Use Cases Tests

Programming Project subject
Group 13.2 – Delivery 3.0

Eduardo Hernández
`edu.hernandez`

Álex Manzaneda
`alex.manzaneda`

Oriol Valencia
`oriol.valencia`

Sara Vidal
`sara.vidal`

December 2025

Contents

1 Profile Use Cases

1.1 Account Creation

- **Test Object**: Check that users can create an account with a unique username and a valid password. Verify that the creation persists in the system.
- **Required Data Files**: N/A
- **Studied Values**:
  - Valid username: `newUser`. This username is not already taken in the system and it's not empty.
  - Valid password: `pass`. The password it's not empty and meets the security requirements.
  - Repetitive username: `newUser`. This username is already taken in the system.
  - Empty username: `""`. The username field is left empty.
  - Empty password: `""`. The password field is left empty.
- **Studied Effects**: When we create the first user with *[Valid username]* and *[Valid password]* It must appear a confirmation message indicating that the account has been created successfully. Then, if the user confirms the message, they should be redirected to the login page. Also we have to check that the new user is added to the database, with the correct username and password.

  If we try to create an user with *[Repetitive username]*, an error message should appear indicating that the username is already taken.

  If we try to create an user with *[Empty username]* or *[Empty password]*, an error message should appear indicating that the fields cannot be empty.

1.2 Log in

- **Test Object**: Verify that users can log in with valid credentials and are prevented from logging in with invalid credentials.
- **Required Data Files**: Existing user, for example: `Existing_User` with password `pass`.
- **Studied Values**:
  - Invalid username: `wrongUser`. This username does not exist in the system.
  - Invalid password: `wrongPass`. The password does not match the username
  - Empty username: `""`. The username field is left empty.
  - Empty password: `""`. The password field is left empty.
- **Studied Effects**: When we log in with an *[Existing user]*, it must appear a welcome message indicating that the login was successful. Then, if the user confirms the message, they should be redirected to the main page.

  If we try to log in with *[Invalid username]* or *[Invalid password]*, an error message should appear indicating that the credentials are incorrect .

  Whem trying to log in with *[Empty username]* or *[Empty password]*, an error message should appear indicating that the fields cannot be empty.

1.3 Modify Username

- **Test Object**: Verify that a logged-in user can modify their username to a new unique username, also check if this change persists.
- **Required Data Files**: Existing user, for example: `Existing_User` with password `pass`. Another user with username `proba` to test the repetitive username case.
- **Studied Values**:
  - Valid new username: `newUser`. This username does not exist in the system. We will change the old username to this new one.
  - Invalid username: `proba`. This username is already taken in the system.
  - Empty username: `""`. The username field is left empty.
- **Studied Effects**: We log in with the existing user.
  If we try to change the username to *[Valid new username]*, after confirming the change, a message should appear indicating that the username has been updated successfully. We should also verify that the new username is reflected in the database.

  When trying to change the username to *[Invalid username]*, an error message should appear indicating that the username is already taken.

  When attempting to change the username to *[Empty username]*, nothing happens and the username remains unchanged.

1.4 Modify Password

- **Test Object**: Verify that a logged-in user can modify their password, ensuring the change is effective and persists.
- **Required Data Files**: Existing user, for example: `Existing_User` with password `pass`.
- **Studied Values**:
  - Invalid old password: `wrongPass`. The password does not match the username
  - Valid new password: `newPass`. The new password it's not empty and meets the security requirements. We will change the old password to this new one.
  - Empty new password: `""`. The new password field is left empty.
- **Studied Effects**: We log in with *[Valid username]* and *[Valid old password]*.
  If we try to change the password to *[Valid new password]*, writting it twice to confirm and using *[Valid old password]* to authorize the change, a message should appear indicating that the password has been updated successfully. We should also verify that the new password is reflected in the database.

  Trying to change the password to *[Valid new password]* but writing different values in the confirmation field should result in an error message indicating that the new passwords do not match.
  If we try to log in again with the old password, it should fail. Logging in with the new password should succeed.

  When trying to change the password using *[Invalid old password]*, an error message should appear indicating that the old password is incorrect.

  When attempting to change the password to *[Empty new password]*, an error message should appear indicating that all fields must be filled.

1.5 Modify Description

- **Test Object**: Verify that a logged-in user can modify their profile description and that the change is saved correctly.
- **Required Data Files**: Existing user, for example: `Existing_User` with password `pass`.
- **Studied Values**:
  - Valid username: `Existing_User`. This username existes in the system and it's not empty.
  - Valid password: `pass`. The password it's not empty and matches the username.
  - New description: `"This is my new profile description."`. A valid string to update the profile description. We will change the old description to this new one.
  - Empty description: `""`. The description field is left empty.
- **Studied Effects**: We log in with *[Valid username]* and *[Valid password]* suposing that any of this values have changed.
  If we change the profile description to *[New description]* and save the changes, a message should appear indicating that the profile has been updated successfully. We should also verify that the new description is reflected in the database and displayed correctly on the user's profile page.

  If we change the profile description to *[Empty description]*nothing happens and the description remains unchanged.

1.6 Delete profile

- **Test Object**: Verify that a logged-in user can delete their profile and that all associated data is removed from the system.
- **Required Data Files**: Existing user, for example: `Existing_User` with password `pass`.
- **Studied Values**: The user accept o declining the deletion in the confirmation dialog.
- **Studied Effects**: We log in with *[Valid username]* and *[Valid password].*
  When we choose to delete the profile, a confirmation dialog should appear asking the user to confirm the deletion. If the user confirms, a message should appear indicating that the profile has been deleted successfully. We should also verify that the user's data is removed from the database and that they are redirected to the account creation or login page.

  If we try to log in again with the deleted profile credentials, it should fail, indicating that the profile no longer exists.
  If the user cancels the deletion in the confirmation dialog, no changes should be made and the user should remain logged in with their profile intact.

2 Form Use Cases

2.1 Create Form and Add Questions

- **Test Object**: Create Form, Add Questions (Modify Form use Case), View Forms
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *Form Management* option and then the *Create Form*.
  - We set the title `''test1''` and description `''desc1''` and pulse the Next button. We are expecting to create a new form, these values are expected to work and do not raise any exception.
  - We will try to add 3 questions, one of each type, filling the required fields with this information and pressing the Save button after the creation of each question. Once we have saved the third question we can pulse the Finish button to exit. The objective of this input dataset is to add questions correctly, without testing exceptions or unexpected values.
  - First question:
    - Question Text: `''q1''`
    - Question Type: `NUMERIC`
    - Minimum Bound: `''1''`
    - Maximum Bound: `''5''`
  - Second question:
    - Question Text: `''q2''`
    - Question Type: `OPEN_ENDED`
  - Third question:
    - Question Text: `''q3''`
    - Question Type: `MULTIPLE_CHOICE`
    - Choices: `''a,b,c''`
    - Not ordered
    - Max choices: `''1''`
  - All forms and questions of each form can be seen when pressing *View Existing Forms*.
- **Studied Effects**: Expecting the form to be created. Once we press the Next button, a panel is shown with *"Form Created Successfully!"* message. We can then add questions to the form created.
  We expect the questions to be linked to the form. When we press the Save button after filling each of the questions, a panel is shown with *"Question Added Successfully!"* message. Once we press the Finish button, we can press *View Existing Forms*. We will see here all the forms in the application. When pulsing on the `test1` form we will see the 3 questions.

2.2 Create Form - Wrong

- **Test Object**: Wrong Create Form
- **Required Data Files**: N/A
- **Studied Values**:

  - We loggin with the user ``prueba'' and password ``prueba'', existing user.
  - Select the *Form Management* option and then the *Create Form*.
  - We set the title ``General Test Form'' and description ``This is a general test form for multiple tests'' and pulse the Next button.

- **Studied Effects**: When we press the Next button we see a message indicating that *"The title is invalid"*. No form is created as there was an existing form with that value.

2.3 Export Form

- **Test Object**: Export Form
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user ``prueba'' and password ``prueba'', existing user.
  - Select the *Form Management* option and then the *View Existing Forms*.
  - Select the ``form' form and pulse the button *Export Form*. We then select the own directory where we want to keep it.
- **Studied Effects**: When the directory is selected and the form is exported the message *"Form Exported Successfully to: path_selected"* is shown. A document .json is in my computer containing the form ``General Test Form''.

2.4 Add Questions - Wrong

- **Test Object**: Wrong Add Questions (Modify Form use Case)
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user ``prueba'' and password ``prueba'', existing user.
  - Select the *Form Management* option and then the *Manage My Forms*.
  - Select the ``General Test Form'' form and *Add Question* option.
  - Letting the question text empty.
  - Numeric question with minimum bound greater than maximum bound:
    - Minimum Bound: ``2''
    - Maximum Bound: ``1''
  - Numeric question with non numeric bound:
    - Minimum Bound: ``a''
    - Maximum Bound: ``1''
  - Numeric question with empty bounds:
    - Minimum Bound: ``''
    - Maximum Bound: ``''
  - Multiple choice question with empty choices.
  - Multiple choice question with max choices greater than number of choices.
  - Multiple choice unordered question with empty max choices.
  - Ordered multiple choice question with max choices.

- **Studied Effects**:
  - When Question Text is empty, the message *"Question Text Cannot be Empty"* is shown.
  - When minimum bound is greater than maximum bound, the message *"Invalid numeric bounds"* is shown.
  - When numeric bounds are invalid or empty, the message *"Invalid numeric value"* is shown.
  - When choices are empty, the message *"Choices cannot be empty"* is shown.
  - When maxChoices is greater than number of choices, the message *"The maxChoices field of the question is incorrect"* is shown.
  - When unordered multiple choice maxChoices is empty, the message *"Please specify the maximum number of choices for unordered questions."* is shown.
  - Ordered multiple choice questions with maxChoices are added successfully, but this value is ignored.
  - Finally, the form `''General Test Form''` only contains 5 questions and the last one must be deleted for future tests.

2.5 Change Title and Description

- **Test Object**: Change Title (Modify Form), Change Description (Modify Form)
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *Form Management* option and then the *Manage My Forms.*
  - Select the `''form to change title''` form and pulse the button *"Change Title"*, we write `''b''` and press *"OK"*. We are expected to change the title as no other form has this title value.
  - Pulse the button *"Change Description"*, we can complete the description shown by adding `''bsolute description''`, without erasing the actual description. When we press *"OK"* the new description will be kept.
- **Studied Effects**: When we set the new title and pulse *"OK"* the message *"Title updated Successfully!"* is shown and the title is modified.
  When we set the new description and pulse *"OK"* the message *"Description updated Successfully!"* is shown and the description is modified.
  We can see at the top of the window that the form title is `''b''` and the description is `''absolute description''`, so the changes have been done.

2.6 Change Title - Wrong

- **Test Object**: Wrong Change Title (Modify Form)
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *Form Management* option and then the *Manage My Forms.*
  - Select the `''Sports Survey''` form and pulse the button *"Change Title"*, we write `''General Test Form''` and press *"OK"*. We are expected to not change the title as there is another form that has this title value.

- **Studied Effects**: When we set the new title and pulse *"OK"* the message *"Error updating title: Title already exists"* is shown and no title is modified.
  We can see at the top of the window that the form title is `''Sports Survey''` , so the changes have never been done.

2.7 Delete Question

- **Test Object**: Delete Question (Modify Form), Wrong Delete Question (Modify Form)
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *Form Management* option and then the *Manage My Forms.*
  - Select the `''Form to delete Question''` form and pulse the button *"Delete Question"*.
  - Select the Q1 and pulse *"Delete Selected Question"*. On the validation panel we press *"Yes"*.
  - Once we have deleted the question, we press the *"Delete Question"* button once again.
- **Studied Effects**: When we validate that we want to delete the question, the message *"Question Deleted Successfully!"* is shown.
  When we try to delete a question on a form that doesn't have any question, no question can be selected. When pressing the button *"Delete Selected Question"*, the message *"Please Select a Question to Delete"* is shown and nothing is deleted.

2.8 Modify Question

- **Test Object**: Modify Question (Modify Form), Wrong Modify Question (Modify Form)
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *Form Management* option and then the *Manage My Forms.*
  - Select the `''form to modify question''` form and pulse the button *"Modify Question"*.
  - Select the Q1 and on the *"New Question Text"* box change the actual text by `''b''`. Press the button *"Modify Question"*.
  - Once we have modified the question, we press the *"Modify Question"* button once again and pulse the *"Modify Selected Question"* button without selecting any question.
- **Studied Effects**: When we validate the change the message *"Question text modified successfully!"* is shown and the question text is changed.
  When we try to modify a question on a form without selecting any question, the message *"Please Select a Question to Modify"* is shown and nothing is modified.

2.9 Delete Form

- **Test Object**: Delete Form
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user ``prueba'' and password ``prueba'', existing user.
  - Select the *Form Management* option and then the *Manage My Forms*.
  - Select the ``form to be deleted'' form and pulse the button *"Delete Form"*. Then press *"Yes"* on the validation panel.
- **Studied Effects**: When we validate the deletion of the form, the message *"Form deleted successfully!"* is shown and the form is no longer on the application. By pressing the *"Manage My Forms"* button once again we will see that now the ``form to be deleted'' form is no longer among my forms.

2.10 Delete Answers of Form

- **Test Object**: Delete answers of form
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user ``prueba'' and password ``prueba'', existing user.
  - Select the *Form Management* option and then the *View My Answered Forms*.
  - Select the ``Sports Survey'' form and pulse the button *"Delete Answers"*. Then press *"Yes"* on the validation panel.
- **Studied Effects**: When we validate the deletion of the answers, the message *"Your answers have been deleted successfully!"* is shown and the answers are no longer on the application. By pressing once again the *"View My Answered Forms"* button we will see that now no answers appear. This way we validate that a user can only delete his own answers, and only if the answers existe.

2.11 Import Form

- **Test Object**: Import Form, Import Form (Wrong)
- **Required Data Files**: `form3question.json`
- **Studied Values**:
  - We loggin with the user ``prueba'' and password ``prueba'', existing user.
  - Select the *Form Management* option and then the *Create Form*.
  - Push the button *"Import from JSON"*, and then the *"Browse Files"*. Finally, select the form ``form3question.json'' from the file mockdata and press *"open"*.
  - Push the button *"Import Form"* to have the new form.
  - We retry the whole process once again, with the form already imported. We expect to not be able to import the form again (same title).
- **Studied Effects**: When we press the final *"Import Form"* button, the form is imported and the message *"Form Imported Successfully!"* is shown.
  When we press the final *"Import Form"* button the second time, the form is never imported and the message *"Error importing the form: The title is invalid."* is shown.
  If we go now to *"Manage My Forms"* we will see that the new form is added.

2.12 Import Form - Wrong

- **Test Object**: Import Form (Wrong)
- **Required Data Files**: `formNoTitle.json`, `formNoField.json`
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *Form Management* option and then the *Create Form.*
  - Push the button *"Import from JSON"*, and then the *"Browse Files"*. Finally, select the form `''formNoTitle.json''` from the file mockdata and press *"open"*. As the title field is empty we expect the form to do not be imported.
  - Push the button *"Import Form"* to try the import.
  - We push again the *"Browse Files"* button and select the form `''formNoField.json''` from the file mockdata and press *"open"*. As no `''title''` field is defined on the form expecting to be imported, there are fields missing and no form will be imported.
- **Studied Effects**: When we press the final *"Import Form"* button, the form is not imported and the message *"Error importing Form: Title cannot be empty."* is shown.
  When we press the final *"Import Form"* button the second time, the form is never imported and the message *"Error importing form: Invalid form structure: Form must have title, description, and at least one question."* is shown.
  If we go now to *"Manage My Forms"* we will see that none of these forms have been added to the profile forms.

3 Answer Form Use Cases

3.1 Import Answer to Form and Answer Form

- **Test Object**: Import Answer to form, Answer Form
- **Required Data Files**: `answerSingleForm.csv`
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *"Answer Form"* option and then the form `''form1''`.
  - Push the button with the import emoji and select the file `''answerSingleForm.csv''`. Once selected press *"Open"*. The file is correct and the answers are expected to be imported.
  - Once the answers are imported, press the *"Submit Answers"* button to answer the form with the answers imported.
- **Studied Effects**: When importing, when we press the final *"Open"* button, the answers are imported and the message *"Answers Imported Successfully from CSV!"* is shown. The answers imported appear on the answers fields.
  When we press the final *"Submit Answers"* button, as there was no answer yet from the user to the form, the message *"Answers submitted successfully"* is shown.
  If we go now to *"Form Management"* and press *"View My Answered Forms"* we will see that the new answer is added.

3.2 Import Answer to Form - Wrong *(Missing Answers)*

- **Test Object**: Wrong Import Answer to form
- **Required Data Files**: `answerLessFields.csv`
- **Studied Values**:
  - We loggin with the user `‘‘prueba’’` and password `‘‘prueba’’`, existing user.
  - Select the *“Answer Form”* option and then the form `‘‘General Test Form’’`.
  - Push the button with the import emoji and select the file `‘‘answerLessFields.csv’’`. Once selected press *“Open”*. The file is incorrect and the answers are expected to not be imported, there are only 3 answers.
- **Studied Effects**: When importing, when we press the final *“Open”* button, the answers are never imported and the message *“The imported file contains 3 answers, but this form has 4 questions. Please select a file with matching answers.”* is shown. The answers imported do not appear on the answers fields.
  If we go now to *“Form Management”* and press *“View My Answered Forms”* we will not see that the new answer imported.

3.3 Import Answer to Form - Wrong *(Invalid ID)*

- **Test Object**: Wrong Import Answer to form
- **Required Data Files**: `answerNoID.csv`
- **Studied Values**:
  - We loggin with the user `‘‘prueba’’` and password `‘‘prueba’’`, existing user.
  - Select the *“Answer Form”* option and then the form `‘‘General Test Form’’`.
  - Push the button with the import emoji and select the file `‘‘answerNoID.csv’’`. Once selected press *“Open”*. The file is incorrect and the answers are expected to not be imported, there is no id.
- **Studied Effects**: When importing, when we press the final *“Open”* button, the answers are never imported and the message *“Error importing answers: Invalid UFID in CSV: aaaaaaa”* is shown. The answers imported do not appear on the answers fields.
  If we go now to *“Form Management”* and press *“View My Answered Forms”* we will not see that the new answer imported.

3.4 Answer Form - Wrong

- **Test Object**: Wrong Answer Form
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user `‘‘prueba’’` and password `‘‘prueba’’`, existing user.
  - Select the *“Answer Form”* option and then the form `‘‘General Test Form’’`.
  - We will fill the fields one by one with the following unexpected values, validating that no answer is kept, pushing the *“Submit Answers”* button after each answer try:
  - On the Q1, we will answer a text longer than 1000 characters. An open ended question may never have more than 1000 chars and an exception is expected to show, without submitting answers.

- Now, we erase the answer box of Q1 and answer Q2 with `''w''`. As `''w''` is not an option an exception is expected to be shown, without submitting answers.
- Still in Q2, we will now answer with more than the max_options, answering `''a, b, c''`, expecting an exception to be shown without submitting answers.
- Now, erasing the previous attempts of answer, we will try to answer Q4. We firstly answer with a non numeric answer: `''magic''`. Expecting an exception to be shown without submitting answers.
- Still in Q4 we will answer with a value outside the bounds: `''1200''`. Expecting an exception to be shown without submitting answers.
- Finally we press the *"Back"* button and go to *"Form Management"* and *"View My Answered Forms"*. We will see that no answer exists to the form `''General Test Form''`.

• **Studied Effects**:

- When answering with more than 1000 chars, the exception *"Error: Question 'Open Ended Question: open_ended question Enter your response (text): ' answer exceeds maximum length of 1000 characters (current: 1002)"* is shown and no answers are submitted.
- When answering a Multiple_Choice question with an unexisting option, the exception *"Error: Question 'Multiple Choice Question: multiple_choice question a, b, c, d, e MaxChoices: 1 Enter your response separated by', ": choice 'w'"* is shown and no answer is submitted.
- When answering a Multiple_Choice question with more than the maximum number of options, the exception *"Error: Too many choices selected. Maximum allowed: 1, but 3 were selected."* is shown and no answer is submitted.
- When answering a Numeric question with a non numeric answer, the exception *"Error: Question 'Numeric Question: question_numeric [Rango: 1 - 100] Enter your response (number): ' expects a numeric answer, but received: 'magic' "* is shown and no answer is submitted.
- When answering a Numeric question with a numeric answer outside of the bounds, the exception *"Error: Question 'Numeric Question: question_numeric [Rango: 1 - 100] Enter your response (number): ' expects a numeric answer, but received: 1200"* is shown and no answer is submitted.

3.5 Answer Form with Unanswered Questions

• **Test Object**: Answer Form

• **Required Data Files**: N/A

• **Studied Values**:

- We loggin with the user `''prueba''` and password `''prueba''`, existing user.
- Select the *"Answer Form"* option and then the form `''form to answer empty''`.
- We will not fill any of the questions (one of each type), but when pressing the *"Submit Answers"* button, questions are expected to be kept (unanswered questions).

• **Studied Effects**: We are studying the unanswered question of a form, and it worked right as the message *"You have unanswered questions. Submit anyway?"* was shown when submitting answers and also the message *"Answers submitted successfully!"*. The answers were submitted and on the *"Form Management" "View My Answered Forms"*, the answer appears.

3.6 Modify Answers of Form

- **Test Object**: Modify answers of form
- **Required Data Files**: N/A
- **Studied Values**:
  - We loggin with the user `''prueba''` and password `''prueba''`, existing user.
  - Select the *"Answer Form"* option and then the form `''Basic Satisfaction Form''`. Validate that we want to modify answers.
  - Once we have introduced a correct answer (for example `''1''`) and pressed *"Update Answers"*, the form answer will be modified.
- **Studied Effects**: We are studying that answers can be modified and that the new answer is kept. We can only modify answers of a form already answered by the user, or else we will be answering the form. The validation *"You have already answered this form. Do you want to edit your answers?"* is shown when selecting a form already answered.
  If we then go to *"Form Management"*, *"View My Answered Forms"* and select `''Basic Satisfaction Form''`, we will see the answer modified.

4 Affinity Group Use Cases

4.1 View an Affinity Group for a Form

- **Test Object**: View Affinity Group
- **Required Data Files**:
  - Existing form with multiple questions
  - Multiple users with answers submitted for that form
  - K-Means algorithm executed on the form (affinity groups generated)
- **Studied Values**:
  - User with assigned affinity group
  - User without assigned affinity group
  - K-Means parameters used (K value)
- **Studied Effects**:
  - Correct display of affinity group information and member count
  - Correct grouping of users based on K-Means results
  - Data consistency between UI and database
  - Page load performance with large member lists
  - Correct rendering of groups with various sizes (small, medium, large)

4.2 View an Affinity Group for a Form without the Algorithm Executed

- **Test Object**: View Affinity Group without K-Means executed
- **Required Data Files**:
  - Existing form with multiple questions
  - Multiple users with answers submitted for that form
  - K-Means algorithm not executed on the form
- **Studied Values**:
  - User attempting to view affinity group
  - System error or algorithm failure scenarios
- **Studied Effects**:
  - Appropriate message indicating that affinity groups are not available
  - No display of affinity group information or members
  - Error message provides guidance on how to proceed

4.3 Visit Affinity Group Member Profile

- **Test Object**: View Affinity Group Member Profile
- **Required Data Files**:
  - Existing form with multiple questions
  - Multiple users with answers submitted for that form
  - K-Means algorithm executed on the form (affinity groups generated)
- **Studied Values**:
  - Selecting different members from the affinity group
  - Users with incomplete or missing profile information
- **Studied Effects**:
  - Correct display of selected member's profile information
  - Accurate reflection of member's username and description
  - Proper handling of navigation back to affinity group view
  - Data integrity: profile data matches database state
  - Graceful handling of deleted or inactive users
  - Page responsiveness when switching between different members

4.4 Export Results

- **Test Object**: Export Affinity Group Results
- **Required Data Files**:
  - Existing form with multiple questions
  - Multiple users with answers submitted for that form
  - K-Means algorithm executed on the form (affinity groups generated)
- **Studied Values**:
  - Exporting affinity group data to JSON format.
- **Studied Effects**:
- - Successful generation of JSON file containing affinity group data
  - Correct formatting of JSON file with appropriate headers and data
  - Accurate representation of users and their assigned affinity groups in the exported file

5 Admin Use Cases

5.1 Import Asnwers from many users

- **Test Object**: Verify that the admin can import answers from many users correctly.
- **Required Data Files**: ”Test Form Imported Answers Expected” (695348328.json), answersToImport.csv and the correspondig users existing in the system.
- **Studied Values**:
  - User Validation
  - Correct association of answers to users.
- **Studied Effects**: First we have to log in as an admin user. Then, we will navigate to the *“Import Answers”* section. Here, we will upload the *answersToImport.csv* file. After the import process is completed, a message should confirm the successful import of answers. Finally, we will verify that each user’s answers have been correctly associated with their profiles by checking a sample of users and their corresponding answers in the system.

5.2 Import Answers from many users - Wrong

- **Test Object**: Verify that if any error occurs during the import of answers from many users, the system handles it gracefully.
- **Required Data Files**: NoProfileExisting.csv WrongFormatAnswer.csv answersToImport.csv
- **Studied Values**:
  - User don’t exist in the system
  - An answer with wrong format
  - Repeated answers for the same user
- **Studied Effects**: First we have to log in as an admin user. Then, we will navigate to the *“Import Answers”* section.
  - Here, we will upload the *answersToImport.csv* file twice. The system should process the file the first time and import the answers successfully. However, when we attempt to upload the same file again, the system should identify the duplicate entries and provide feedback indicating which users already have answers imported, preventing duplication.
  - After that, we will upload the *NoProfileExisting.csv* file. The system should process the file but indicate that some users do not exist in the system and their answers could not be imported.
  - Finally, we will upload the *WrongFormatAnswer.csv* file. The system should identify the incorrectly formatted answers and provide feedback on which entries were problematic, while still importing any correctly formatted answers.

5.3 Clustering algorithm execution with elbow method

- **Test Object**: Verify that executing kmeans creates the correct clusters,plot and affinity groups using elbow method.
- **Required Data Files**: Existing form, with enough answers.
- **Studied Values**:
  - Executing kmeans without introducing K (elbow method).

  - The user is presented with the list of users in every cluster.
  - A 3D plot of the clusters is shown to the user.

- **Studied Effects**:
  - First we login as admin, once on the admin panel we navgate to *"Execute k-means"*.
    Enter the formID for a form with answers like 207838881 (Sports Survey).
    Then the view changes and the affinity groups are presented with the username. A window pops up showing the 3D plot and some basic information , for this form k = 2 is chosen with elbow method.
  - After that we will test that for formID -1190838527 (a form with 3 numeric questions representing x,y,z) the plot is correct, the elbow method choses K = 3.
    No username is shown because the users for the answers are not created, the answer itself is so there's no need to create them.

5.4 Clustering algorithm execution with a set K

- **Test Object**: Verify that executing kmeans creates the correct clusters setting a valid K.
- **Required Data Files**: Existing form, with enough answers.
- **Studied Values**:
  - Executing kmeans introducing K.
  - The user is presented with the list of users in every cluster, there are K clusters.
  - A 3D plot of the clusters is shown to the user.
- **Studied Effects**:
  - First we login as admin, once on the admin panel we navgate to *"Execute k-means"*.
    Enter the formID for a form with answers like 207838881 (Sports Survey) , for K we can chose 5.
    Then the view changes and the affinity groups are presented with the username. A window pops up showing the 3D plot and some basic information, we can confirm that indeed 5 clusters were created.
  - After that we will test that for formID 1157713834 (form for loan test dataset), we can chose K = 2.
    The clusters are shown, the plot is also shown both with 2 clusters.Again the users are no created.

5.5 Clustering algorithm execution with a form that has no answers

- **Test Object**: Verify that an error is shown when there are no answers to the form.
- **Required Data Files**: Existing form that has no answers.
- **Studied Values**:
  - Check the error is shown for elbow method execution.
  - Check the error is shown for set k execution.
- **Studied Effects**:
- First we login as admin, once on the admin panel we navgate to *"Execute k-means"*.
  Enter the formID for a form with no answers like 599496508 (Empty form) ,leave K empty to use elbow method.
  The correct error is shown.

- From the admin panel we navgate to *"Execute k-means"*.
  Enter the formID for a form with no answers like 599496508 (Empty form) ,enter any K, for example 3
  The correct error is shown.

5.6 Evaluation of clustering

- **Test Object**: Verify the evaluation metrics of the clustering algorithm are calculated and displayed correctly.
- **Required Data Files**: Existing form that has enough answers.
- **Studied Values**:
  - Check the evaluation metrics are calculated and displayed correctly for elbow method execution.
  - Check the evaluation metrics are calculated and displayed correctly for set k execution.
- **Studied Effects**:
- First we login as admin, once on the admin panel we navgate to *"Evaluate Clustering"*.
  Enter the formID for a form with enough answers like -1190838527 (3D points) ,leave K empty to use elbow method.
  The result is shown, a high value is expected because the clusters are well difined.
- From the admin panel we navgate to *"Evaluate Clustering"*.
  Enter the formID for a form with enough answers like 1157713834 (Loan dataset) ,enter any K, for example 3
  The result is shown, a lower score is expected for this dataset that has many distinct answers so the clusters might be mixed.