Acceptance tests   
Acme Hacker-Rank v1.3

In this document we provide all acceptance tests required in D04 – Acceptance testing lesson. For the Acme-Hacker rank project

|  |  |
| --- | --- |
| ****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;  Carrasco, Antonio; |
| **Notes** |  |

Table of contents

[Use case UC7.1: Register to the system as a Company or a Hacker 3](#_Toc7698073)

[Use case UC7.2: List positions and navigate to the companies 5](#_Toc7698074)

[Use case UC7.3: List positions and navigate to the companies 5](#_Toc7698075)

[Use case UC7.4: Search for a position 6](#_Toc7698076)

[Use case UC8.2: Edit personal data 6](#_Toc7698077)

[Use case UC9.1: Manage the positions 8](#_Toc7698078)

[Use case UC9.2: Manage their problems 11](#_Toc7698079)

[Use case UC9.3: Manage the applications to their positions 13](#_Toc7698080)

[Use case UC10.1: Manage his or her applications 14](#_Toc7698081)

[Use case UC11.1: Create new administrators 16](#_Toc7698082)

[Use case UC11.2: Display a dashboard (Level C) 17](#_Toc7698083)

[Use case UC14: Customize system at run time 17](#_Toc7698084)

[Use case UC17.1 Manage his or her curricula 18](#_Toc7698085)

[Use case UC17.2 Manage his or her finder 24](#_Toc7698086)

[Use case UC18.1: Display a dashboard (Level B) 25](#_Toc7698087)

[Use case UC23.1 Manage his or her social profiles 25](#_Toc7698088)

[Use case UC23.2 Manage his or her messages 27](#_Toc7698089)

[Use case UC24.1 Broadcast a notification message 28](#_Toc7698090)

[Use case UC24.2 Launch the spammer or not-spammers process 29](#_Toc7698091)

[Use case UC24.3 Ban an actor with the spammer flag 29](#_Toc7698092)

[Use case UC24.4 Unban an actor with the spammer flag 30](#_Toc7698093)

[Additional tests 30](#_Toc7698094)

# 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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

\*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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **Test <#5> Delete a position** | |
| **Description** | Click on delete on the position you want to delete |
| **Expected** | It returns the view with all the postions, but the one you just chose to delete. |
| **Outcome** | //TODO |
| **Notes** | //TODO |

# 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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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 |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** |  |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **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** |  |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

# 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: ‘\*username\*, hacker1, hacker2, hacker3, hacker4, hacker5, hacker6, company1, company2, company3, admin’\*\*\*  Subject: ’test’, Mesagges Body: ’test’ , Tags: ‘TAG’ |
| **Expected** | The system must return a view with the message you just created ( you must send it to yourself) |
| **Outcome** | //TODO |
| **Notes** | //TODO |

\*\*\* 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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

|  |  |
| --- | --- |
| **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** | //TODO |
| **Notes** | //TODO |

# 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 the a message list view that contains the new message entity in sent message list. |
| **Outcome** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

# Use case UC24.4 Unban 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 “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** | //TODO |
| **Notes** | //TODO |

# 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** | //TODO |
| **Notes** | //TODO |

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 atribute and true or false. |
| **Outcome** | //TODO |
| **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** | //TODO |
| **Notes** | //TODO |