Card Number:”1234123412340000”, CVV:”123”, Expiration month:”5”, Expiration Year: “2020”, Username: “providertest”, Password & Repeat password: “providertest”, Name: “test”, Surname: “test”, VAT%: “ASD12341234”, Photo: “http://www.photo.com/test.jpg”, Email:”email@email.com”, Address: “test”, phone: “612345678”, Provider Make: “providertest”; |
| **Expected** | The system must save the provider properly and return the welcome page. You must be able to login into the system with that credentials. |
| **Outcome** | Result: Correct  The form was filled with the provided data.    After pressing submit, I was redirected to the main page. Then, I clicked on login and filled the form with “providertest/ providertest” credentials and it logged with success and this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#1>** | |
| **Description** | The form must be filled in with the following data: Holder: “TEST”, Make: ‘’, Credit card number: ‘’, , CVV: “555”, Exp. Month: “5”, Exp. Year: “2021”, Username: “providertester”, Password & , Repeat the password: : “providertester”, Name: “test”, Surnames: “test”, Vat number: ‘’, Photo: “https://www.instagram.com”, Email: “email@email.com”, , Address:’’, Phone: “612345678”, Provider make: ’’. |
| **Expected** | The system must not save the provider and return the provider register showing the following errors:  - On fields credit card make and provider make, ‘Must not be blank’.  - On field vat, ‘Does not match pattern’.  - On field credit card, ‘Does not match pattern’ and ‘Size must be from 16 to 16’. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Submit” button this was the outcome: |
| **Notes** |  |

# Use case UC10.1 Manage his or her catalogue of items

Description

1. A provider wishes to list their items; he or she login into the system, and finally clicks on ‘List my items’ on the main menu.
2. A provider wishes to show their items; he or she login into the system, clicks on ‘List my items’ and finally clicks on the ‘Show’ link of an item from the item list.
3. A provider wishes to create an item; he or she login into the system, clicks on ‘List my items’, clicks on the ‘Create item’, fills the form and finally hits the ‘Save’ button.
4. A provider wishes to update an item; he or she login into the system, clicks on ‘List my items’, clicks on the ‘Edit’ link of an item from the item list, fills de form and finally hits the ‘Save’ button.
5. A provider wishes to delete an item; he or she login into the system, clicks on ‘List my items’, and finally clicks on the ‘Delete’ link of an item from the item list.

Access

1. Login > Main Menu List my items
2. Login > Main Menu > List my items > Show (Item list)
3. Login > Main Menu > List my items > Create an item
4. Login > Main Menu > List my items > Edit (Item list)
5. Login > Main Menu > List my items > Delete (Item list)

Test

|  |  |
| --- | --- |
| **Test <#001> List items** | |
| **Description** | The provider items must be listed. |
| **Expected** | The system must return the provider item list. |
| **Outcome** | Result: Correct  After pressing the “List my items” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#002> Show item** | |
| **Description** | A provider item must be displayed. |
| **Expected** | The system must display all provider item details. |
| **Outcome** | Result: Correct  After clicking on “Show” link of one item from the item list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#003> Create item** | |
| **Description** | The form must be filled in with the following data: Name: ‘Test’, Description: ‘Test’, Link: ‘https://www.test.com’, Pictures: ‘’. |
| **Expected** | The system must save the item properly, return the item list and the new item must appear on the list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#004> Edit item** | |
| **Description** | The form must be filled in with the following data: Name: ‘’, Description: ‘’, Link: ‘’, Pictures: ‘’. |
| **Expected** | The system must not save the item properly, return the item edit view and ‘Must not be blank’ error messages must appear in Name, Description and Link. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#003> Edit item** | |
| **Description** | The form must be filled in with the following data: Name: ‘Test update’, Description: ‘Test update’, Link: ‘https://www.testupdate.com’, Pictures: ‘’. |
| **Expected** | The system must save the item properly, return the item list and the updated item must appear on the list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#005> Delete item** | |
| **Description** | A provider item must be deleted. |
| **Expected** | The system must return the item list view and the item must not appear on the list. |
| **Outcome** | Result: Correct  After clicking on “Delete” link of one item from the item list this was the outcome: |
| **Notes** |  |

# Use case UC11.1 Display a dashboard (Level B)

Description

An administrator wishes to display his or her dashboard; he or she login into the system, clicks on ‘System’ and finally clicks on ‘Dashboard’.

Access

Login > Main Menu > System > Dashboard

Tests

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | The administrator dashboard must be displayed. |
| **Expected** | The system must return a list where all level B queries results are shown. |
| **Outcome** | Result: Correct  After pressing the “Dashboard” button on the menu bar this was the outcome: |
| **Notes** |  |

# Use case UC13.1 Manage his or her sponsorships

Description

1. A provider wishes to list their sponsorships; he or she login into the system and finally clicks on ‘My sponsorships’ on the main menu.
2. A provider wishes to show their sponsorships; he or she login into the system, clicks on ‘My sponsorships’ and finally clicks on the ‘Show’ link of a sponsorship from the sponsorship list.
3. A provider wishes to create a sponsorship; he or she login into the system, clicks on ‘My sponsorships’, clicks on the ‘Create new sponsorship’, fills the form and finally hits the ‘Save’ button.
4. A provider wishes to update a sponsorship; he or she login into the system, clicks on ‘My sponsorships’, clicks on the ‘Edit’ link of a sponsorship from the sponsorship list, fills de form and finally hits the ‘Save’ button.
5. A provider wishes to delete a sponsorship; he or she login into the system, clicks on ‘My sponsorships’ and finally clicks on the ‘Delete’ link of a sponsorship from the sponsorship list.

Access

1. Login > Main Menu > My sponsorships
2. Login > Main Menu > My sponsorships > Show (Sponsorship list)
3. Login > Main Menu > My sponsorships > Create new sponsorship
4. Login > Main Menu > My sponsorships > Edit (Sponsorship list)
5. Login > Main Menu > My sponsorships > Delete (Sponsorship list)

Test

|  |  |
| --- | --- |
| **Test <#001> List sponsorship** | |
| **Description** | The provider sponsorships must be listed. |
| **Expected** | The system must return the provider sponsorship list. |
| **Outcome** | Result: Correct  After pressing the “My sponsorships” button on the menu bar this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#002> Show sponsorship** | |
| **Description** | A provider sponsorship must be displayed. |
| **Expected** | The system must display all provider sponsorship details. |
| **Outcome** | Result: Correct  After clicking on “Show” link of one sponsorship from the sponsorship list this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#003> Create sponsorship** | |
| **Description** | The form must be filled in with the following data: Banner: ‘https://www.test.com’, Target: ‘https://www.test.com’, Position: ‘Position1 programacks’. |
| **Expected** | The system must save the sponsorship properly, return the sponsorship list and the new sponsorship must appear on the list. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#004> Edit sponsorship** | |
| **Description** | The form must be filled in with the following data: Banner: ‘https://www.test.com’, Target: ‘https://www.test.com’, Position: ‘’. |
| **Expected** | The system must not save the sponsorship properly, return the sponsorship edit view with the error message ‘This operation cannot be committed’ since you need to choose a position. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#003> Edit sponsorship** | |
| **Description** | The form must be filled in with the following data: Banner: ‘test.com’, Target: ‘https://www.test.com’, Position: ‘Position1 programacks’. |
| **Expected** | The system must not save the sponsorship properly, return the sponsorship edit view and ‘Invalid URL’ message error on banner field. |
| **Outcome** | Result: Correct  The form was filled with the provided data and after pressing the “Save” button this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#005> Delete sponsorship** | |
| **Description** | A provider sponsorship must be deleted. |
| **Expected** | The system must return the sponsorship list view and the sponsorship must not appear on the list. |
| **Outcome** | Result: Correct  After clicking on “Delete” link of one sponsorship from the sponsorship list this was the outcome: |
| **Notes** |  |

# Use case UC14.1 Display a dashboard (Level A)

Description

An administrator wishes to display his or her dashboard; he or she login into the system, clicks on ‘System’ and finally clicks on ‘Dashboard’.

Access

Login > Main Menu > System > Dashboard

Tests

|  |  |
| --- | --- |
| **Test 022** | |
| **Description** | The administrator dashboard must be displayed. |
| **Expected** | The system must return a list where all level a queries results are shown. |
| **Outcome** | Result: Correct  After pressing the “Dashboard” button on the menu bar this was the outcome: |
| **Notes** |  |

# Additional tests

|  |  |
| --- | --- |
| **Test <#999>** | |
| **Description** | A position must be shown, and the corresponding actor must be able to browse its audits, if any. |
| **Expected** | The system must list all position audits. |
| **Outcome** | Result: Correct  After pressing the “List of positions” button on the menu bar, I clicked on “Show” link of one position from the position list and this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#999>** | |
| **Description** | A company profile must be displayed, and its score must be shown. |
| **Expected** | The system must show the company score. |
| **Outcome** | Result: Correct  After pressing the “List of companies” button on the menu bar, I clicked on “Show” link of one company from the company list and this was the outcome: |
| **Notes** |  |

|  |  |
| --- | --- |
| **Test <#999>** | |
| **Description** | A position must be shown and one of its sponsorships too. |
| **Expected** | The system must show the sponsorship banner and the flat rate must be charged. |
| **Outcome** | Result: Correct  After pressing the “List of positions” button on the menu bar, I clicked on “Show” link of one position from the position list and this was the outcome: |
| **Notes** | A sponsorship is displayed if the position have at least one sponsorship. |