

ACME Requirements Specification: A Case Study
Version 6.1
Alvin J. Alexander
<http://www.devdaily.com>
Jan. 28, 2004
Copyright DevDaily Interactive (devdaily.com), 2004.
All Rights Reserved.

**Revised by Diego Clerissi & Gianna Reggio applying the Disciplined use cases with
mockups method**

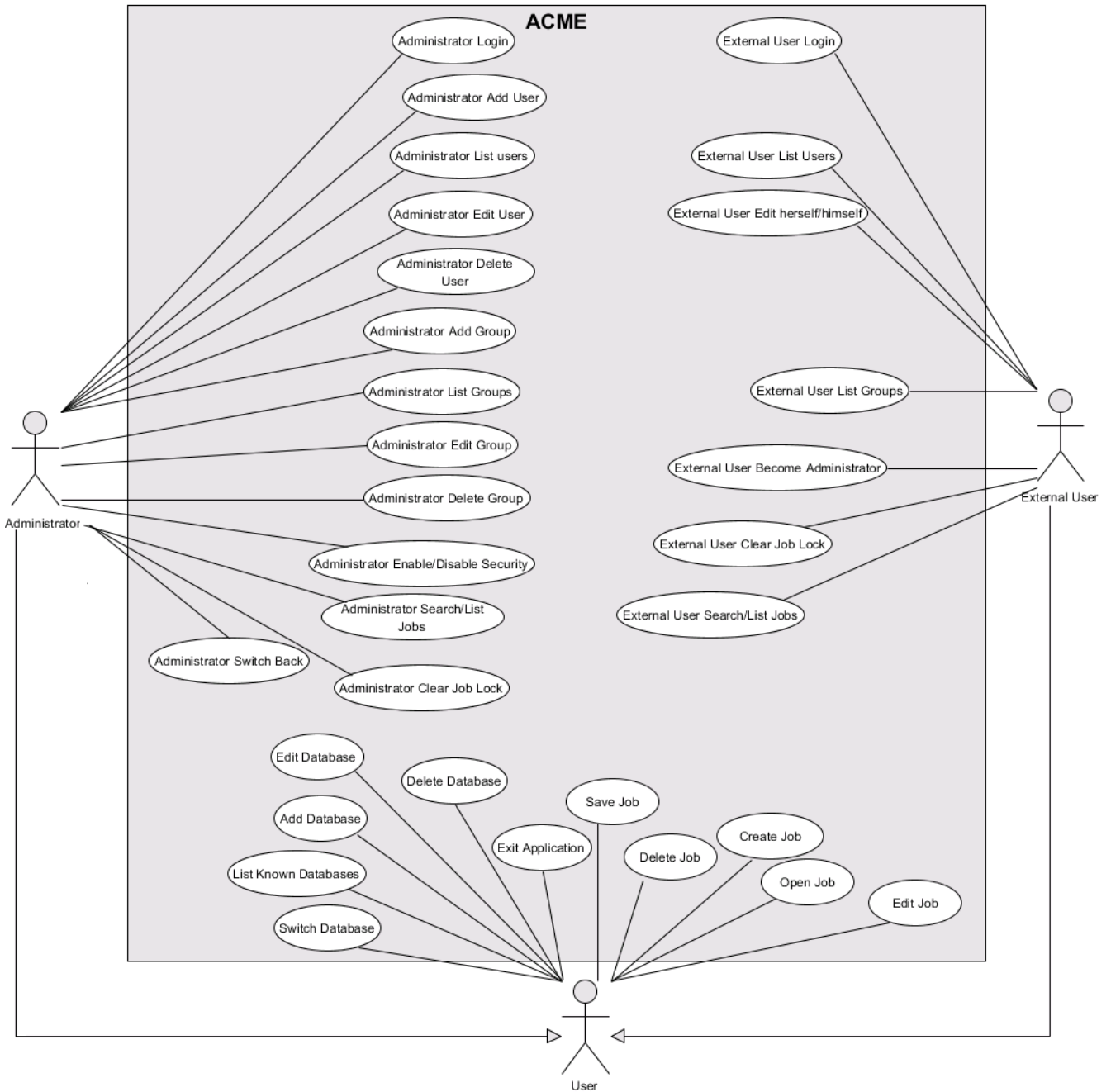
Table of Contents

Use Case Diagram.....	3
Use case: External User Login.....	4
Use case: Administrator Login.....	6
Use Case: Administrator Add User.....	8
Use Case: Administrator List Users.....	13
Use Case: External User List Users.....	14
Use Case: Administrator Edit User.....	15
Use Case: External User Edit Personal Info.....	19
Use Case: Administrator Delete User.....	22
Use Case: Administrator Add Group.....	24
Use Case: Administrator List Groups.....	28
Use Case: External User List Groups.....	30
Use Case: Administrator Edit Group.....	32
Use Case: Administrator Delete Group.....	35
Use Case: Administrator Enable/Disable Security.....	37
Use Case: External User Become Administrator.....	38
Use Case: Administrator Switch Back.....	42
Use Case: Create Job.....	44
Use Case: Administrator Search/List Jobs.....	49
Use Case: External User Search/List Jobs.....	52
Use Case: Open Job.....	55
Use Case: Edit Job.....	59
Use Case: Save Job.....	63
Use Case: Delete Job.....	65
Use Case: Administrator Clear Job Lock.....	68
Use Case: External User Clear Job Lock.....	72
Use Case: Exit Application.....	77
Use Case: Add Database.....	79
Use Case: Delete Database.....	84
Use Case: Edit Database.....	87
Use Case: List Known Databases.....	90
Use Case: Switch Database.....	91
Glossary.....	94
Data Entry.....	94
System (Abstract State).....	96

In this document we show the results of the application of our method DUSM (*Disciplined Use cases with Screen Mockups*) applied to a real case study. Our goal is to make use cases-based requirements specification clearer (i.e. reducing ambiguities, inconsistencies and incompleteness) by adopting a glossary of terms, connecting to use cases steps a number of screen mockups to represent UI interfaces, and following a strict template/description structure. We chose ACME specifications since it is complete and use cases-based, presents a simple domain, and includes some problems that our method could fix (e.g. describing the UI in a textual way only). In the following, we present the use case diagram of our case study and the use cases after DUSM.

Use Case Diagram

All the use case have the "User Goal" level, thus for simplicity we have not reported it in the picture.



Use case: External User Login

This use case defines the process of logging the External User into the ACME application.

Notes

- All user types can log in multiple workstations, having multiple ACME Editors open on their desktop (each with a different job)
- The “default” database is populated with a list of default customers
- The first time ACME is launched, Security_Enabled* is false

Level userGoal

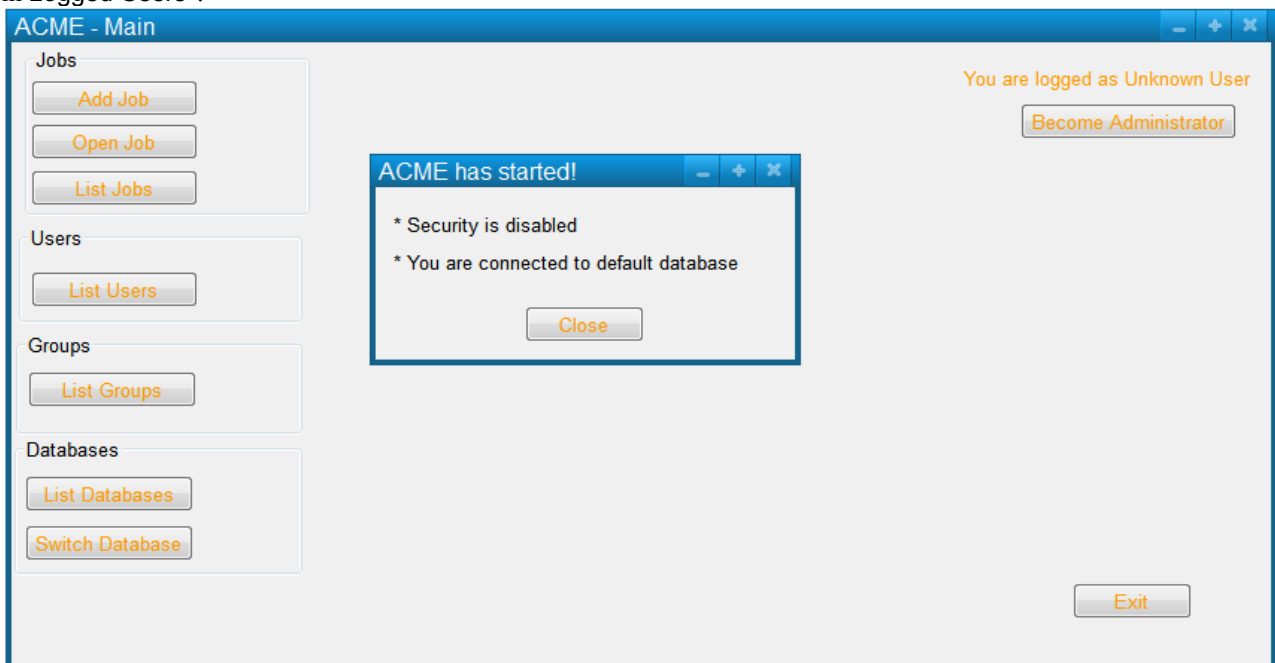
Priority 1

Frequency frequently

Primary Actor External User

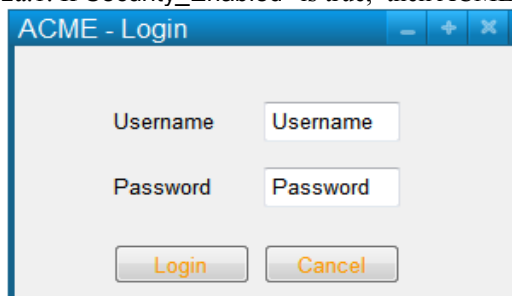
Main success scenario

1. The External User starts ACME.
2. If Security_Enabled* is false, then ACME informs that it has started; the External User is logged as Unknown User in Logged Users*.

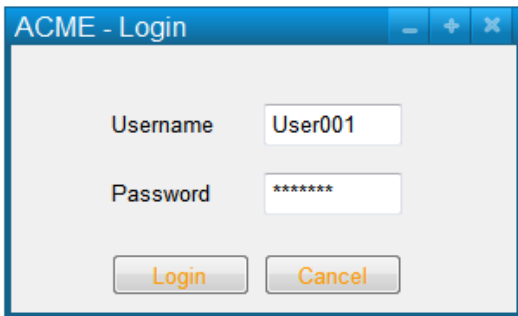


Extensions

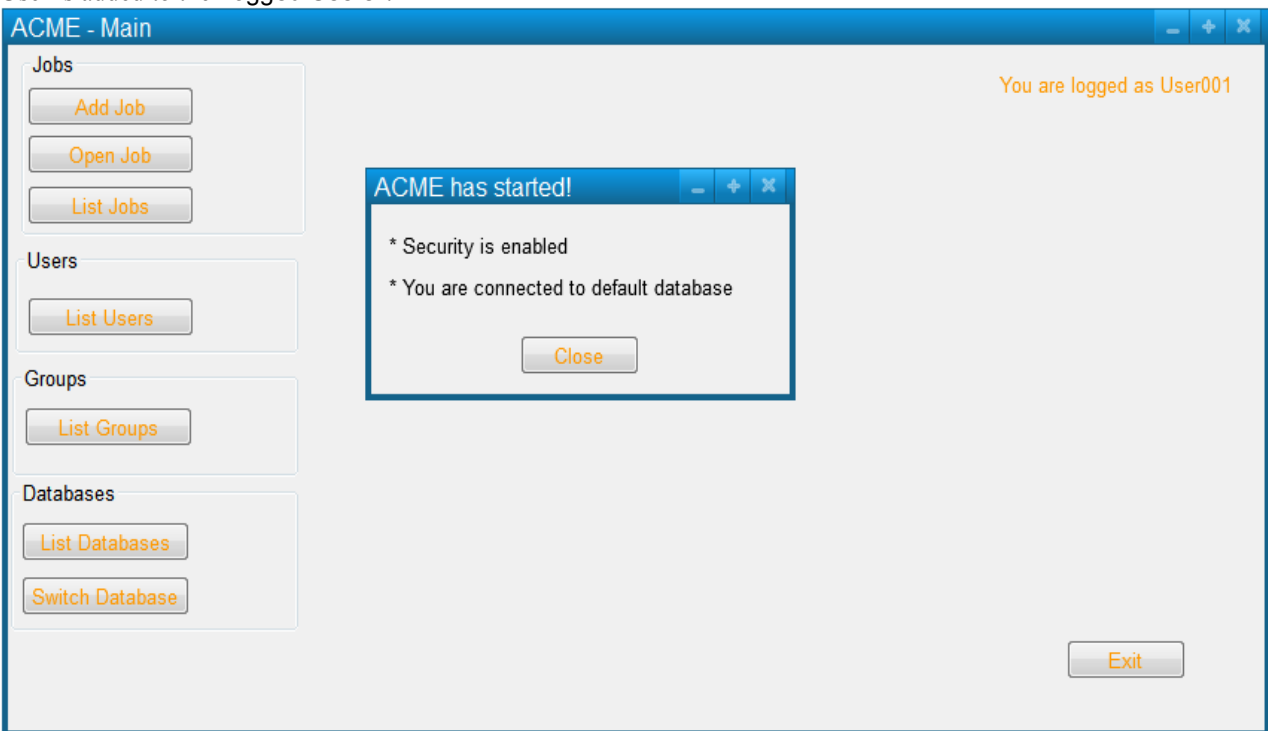
- 2a.1. If Security_Enabled* is true, then ACME requests to log in.



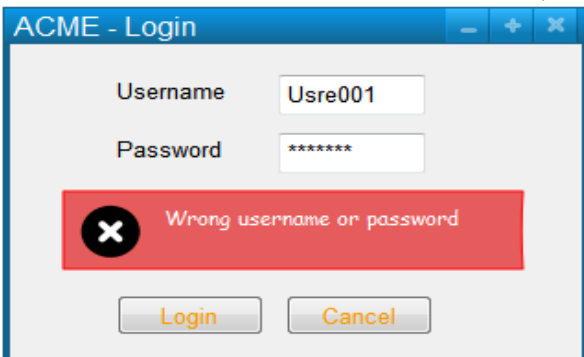
2a.3. The External User enters Username* and Password*.



2a.4. If Username* and Password* are valid, then ACME informs the External User that it has started; the External User is added to the Logged Users*.



2a.4a.1. If Username* and Password* are invalid, then ACME informs that the given data is wrong. The use case fails



Use case: Administrator Login

Goal This use case allows the Administrator to log into the ACME application.

Level userGoal

Priority 1

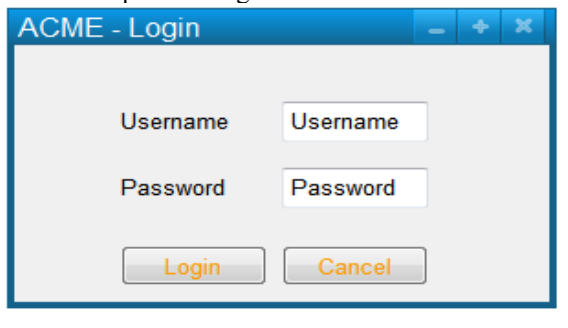
Frequency periodically

Primary Actor Administrator

Pre-Conditions Security_Enabled* is true

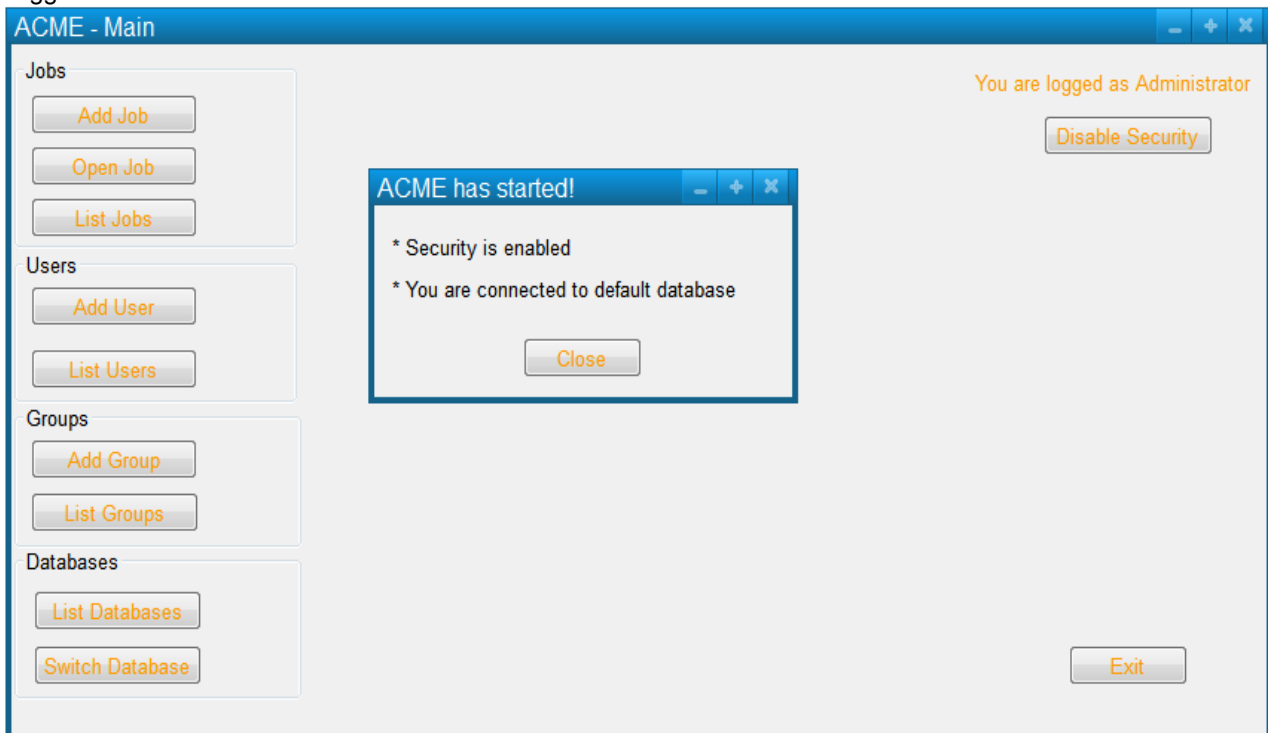
Main success scenario

1. The Administrator starts ACME.
2. ACME requests to log in.



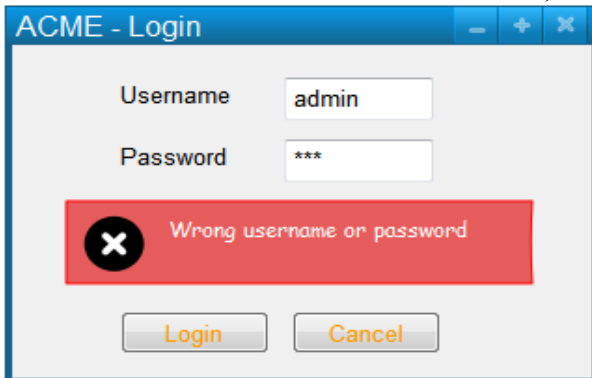
4. The Administrator enters Username* and Password*.

5. If Username* and Password* are valid, then ACME informs that it has started; the Administrator is added to Logged Users*.



Extensions

5a.1. If the Username* and Password* are invalid, then ACME informs that the given data is wrong. The use case fails.



Use Case: Administrator Add User

Goal This use case describes the process of adding a new user to the ACME system.

Level userGoal

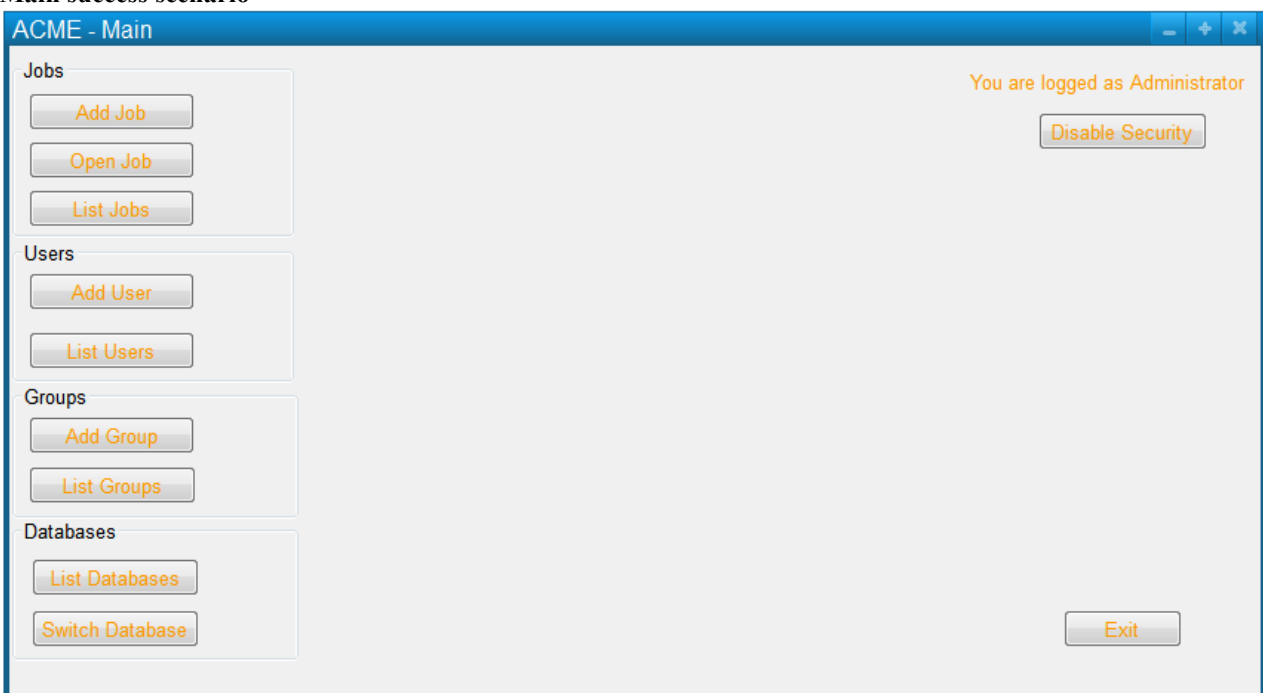
Priority 1

Frequency frequently

Primary Actor Administrator

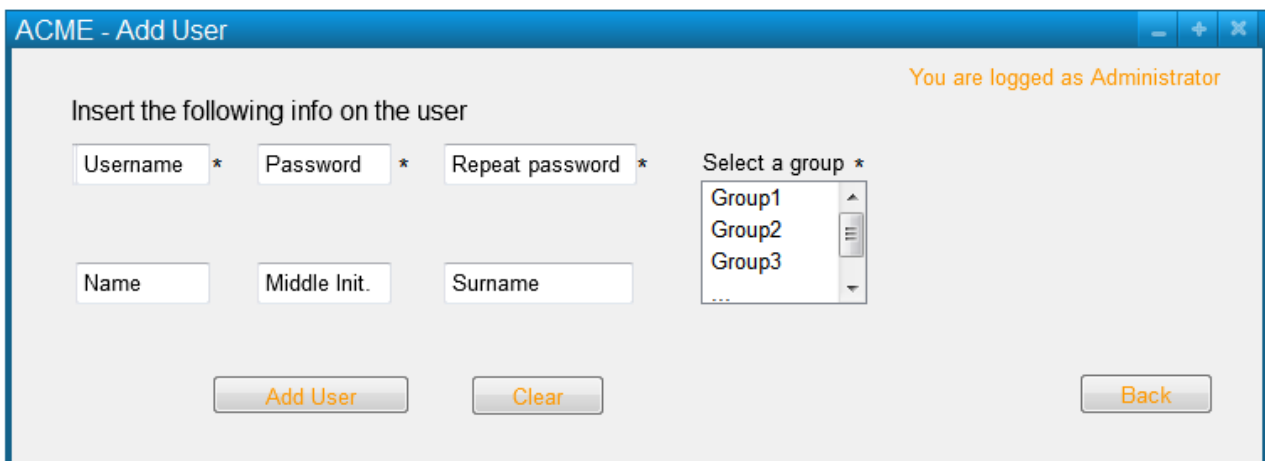
Pre-Conditions Administrator_Logged* is true

Main success scenario



1. The Administrator requests to add a new user.

2. ACME prompts for the User_Info*.



3. The Administrator enters the User_Info* and submits the given data.

The screenshot shows a window titled "ACME - Add User" with a blue header. In the top right corner, it says "You are logged as Administrator". The main area contains the text "Insert the following info on the user". There are three input fields in the first row: "User002" with an asterisk, "abc123" with an asterisk, and "abc123" with an asterisk. Below these are three more input fields: "John", an empty field, and "Smith". To the right is a dropdown menu labeled "Select a group *" with options "Group1", "Group2", "Group3", and "...". At the bottom, there are three buttons: "Add User", "Clear", and "Back".

4. If User_Info* is complete, Username* and Password* are valid, and Username* is not used in Users*, then ACME confirms the creation; a new user characterized by User_Info* is added to Users*.

This screenshot is similar to the previous one, but a dialog box titled "ACME - Success!" is overlaid in the center. The dialog box contains the text "User002 has been added to the System and enrolled to Group1!" and an "Ok" button. The background form is partially obscured by the dialog box.

Extensions

4a.1. If User_Info* is incomplete, then ACME informs of which required fields are missing. The use case fails.

The screenshot shows the "ACME - Add User" form with a red error message box on the right. The error message says "Incomplete information. The password is missing!". In the form, the second input field (password) is empty, while the others contain "User002", "abc123", "John", and "Smith". The "Add User" button is disabled.

4b.1. If Username* is already in use, then ACME informs that he must choose a different name. The use case fails.

ACME - Add User

You are logged as Administrator

Insert the following info on the user

User# * Password * Confirm Password *

Group *

First Name * Last Name *

Buttons: Add User, Clear, Back

Error Message: Username is already in use. Choose a different one!

4c.1. If Username* is invalid, then ACME informs that the given data is wrong and displays the valid naming criteria. The use case fails.

ACME - Edit User

You are logged as Administrator

Insert the following info on the user

User# * Password * Confirm Password *

Group *

First Name * Last Name *

Buttons: Save Changes, Clear, Back

Error Message: Username is not valid. It must be up to 20 characters, containing only letters, numbers, _, -, and blank spaces.

4d.1. If Password* is invalid, then ACME informs that the given data is wrong, and valid password criteria is displayed. The use case fails.

4e.1. If the two passwords are not the same, then ACME informs that the given data is wrong. The use case fails.

The screenshot shows a web application window titled "ACME - Edit User". The window has a blue header bar with standard window controls (minimize, maximize, close). The main content area is light gray and contains the following elements:

- Text: "Insert the following info on the user" (top left) and "You are logged as Administrator" (top right).
- Form fields: A row of three text boxes containing "User001", "ABC123", and "123ABC", each followed by an asterisk (*). Below this is another row with "John", an empty box, and "Black".
- Dropdown menu: A "Select a group *" dropdown menu with options "Group1", "Group2", "Group3", and "...".
- Buttons: "Save Changes" (orange text), "Clear" (gray text), and "Back" (orange text).
- Validation message: A red box with a black 'X' icon and the text "Password and repeated password are not the same!".

Post-Conditions

A new user characterized by User_Info* is added to Users*

Use Case: Administrator List Users

Goal This use case allows to list the users on the ACME system

Level userGoal

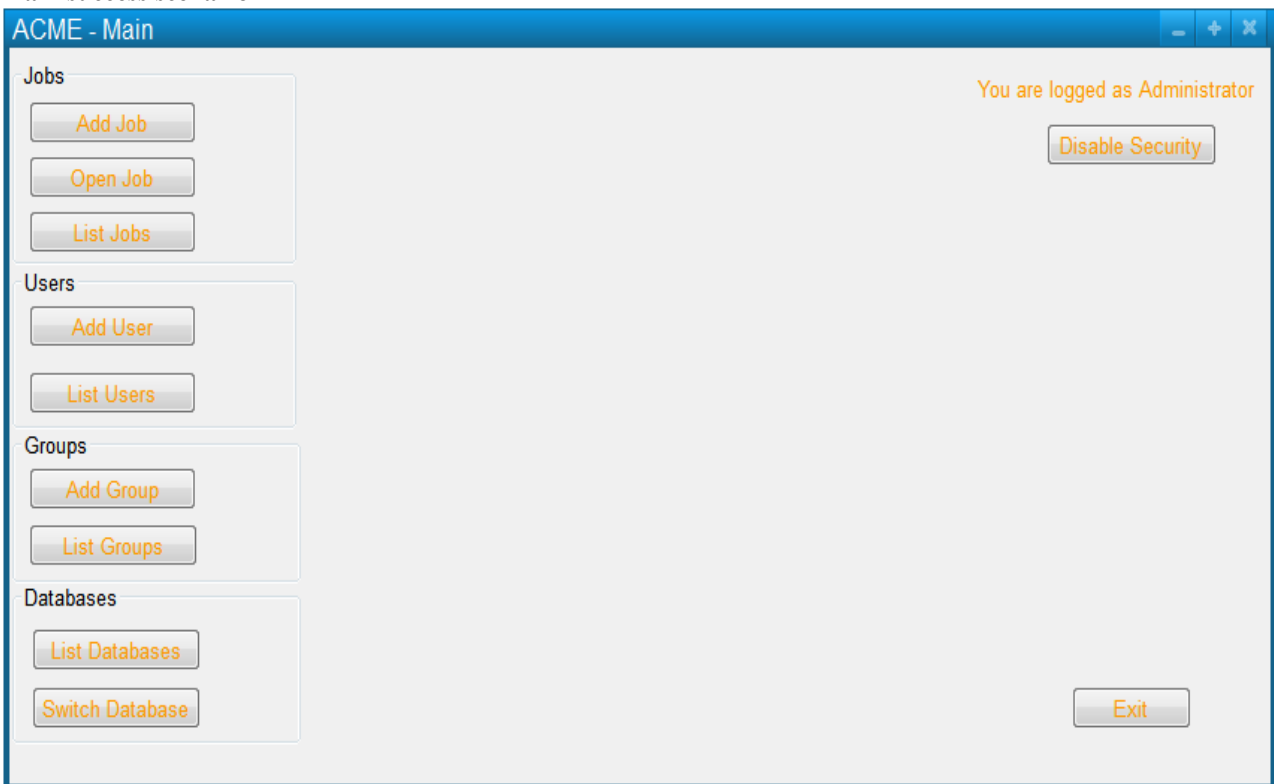
Priority 3

Frequency periodically

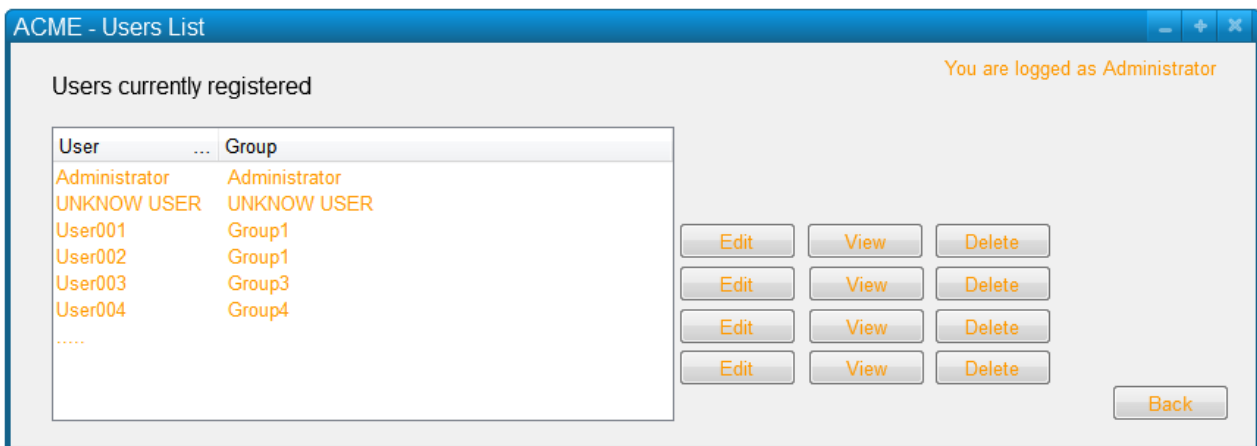
Primary Actor Administrator

Pre-Conditions Administrator_Logged* is true

Main success scenario



1. The Administrator requests to list the current users.
2. ACME displays, for every user in Users*: Username*, group name, and the options to edit, view and delete users.



Post-Conditions A list of users is displayed.

Use Case: External User List Users

Goal This use case allow an external user to list the registered users.

Level userGoal

Priority 3

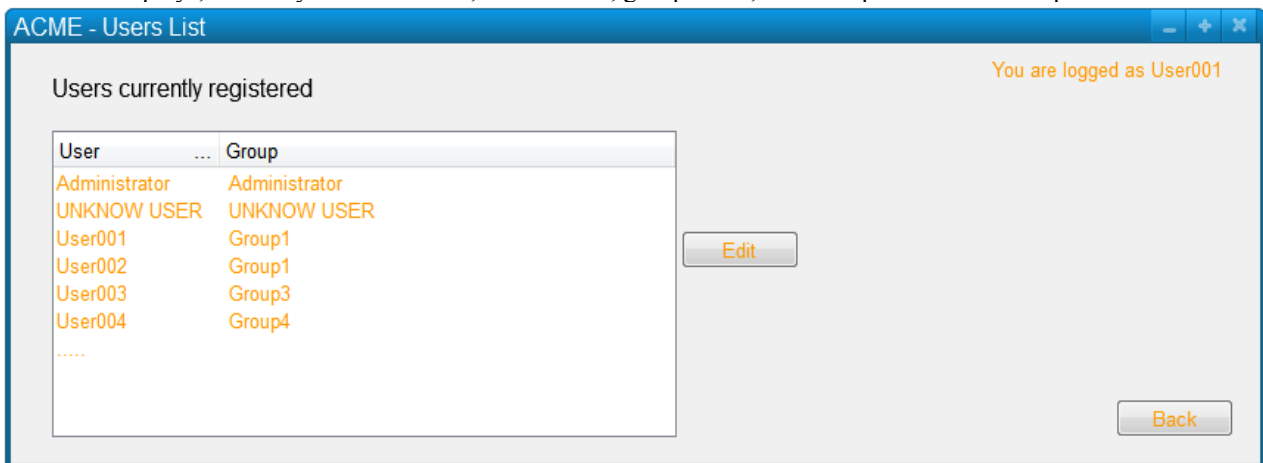
Frequency periodically

Primary Actor External User

Pre-Conditions The External User belongs to Logged Users*

Main success scenario

1. The External User requests to list the users.
2. ACME displays, for every user in Users*, Username*, group name, and the options to edit the personal data.



Post-Conditions A list of users is displayed.

Use Case: Administrator Edit User

Goal This use case allows the administrator to edit a user.

Level userGoal

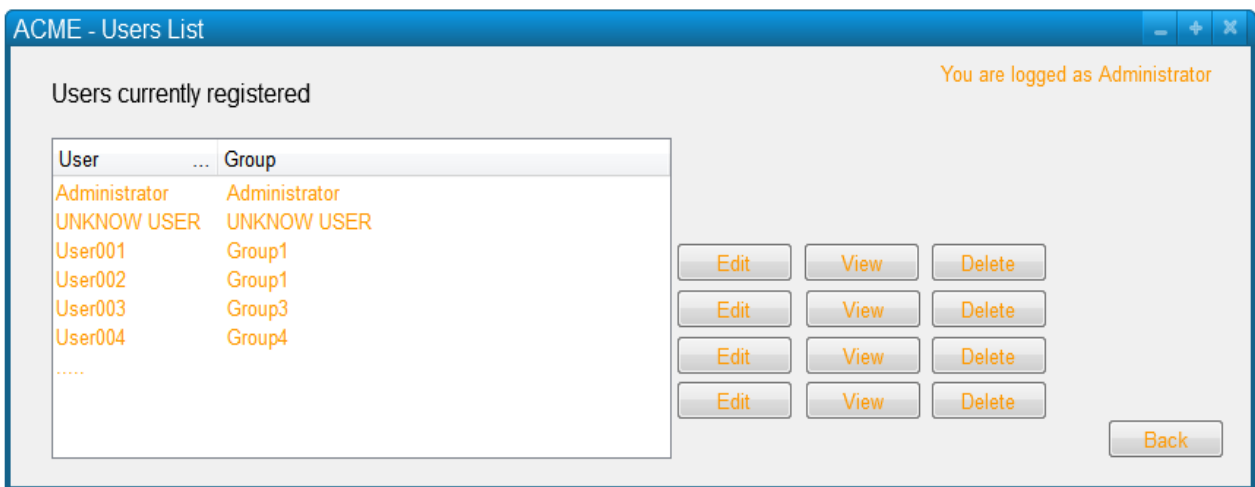
Priority 4

Frequency periodically

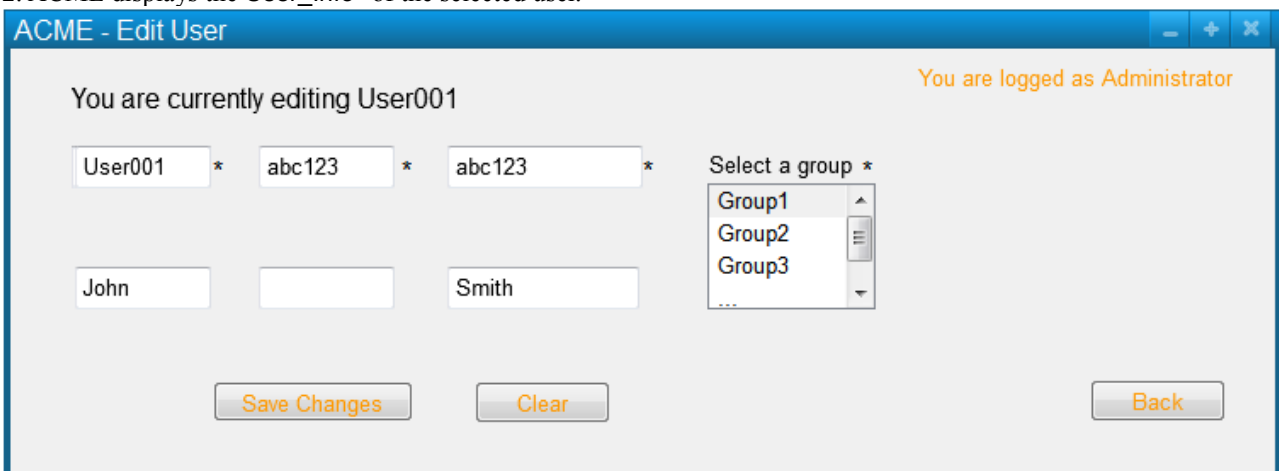
Primary Actor Administrator

Pre-Conditions Administrator_Logged* is true and a list of users is displayed.

Main success scenario



1. The Administrator requests to edit a user.
2. ACME displays the User_Info* of the selected user.



3. The Administrator enters all desired data and confirms the changes.

The screenshot shows the 'ACME - Edit User' window. At the top, it says 'You are currently editing User001' and 'You are logged as Administrator'. The form contains several input fields: 'User001' (with an asterisk), 'ABC123' (with an asterisk), 'ABC123' (with an asterisk), 'John', and 'Black'. A dropdown menu labeled 'Select a group *' is open, showing 'Group1', 'Group2', and 'Group3'. At the bottom, there are three buttons: 'Save Changes', 'Clear', and 'Back'.

4. If User_Info* is complete, Username* and Password* are valid, Username* is not used in Users*, and the two passwords are the same, then ACME displays the users list; User_Info* are updated with the new data.

The screenshot shows the 'ACME - Users List' window. At the top, it says 'You are logged as Administrator'. Below this, there is a table titled 'Users currently registered' with columns 'User' and 'Group'. The table contains the following rows: Administrator (Administ), UNKNOW USER (UNKNOV), User001 (Group1), User002 (Group1), and User004 (Group4). To the right of the table are buttons for 'View' and 'Delete' for each user. A modal dialog box titled 'ACME - Success!' is open in the center, displaying the message 'User001 has been successfully updated!' and an 'Ok' button. At the bottom right, there is a 'Back' button.

Extensions

4a.1. If User_Info* is incomplete, then ACME informs of which required fields are missing. The use case fails.

The screenshot shows the 'ACME - Edit User' window. At the top, it says 'You are currently editing User001' and 'You are logged as Administrator'. The form contains several input fields: 'User001' (with an asterisk), a field with a vertical bar '|', 'ABC123' (with an asterisk), 'John', and 'Black'. A dropdown menu labeled 'Select a group *' is open, showing 'Group1', 'Group2', and 'Group3'. A red error message box is displayed on the right side, containing a black 'X' icon and the text 'Incomplete information. The password is missing!'. At the bottom, there are three buttons: 'Save Changes', 'Clear', and 'Back'.

4b.1. If the Username* is already in use in Users*, then ACME informs that he must choose a different name. The use case fails.

The screenshot shows a web browser window titled "ACME - Edit User". The page content includes a status bar at the top right that says "You are logged as Administrator". Below this, a message states "You are currently editing User001". The form contains several input fields: "User002" (with an asterisk), "ABC123" (with an asterisk), "ABC123" (with an asterisk), "John", a blank field, and "Black". To the right of these fields is a dropdown menu labeled "Select a group *" with options "Group1", "Group2", "Group3", and "...". At the bottom of the form are three buttons: "Save Changes", "Clear", and "Back". A red error message box on the right side of the form contains a black 'X' icon and the text: "Username is already in use. Choose a different one!"

4c.1. If the Username* is invalid, then ACME informs that the given data is wrong, and displays the valid naming criteria. The use case fails.

The screenshot shows a web browser window titled "ACME - Edit User". The page content includes a status bar at the top right that says "You are logged as Administrator". Below this, a message states "You are currently editing User001". The form contains several input fields: "User#2" (with an asterisk), "ABC123" (with an asterisk), "ABC123" (with an asterisk), "John", a blank field, and "Black". To the right of these fields is a dropdown menu labeled "Select a group *" with options "Group1", "Group2", "Group3", and "...". At the bottom of the form are three buttons: "Save Changes", "Clear", and "Back". A red error message box on the right side of the form contains a black 'X' icon and the text: "Username is not valid. It must be up to 20 characters, containing only letters, numbers, -, ., and blank spaces."

4d.1. If the Password* is invalid, then ACME informs that the given data is wrong, and displays the valid password criteria. The use case fails.

4e.1. If the two passwords are not the same, then ACME informs that the given data is wrong. The use case fails

The screenshot shows a web browser window titled "ACME - Edit User". The page content includes a status bar at the top right that says "You are logged as Administrator". Below this, a message states "You are currently editing User001". The form contains several input fields: "User001" (with an asterisk), "ABC123" (with an asterisk), "123ABC" (with an asterisk), "John", a blank field, and "Black". To the right of these fields is a dropdown menu labeled "Select a group *" with options "Group1", "Group2", "Group3", and "...". At the bottom of the form are three buttons: "Save Changes", "Clear", and "Back". A red error message box on the right side of the form contains a black 'X' icon and the text: "Password and repeated password are not the same!"

Use Case: External User Edit Personal Info

Actor This use case allows an external user self-editing hers/his information

Level userGoal

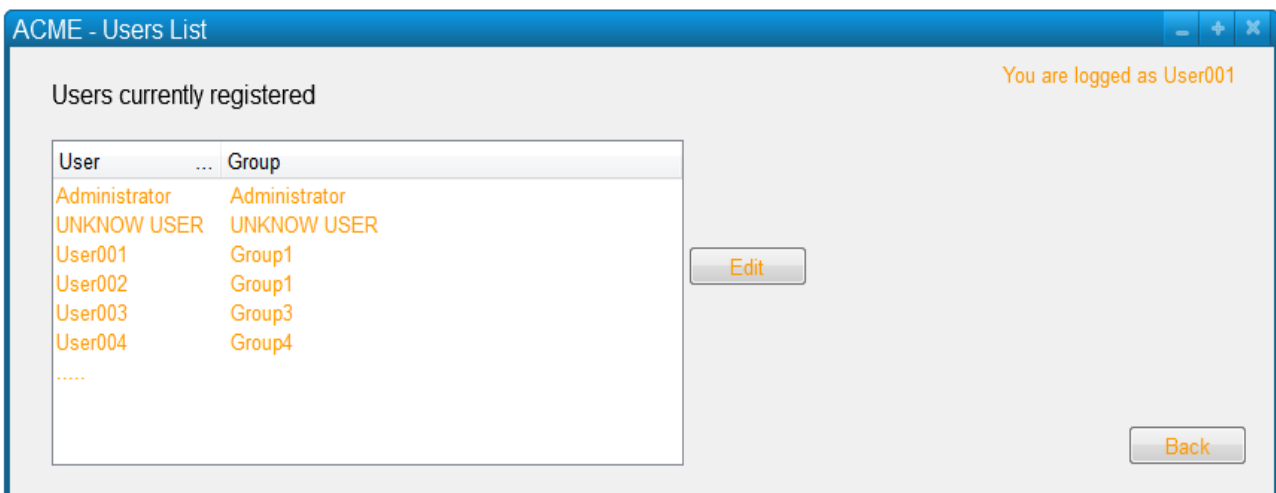
Priority 4

Frequency periodically

Primary Actor External User

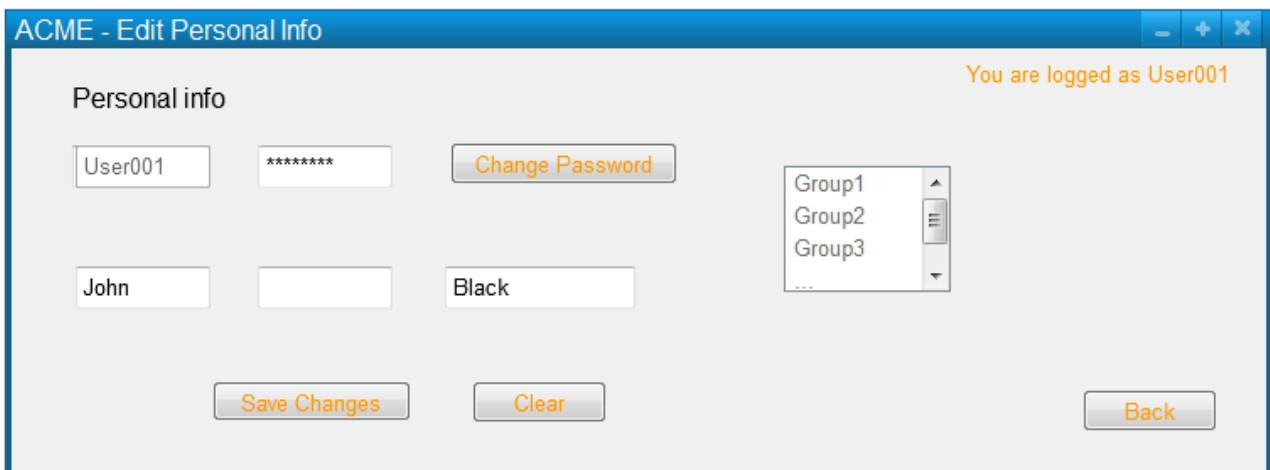
Pre-Conditions The External User belongs to Logged Users*, and a list of users is displayed

Main success scenario



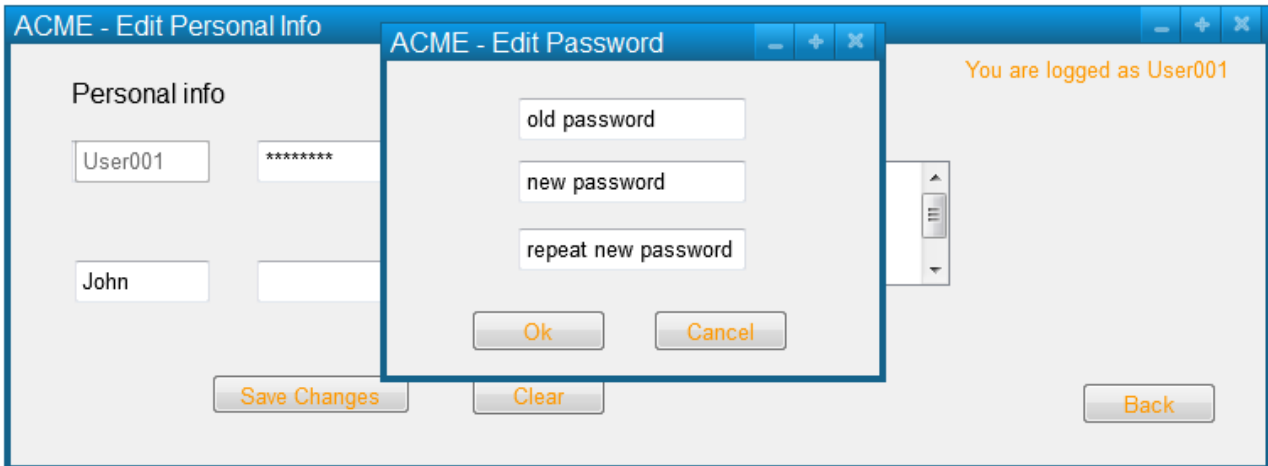
1. The External User requests to edit his personal info.

2. ACME displays the External User User_Info*.



3. The External User requests to change his Password*.

4. ACME prompts the External User for his old and new Password*.

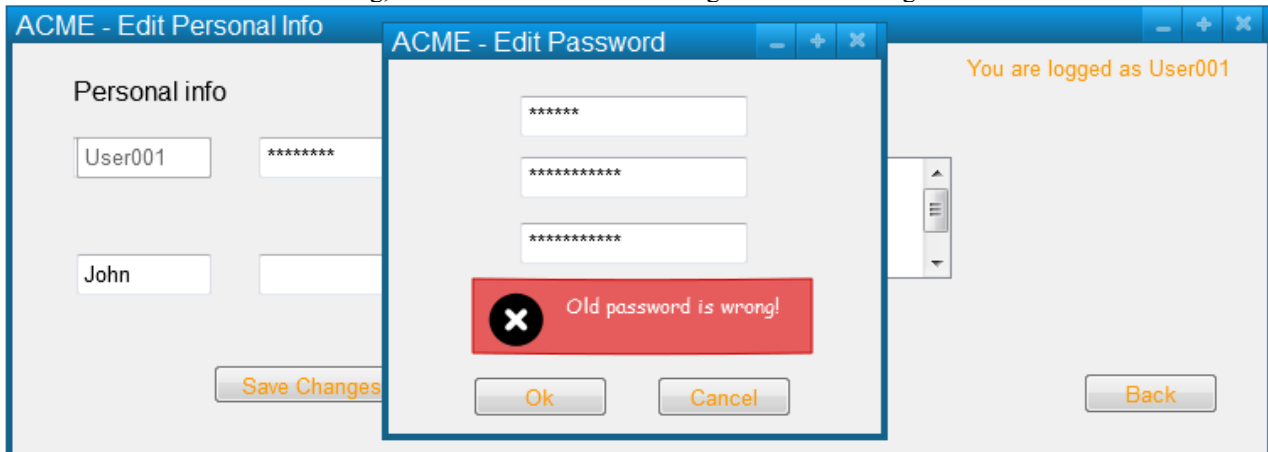


5. The External User enters old and new Password* and confirms the changes.

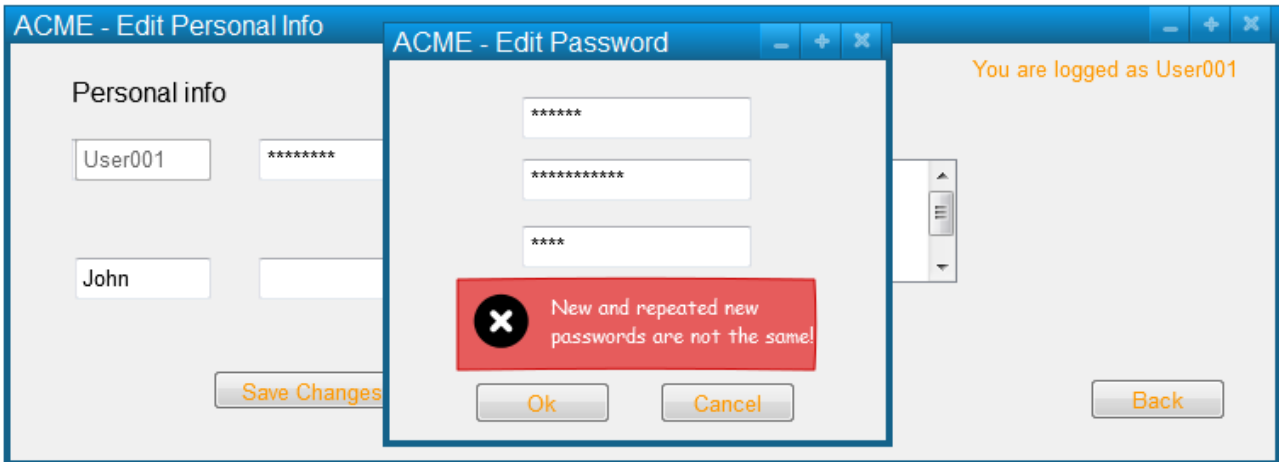
6. If the old Password* is correct and the two Password* are identical and valid, then ACME displays the user lists; the User_Info* in Users* are updated.

Extensions

6a.1. If the old Password* is wrong, then ACME informs that the given data is wrong. The use case fails.



6b.1 If new and repeated new Password* are not equal, then ACME informs that the given data is wrong. The use case fails.



6c.1 If the new Password* is invalid, then ACME informs that the given data is wrong, and valid Password* criteria is displayed. The use case fails.

Use Case: Administrator Delete User

Goal This use case allows the administrator to delete some users from ACME.

Notes

Note that in this use case clearing the locks is an “all or nothing” proposition. Selectively clearing job locks is not available from this point in the application

Level userGoal

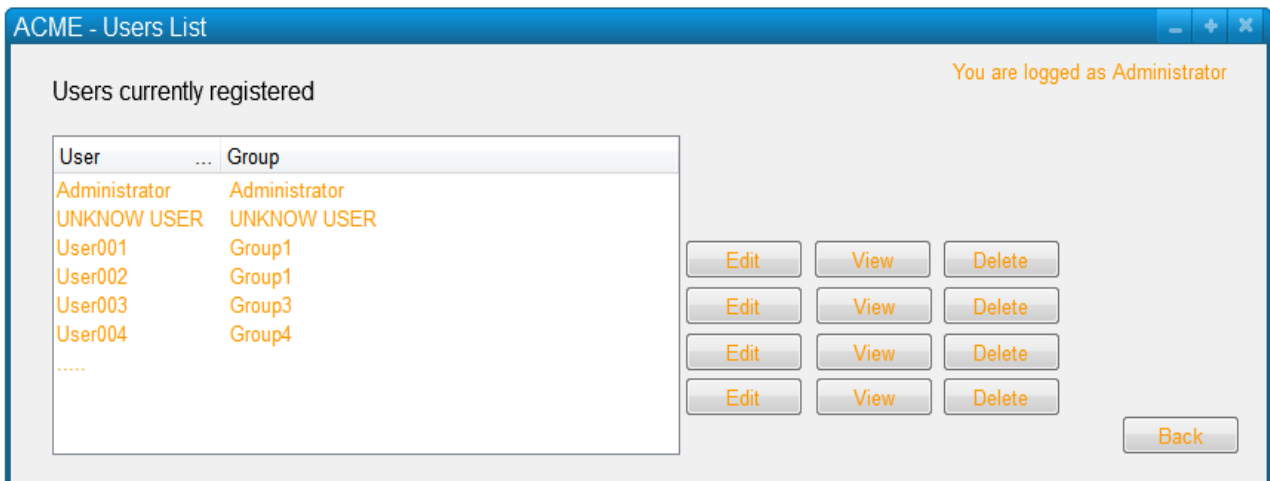
Priority 4

Frequency exceptionally

Primary Actor Administrator

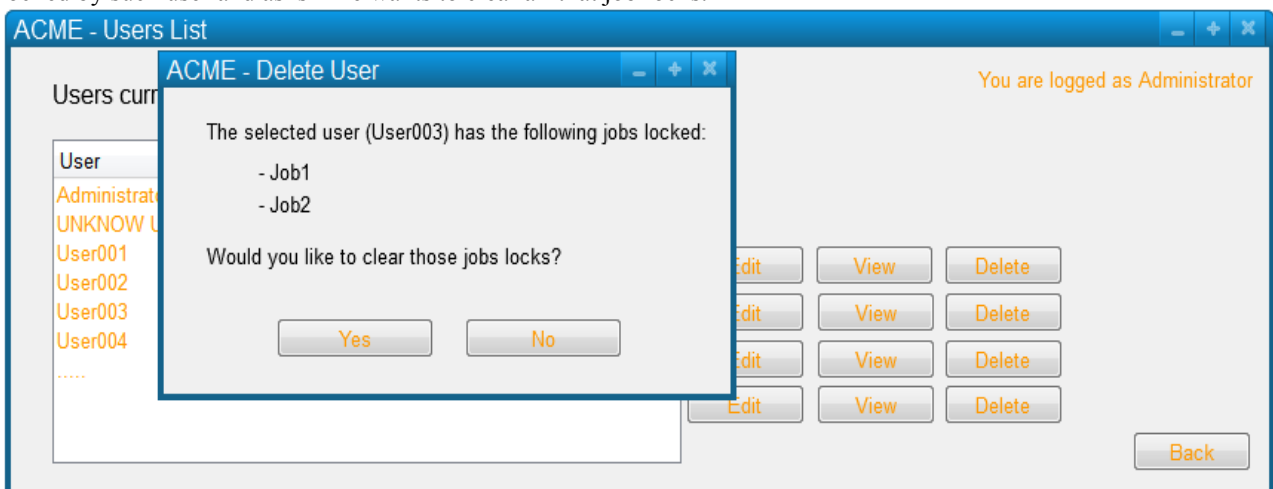
Pre-Conditions Administrator_Logged* is true, and a list of users is displayed

Main success scenario



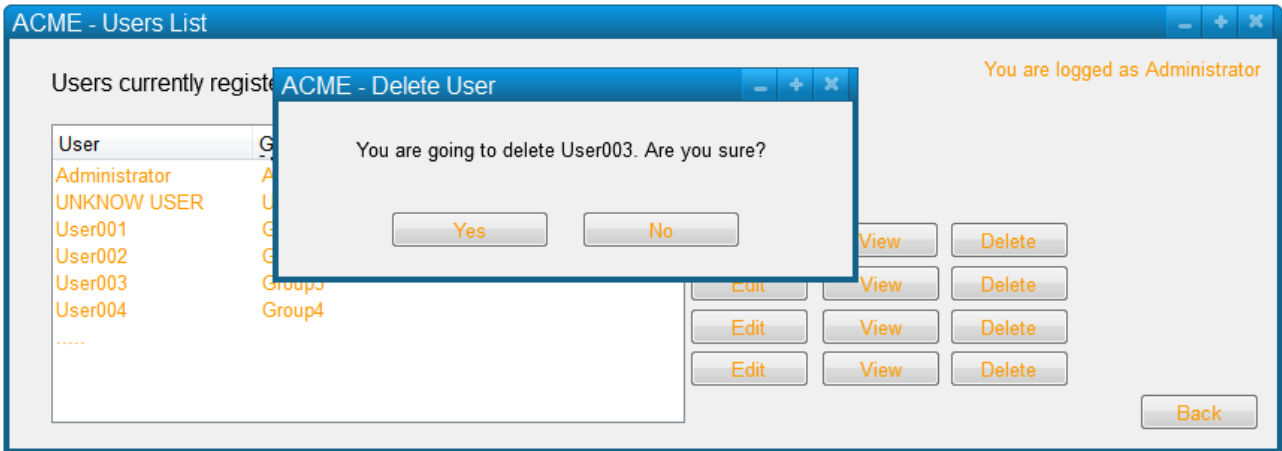
1. The Administrator requests to delete a user.

2. If the selected user has any job locks in Jobs*, then ACME informs the Administrator showing the list of the jobs locked by such user and asks if he wants to clear all that job locks.



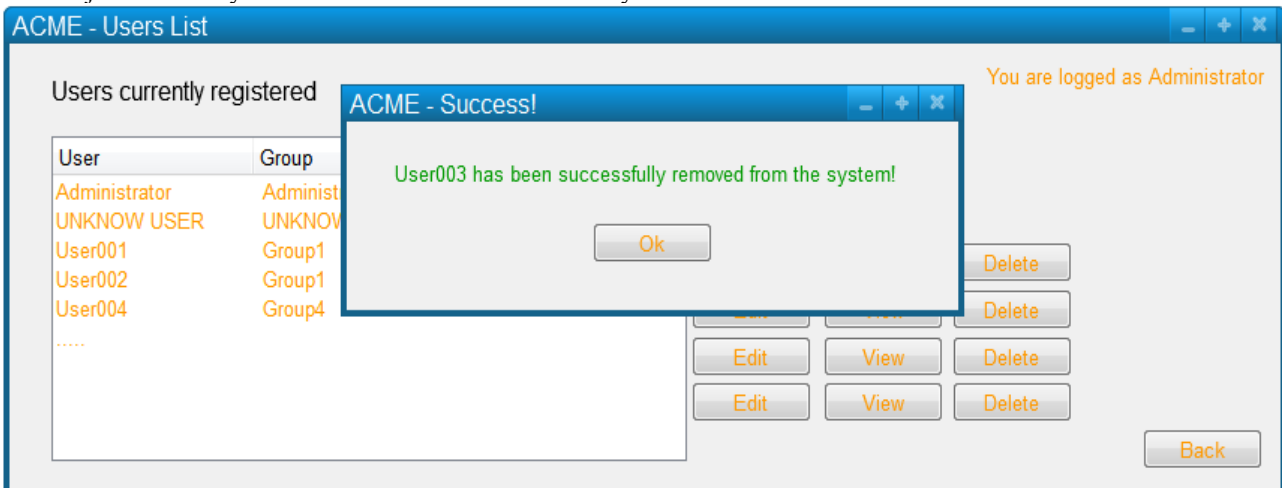
3. The Administrator requests to clear all job locks.

4. ACME asks the Administrator for a confirmation to delete the selected user.



5. The Administrator confirms the intention to delete the selected user.

6. ACME confirms the deletion of the user; the user is deleted from Users*; any lock of the user is removed in Jobs*, and the jobs owned by the user in Jobs* are now owned by Unknown User.



Extensions

2a.1. If the selected user has no job locks in Jobs*, then the use case continues at step 4.

3a.1. The Administrator refuses to clear all the locks. The use continues at step 4.

5a.1. The Administrator aborts the operation. The use case fails.

Post-Conditions Any job created by a deleted user is now owned by Unknown User.

Use Case: Administrator Add Group

Goal This use case allows the administrator to add a new group to the ACME system.

Notes Only the Administrator can CRUD group information

Level userGoal

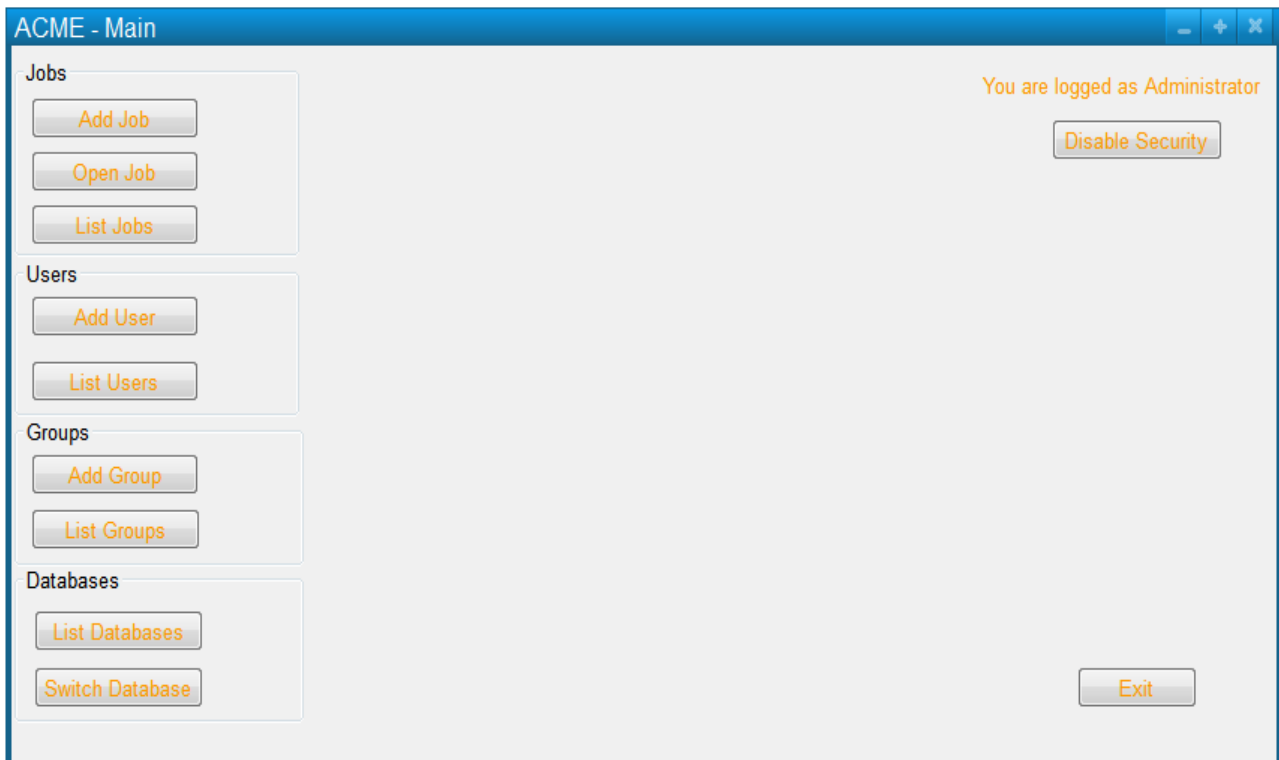
Priority 1

Frequency frequently

Primary Actor Administrator

Pre-Conditions Administrator_Logged* is true

Main success scenario



1. The Administrator requests to add a new group to ACME.

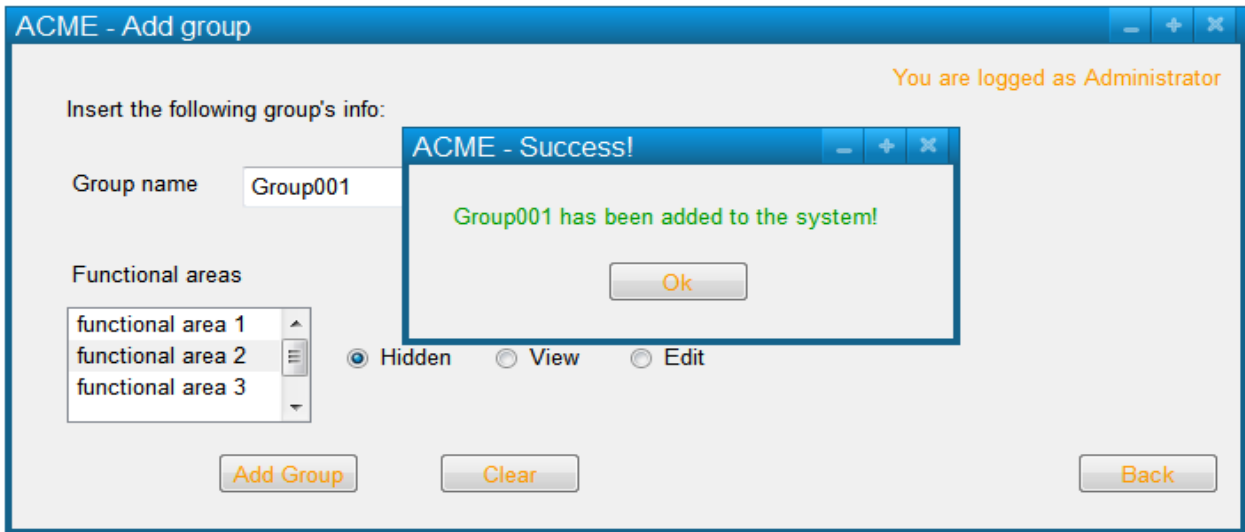
2. ACME prompts the Administrator for the Group Info*.

The screenshot shows a window titled "ACME - Add group" with a blue header bar. In the top right corner, it says "You are logged as Administrator". The main content area has the text "Insert the following group's info:". Below this, there is a "Group name" label followed by a text input field containing "group name" and an asterisk. Underneath, there are two sections: "Functional areas" and "Access Level". The "Functional areas" section has a list box with three items: "functional area 1", "functional area 2", and "functional area 3". The "Access Level" section has three radio buttons: "Hidden" (which is selected), "View", and "Edit". At the bottom of the form, there are three buttons: "Add Group", "Clear", and "Back".

3. The Administrator enters the required information and confirms

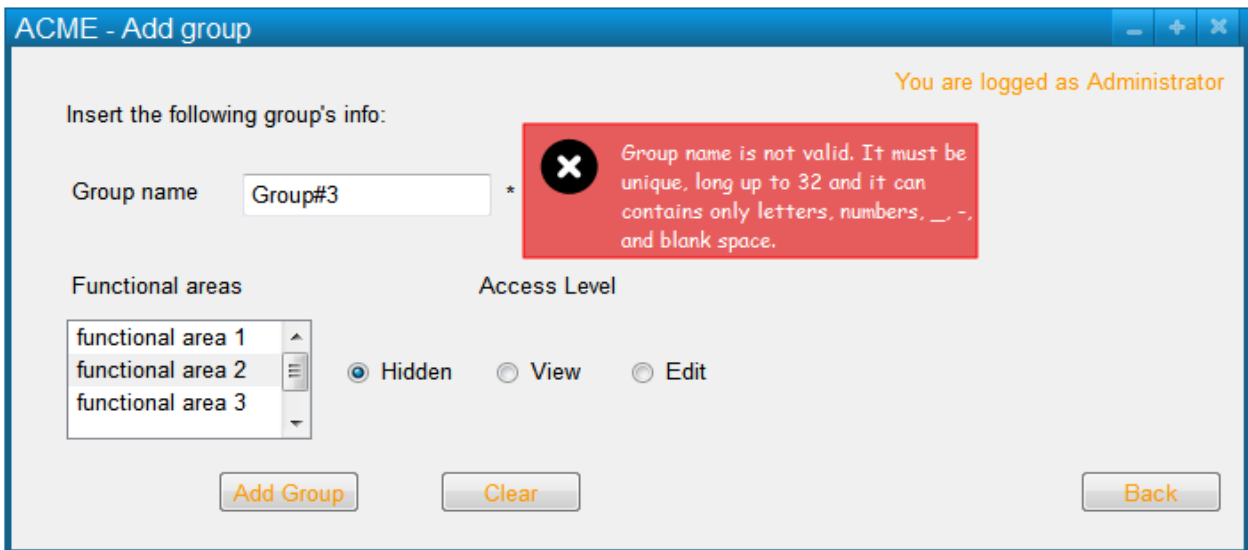
This screenshot is identical to the previous one, but the text input field for "Group name" now contains "Group001". The "Hidden" radio button under "Access Level" remains selected. All other elements, including the "Functional areas" list and the "Add Group", "Clear", and "Back" buttons, are the same as in the previous screenshot.

4. If the Group Name* is valid, it is not already in use in Groups*, and the Access Level* for all Functional Areas* is selected, then ACME confirms the addition; a group characterized by the given information is added to Groups*.



Extensions

4a.1. If the Group Name* is invalid, then ACME informs the Administrator, and shows the proper group naming convention. The use case fails.



4b.1. If the Group Name* is already in use in Groups*, then ACME informs that the given data is wrong. The use case fails.

The screenshot shows a window titled "ACME - Add group" with a blue header bar. In the top right corner of the window, it says "You are logged as Administrator". Below the header, the text "Insert the following group's info:" is displayed. There is a text input field labeled "Group name" containing the text "Group001". To the right of this field is a red error box with a black 'X' icon and the text "Group name is already in use. Choose a different one!". Below the "Group name" field, there are two sections: "Functional areas" and "Access Level". The "Functional areas" section contains a list box with three items: "functional area 1", "functional area 2", and "functional area 3". The "Access Level" section contains three radio buttons: "Hidden" (which is selected), "View", and "Edit". At the bottom of the window, there are three buttons: "Add Group", "Clear", and "Back".

Post-Conditions The group is created on the system, with associated access levels for the functional areas.

Use Case: Administrator List Groups

This use case describes the process of listing groups in the ACME system by the Administrator.

Notes

This use case can take the Administrator to the Edit Group use case. Users cannot be deleted from groups here, that must be done with the use case “Edit User” assigning him/her to a different group.

Level userGoal

Priority 3

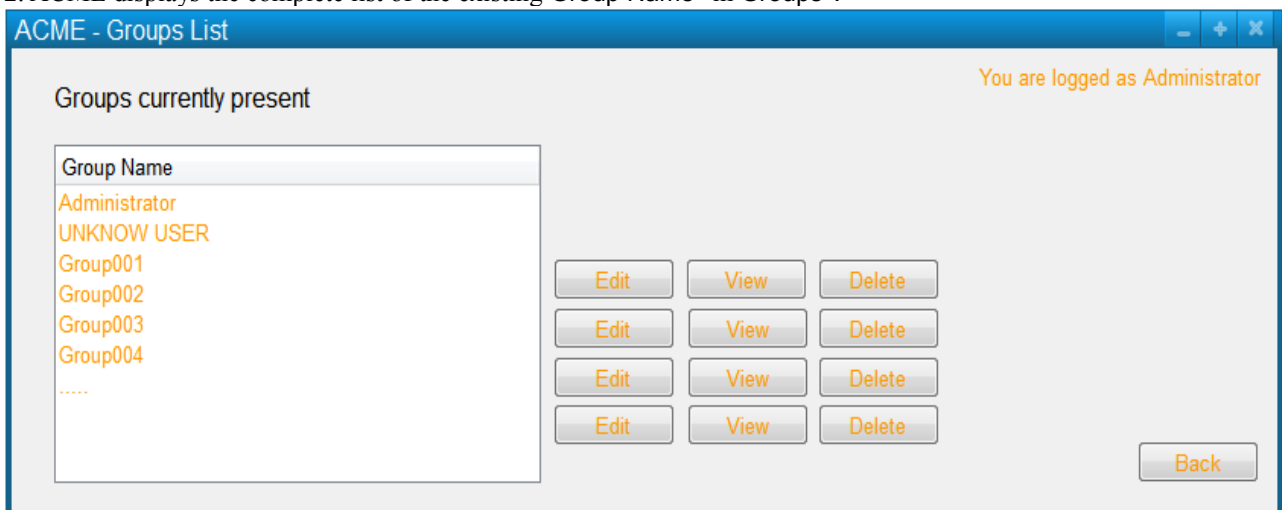
Frequency periodically

Primary Actor Administrator

Pre-Conditions Administrator_Logged* is true.

Main success scenario

1. The Administrator requests to list the groups on the system.
2. ACME displays the complete list of the existing Group Name* in Groups*.



3. The Administrator requests to view the Access Level* for the Functional Areas* for some groups.

4. ACME displays the access levels for the functional areas in read/write mode.

ACME - Show access level

You are logged as Administrator

Group Group001's details

Functional Areas	Access Levels
Functional Area 1	Hidden
Functional Area 2	View
Functional Area 3	Hidden
...	

Change Access Level

Change Access Level

Change Access Level

Back

Extensions

3a.1. Nothing

Use Case: External User List Groups

This use case describes the process of listing groups in the ACME system by an External User.

Level userGoal

Priority 3

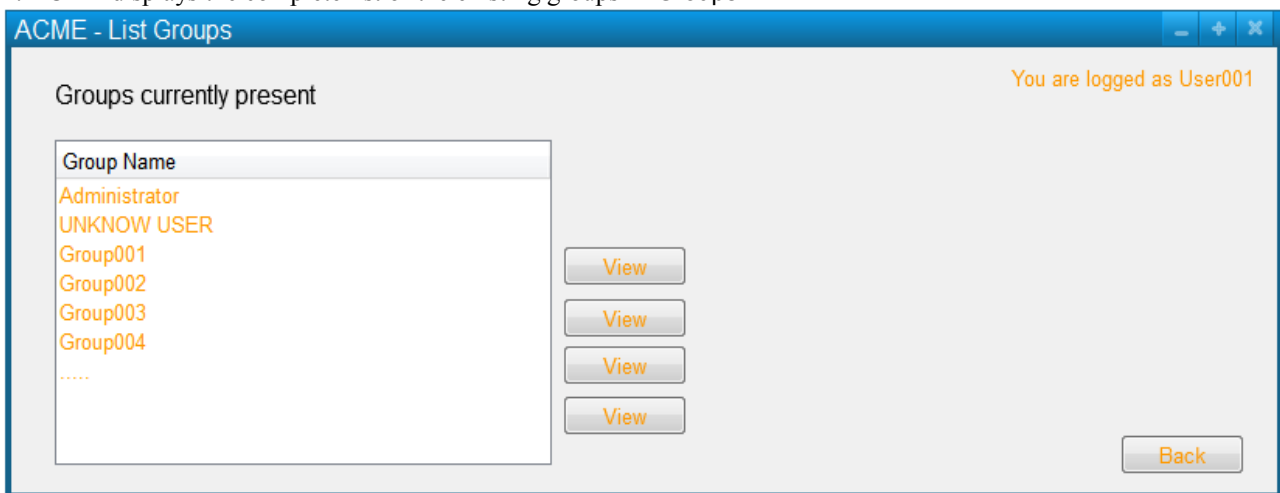
Frequency periodically

Primary Actor External User

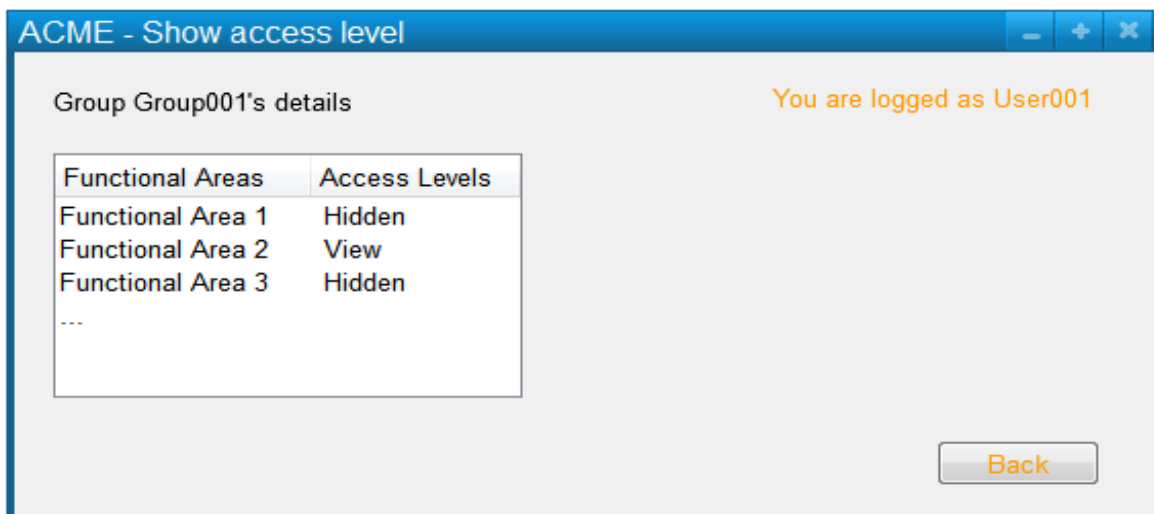
Pre-Conditions The External User belongs to Logged Users*.

Main success scenario

1. The External User requests to list the groups on the system.
2. ACME displays the complete list of the existing groups in Groups*.



3. The External User requests to view the Access Level* for the Functional Areas* for some groups.
4. ACME displays these access levels for the functional areas in read mode.



Extensions

3a.1 Nothing

Use Case: Administrator Edit Group

Goal This use case allows the administrator to edit a group.

Notes Users currently logged in are not affected.

Level userGoal

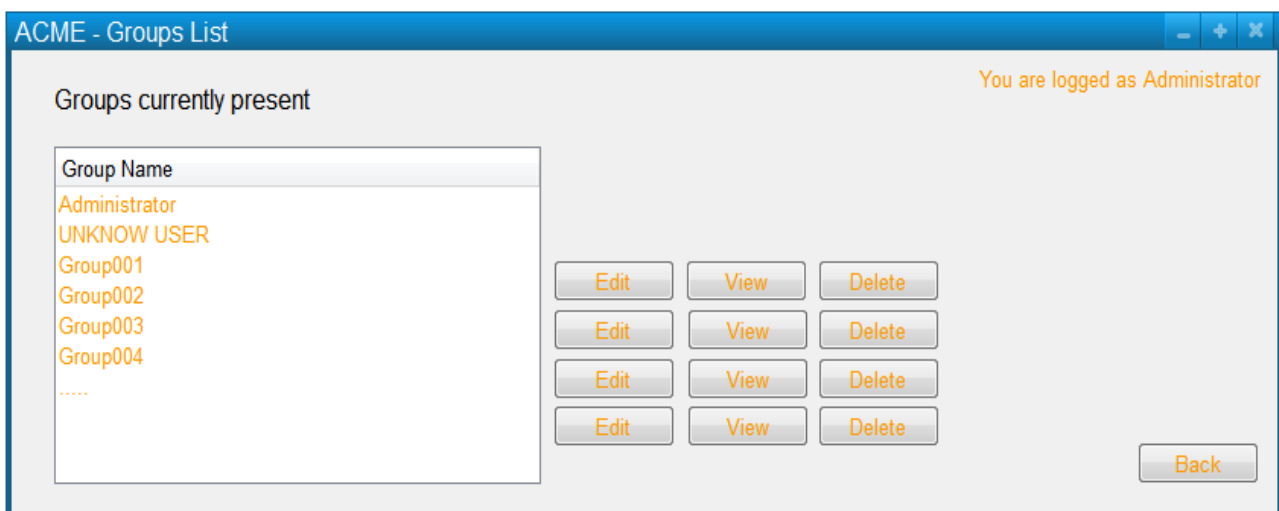
Priority 4

Frequency periodically

Primary Actor Administrator

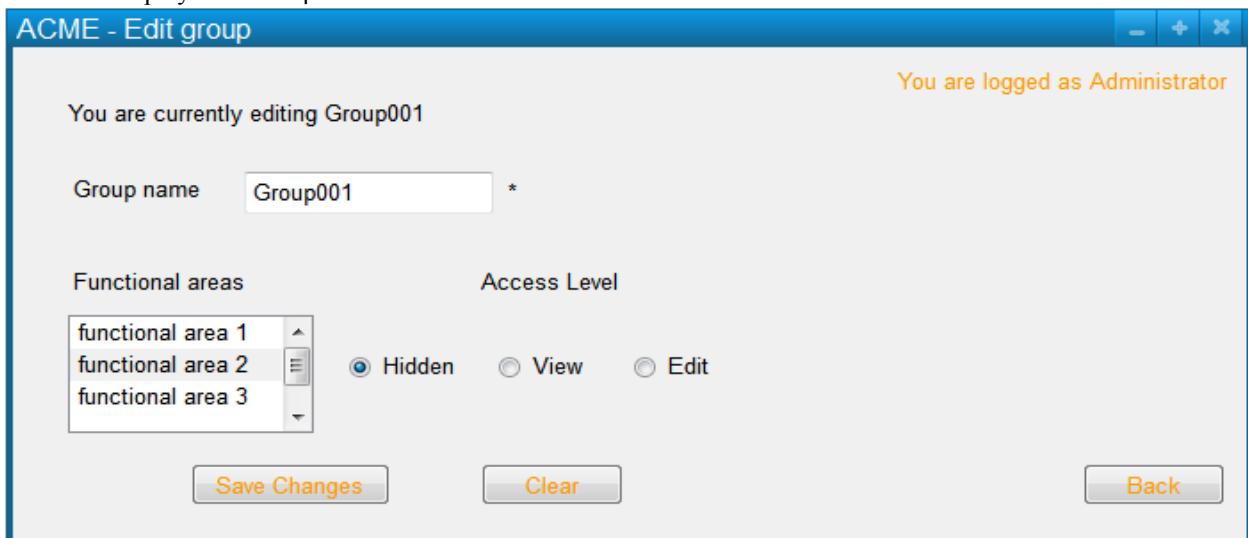
Pre-Conditions Administrator_Logged* is true, and a list of groups is displayed.

Main success scenario



1. The Administrator requests to edit a group.

2. ACME displays the Group Info* in read/write mode



3. The Administrator enters Group Info* of the selected group and confirms.

ACME - Edit group

You are currently editing Group001

You are logged as Administrator

Group name *

Functional areas

- functional area 1
- functional area 2
- functional area 3

Access Level

Hidden View Edit

4. If the Group Name* is valid and not in use in Groups*, then ACME confirms the changes and updates the group information in Groups*

ACME - Groups List

Groups currently present

- Administrator
- UNKNOW USER
- Group1
- Group002
- Group003
- Group004
-

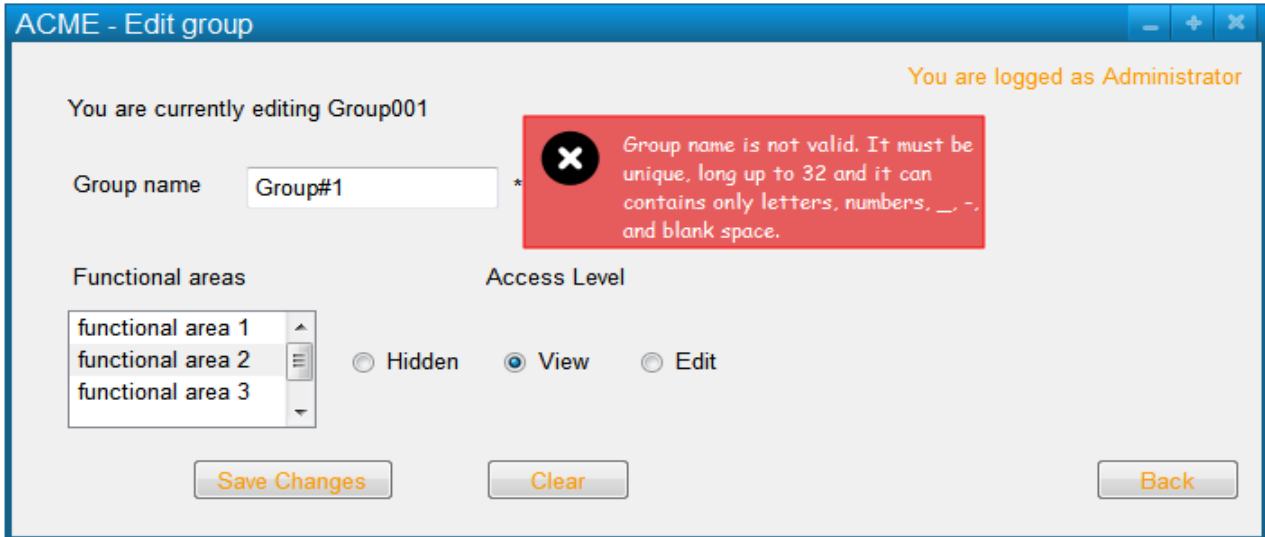
ACME - Edit Group

The selected group has been updated!

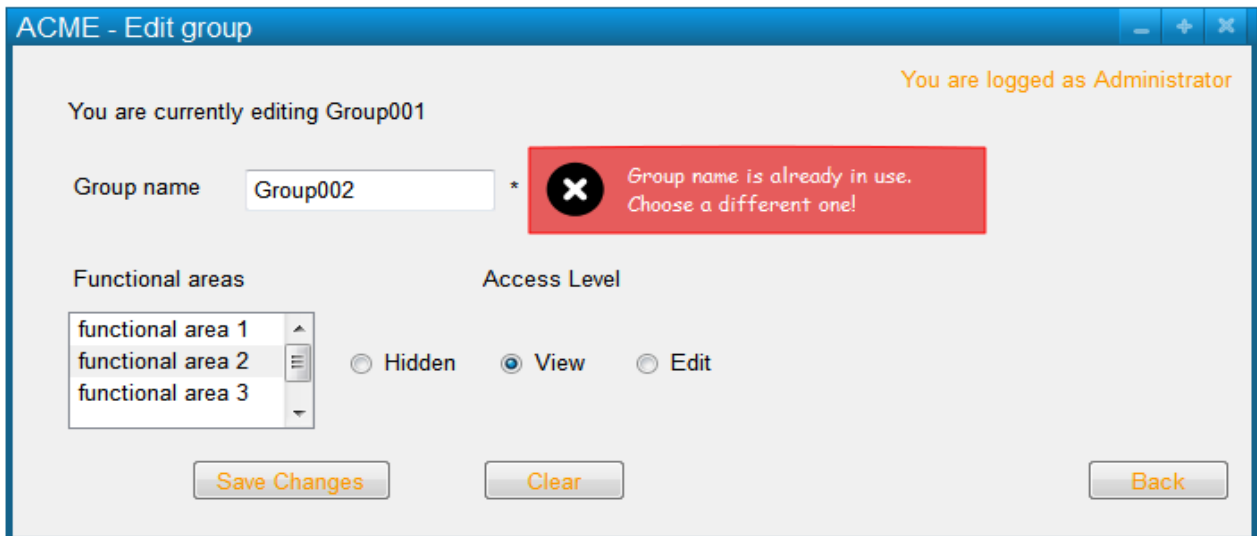
You are logged as Administrator

Extensions

4a.1. If the Group Name* is invalid, then ACME informs the Administrator and shows the proper group naming convention. The use case fails.



4b.1 If the Group Name* is already in use in Groups*, then ACME informs the Administrator to choose a different one. The use case fails.



Post-Conditions All users that are part of an edited group remain in that group, even if the name has been changed

Use Case: Administrator Delete Group

Goal This use case allows the administrator to delete a group.

Level userGoal

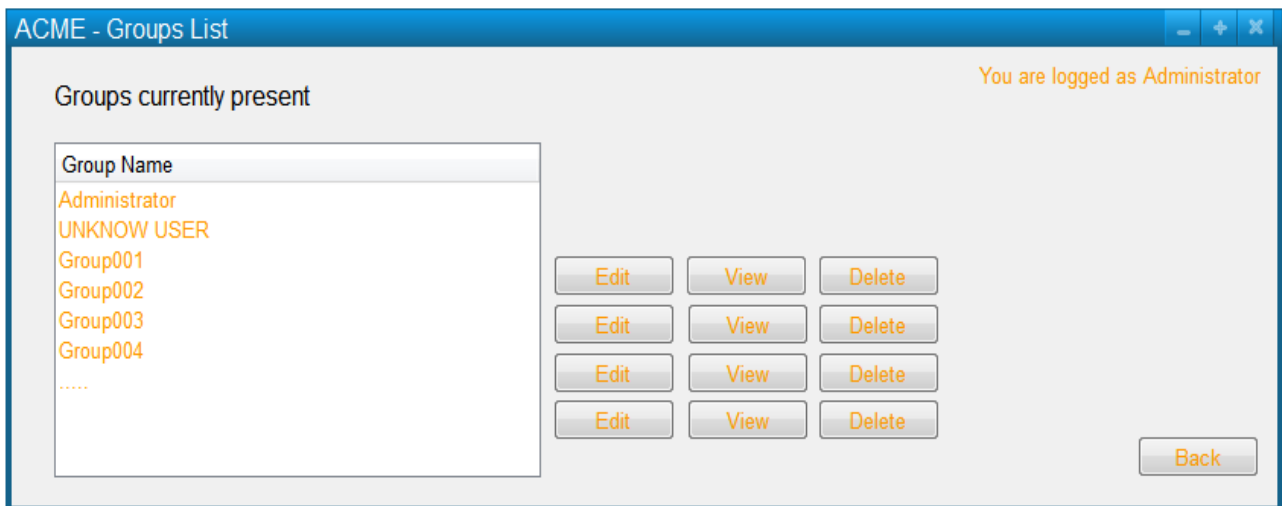
Priority 4

Frequency exceptionally

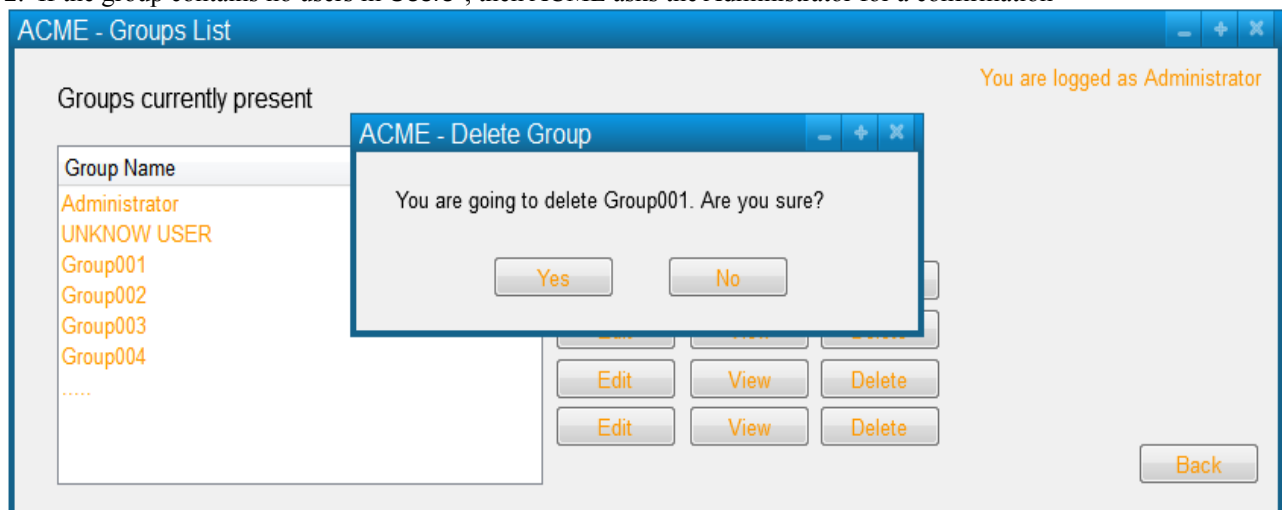
Primary Actor Administrator

Pre-Conditions Administrator_Logged* is true, and a list of groups is displayed.

Main success scenario

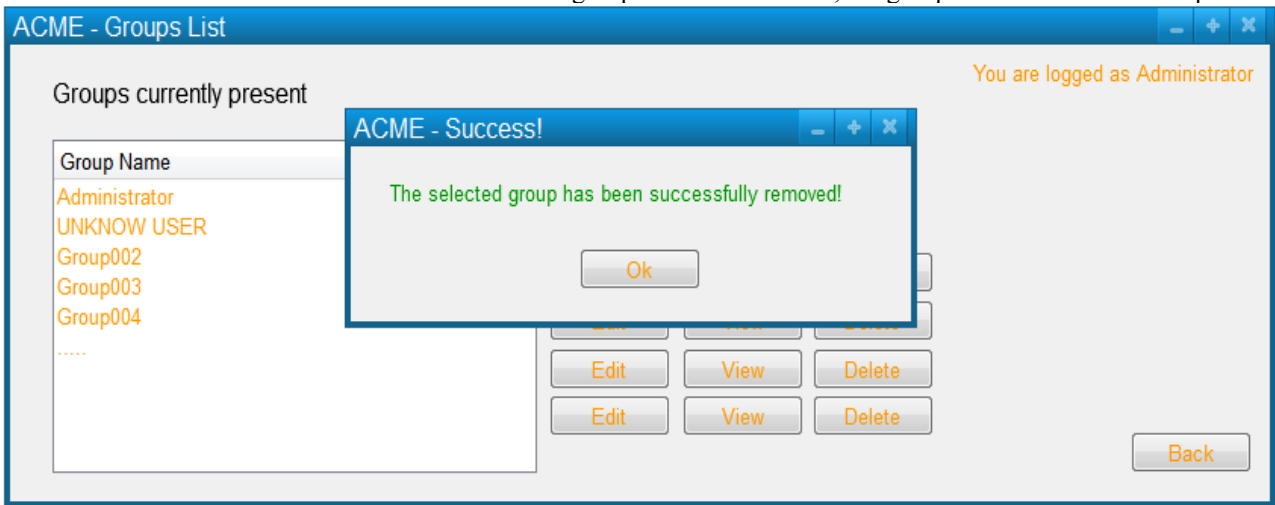


1. The Administrator requests to delete a group from the system.
2. If the group contains no users in Users*, then ACME asks the Administrator for a confirmation



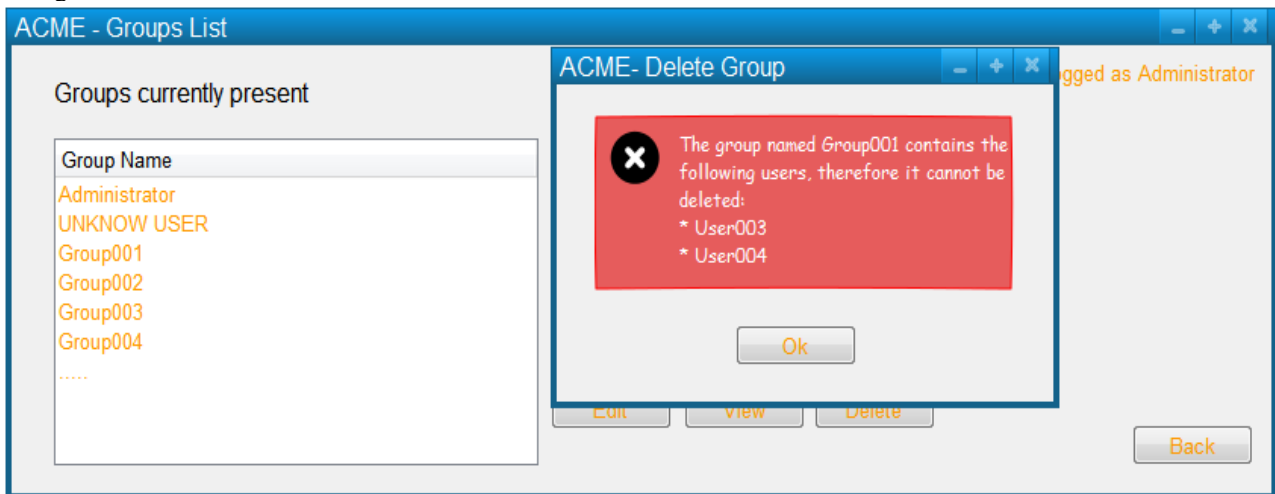
3. The Administrator confirms the intention to delete the selected group.

4. ACME informs the Administrator that the selected group has been removed; the group is removed from Groups*.



Extensions

2a.1. If the group contains any users in Users*, then ACME informs the Administrator that the group cannot be deleted, listing the associated usernames. The use case fails.



3a.1. The Administrator cancels the operation. The use case fails.

Post-Conditions Selected group is removed from Group* if it does not contains any user in Users*.

Use Case: Administrator Enable/Disable Security

Goal This use case allows the administrator to enable/disable the security.

Notes Users currently logged in are not affected

Level userGoal

Priority 1

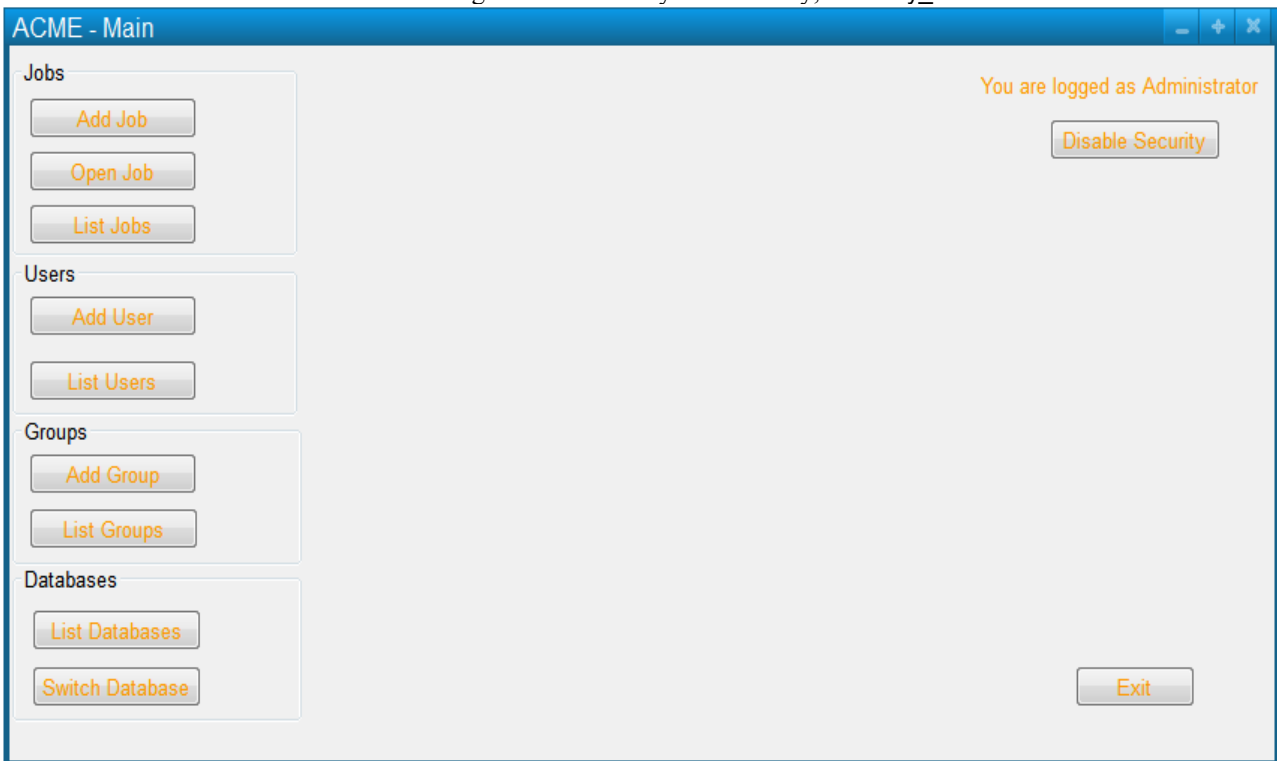
Frequency exceptionally

Primary Actor Administrator

Pre-Conditions Administrator_Logged* is true

Main success scenario

1. If Security_Enabled* is false, then the Administrator requests to enable system security.
2. ACME shows the functionalities according to the enabled system security; Security_Enabled* becomes true.



Extensions

- 1a.1. If Security_Enabled* is true, then the Administrator requests to disable system security.
- 1a.2. ACME shows the functionalities according to the disabled system security; Security_Enabled* becomes false

Use Case: External User Become Administrator

Goal This use case allows any user to become the Administrator.

Notes

The motivation for this use case are:

- It is preferred that the Login screen is not displayed when security is disabled
- A user cannot login as the Administrator when security is disabled

Level userGoal

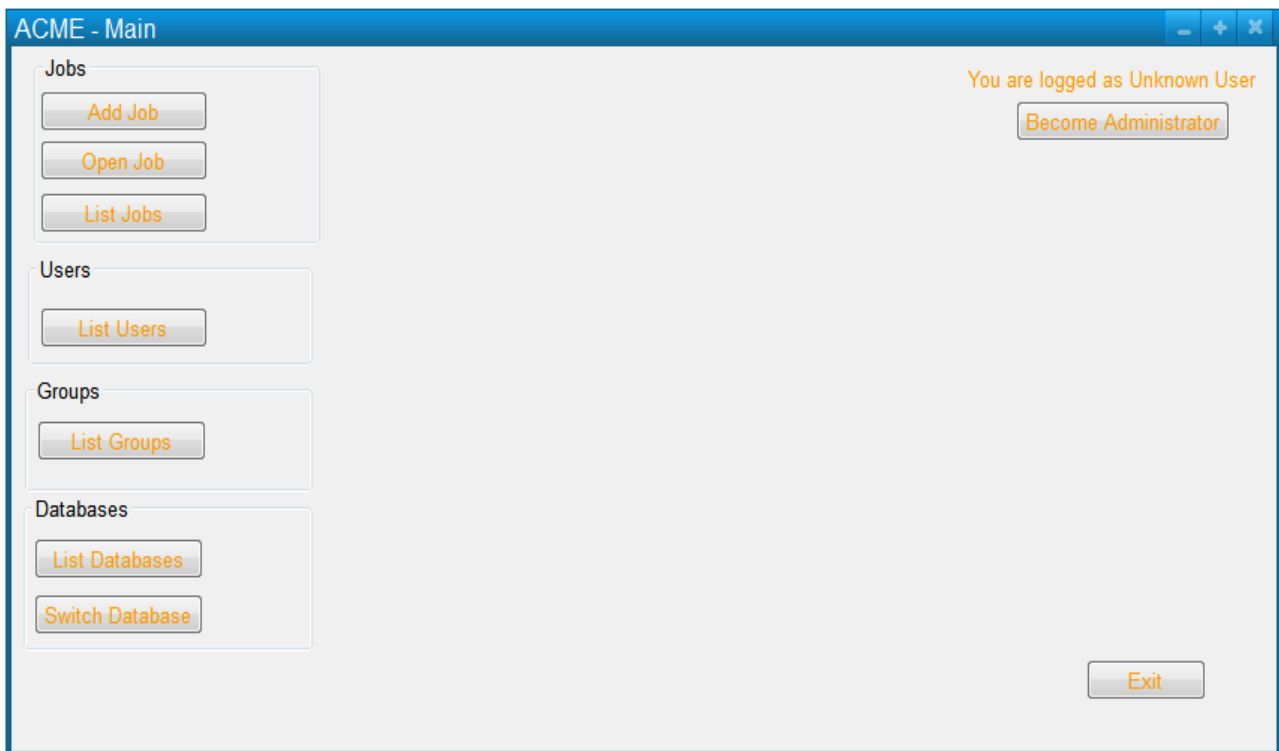
Priority 1

Frequency exceptionally

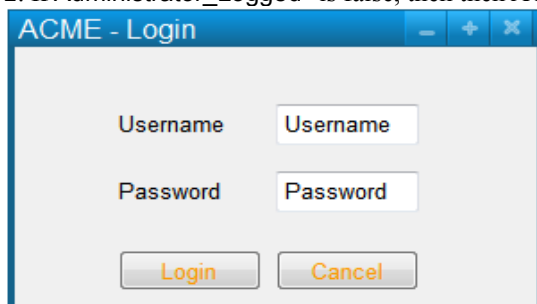
Primary Actor External User

Pre-Conditions The External User belongs to Logged Users*, and Security_Enabled* is false.

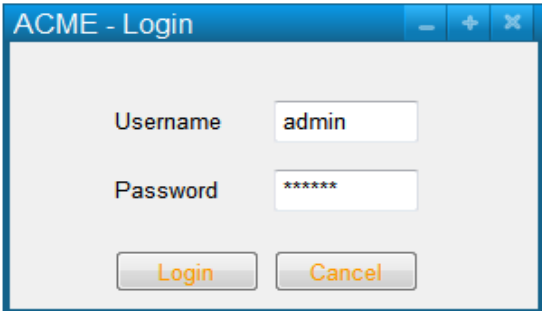
Main success scenario



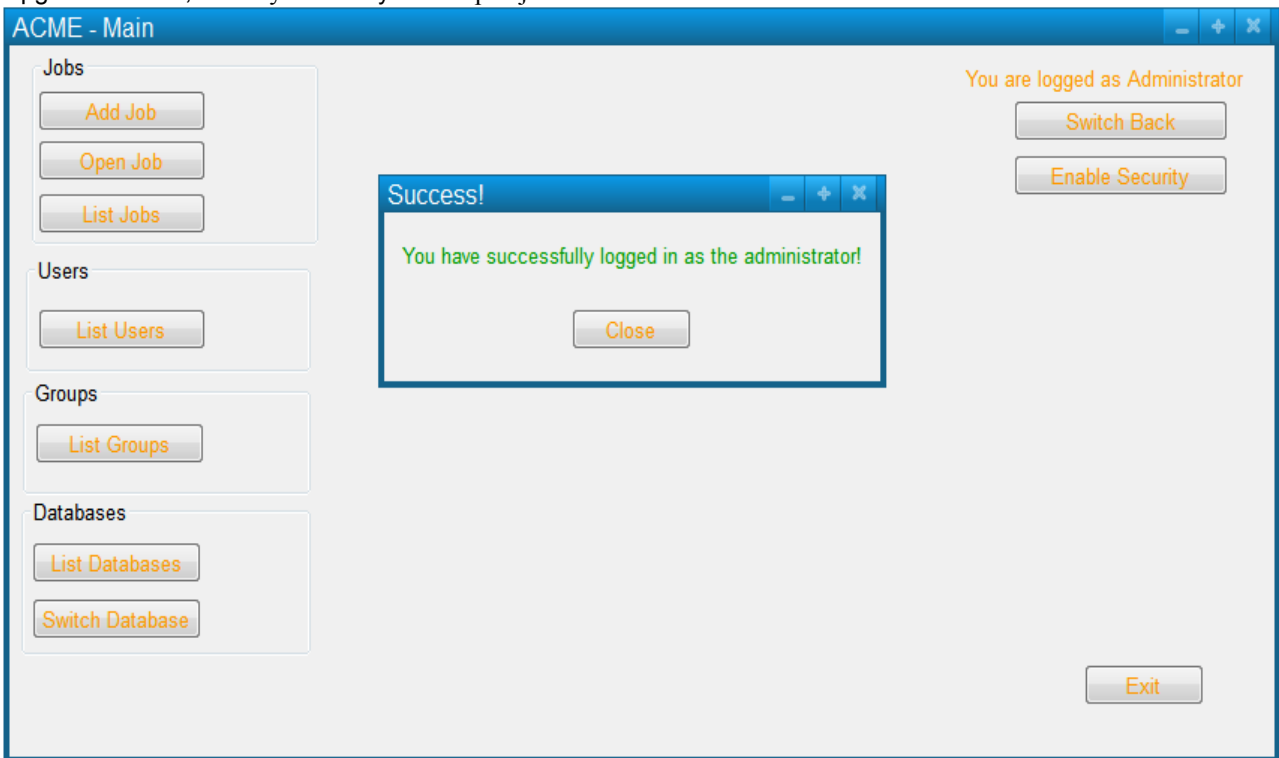
1. The External User requests to become the Administrator.
2. If Administrator_Logged* is false, then then ACME requests to log in



3. The External User enters Username* and Password*.



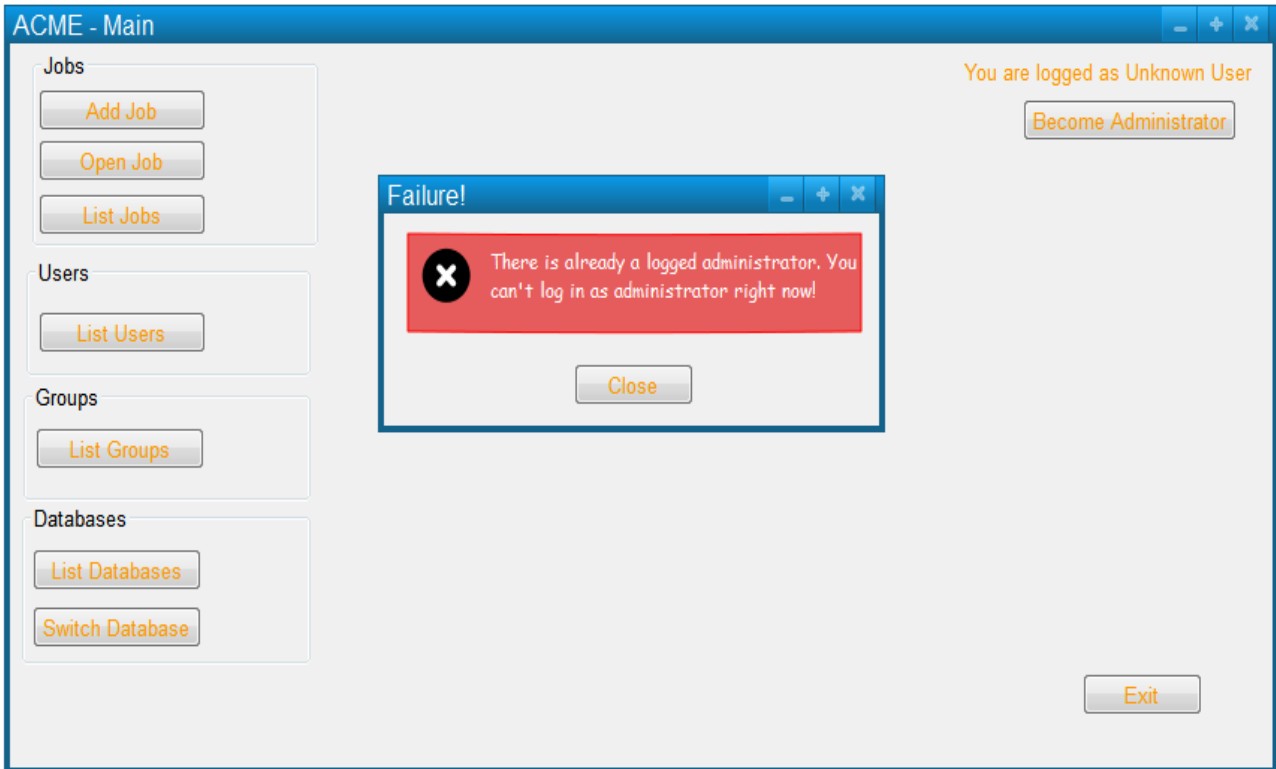
4. If the Username* and Password* are valid and the External User does not have a current Dirty Job* open in Jobs*, then ACME confirms the change of role; Administrator_Logged* becomes true, the External User name is recorded as Upgraded User*, and any non-Dirty Job* open jobs of the External User in Jobs* are closed.



Post-Conditions The External User is now an administrator and he can performs his administrative functions.

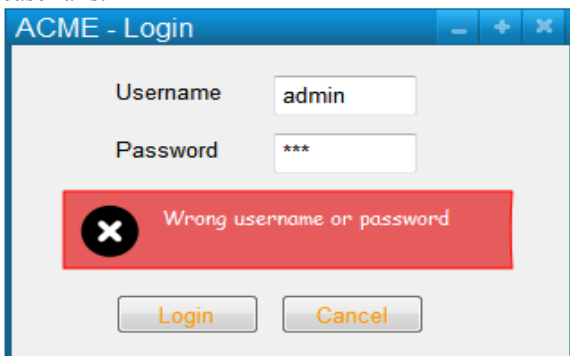
Extensions

2a.1. If Administrator_Logged* is true, then ACME informs the External User that it cannot log as Administrator. The use case fails.

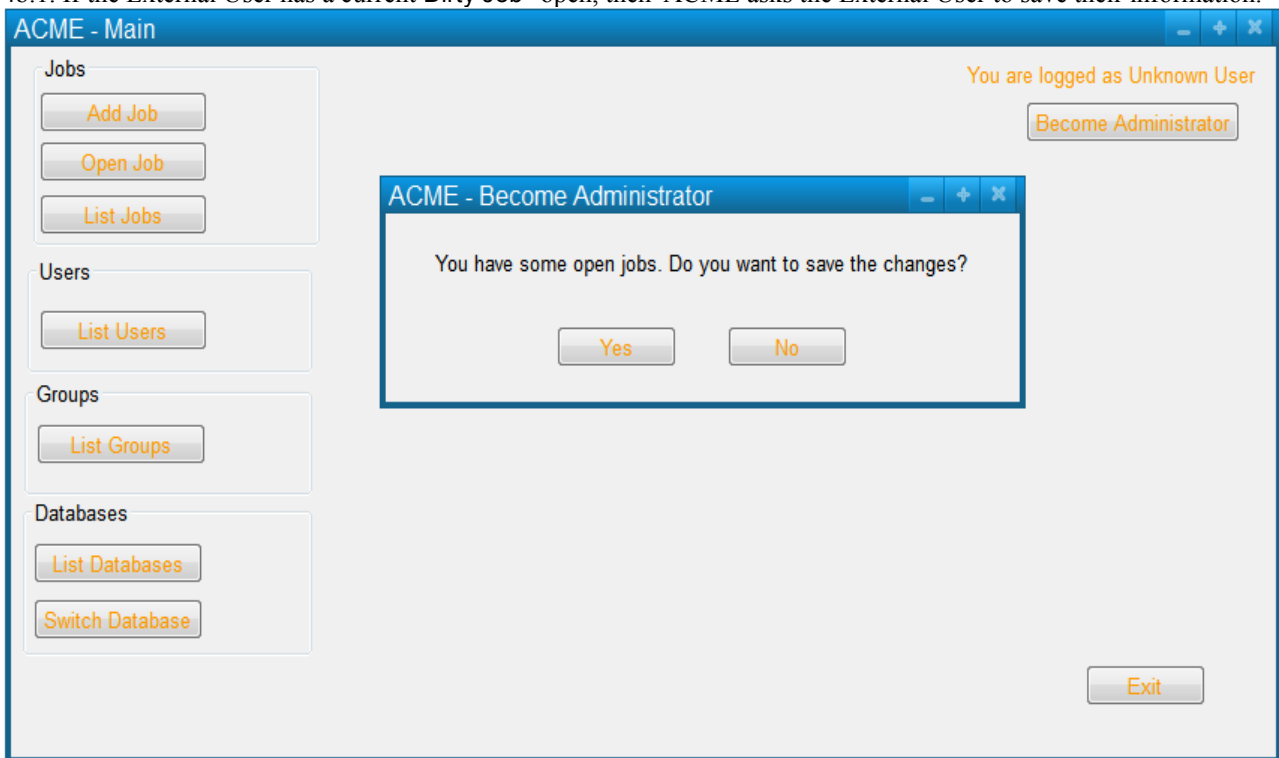


3a.1. The External User aborts the operation. The use case fails.

4a.1. If either the Username* or the Password* are invalid, then ACME informs that the given data is wrong. The use case fails.



4b.1. If the External User has a current Dirty Job* open, then ACME asks the External User to save their information.



4b.2. The External User confirms the intention to save the information.

4b.3. ACME saves the information and close the jobs. The use case continues to step 4.

4b.2.a.1. The External User refuses to save the information.

4b.2.a.2. ACME discards his changes, and closes the jobs. The use case continues to step 4.

4b.2.b.1. The External User aborts the operation. The use case fails.

Use case: Administrator Switch Back

Goal This use case allows an administrator to switch back to her/his previous role

Level userGoal

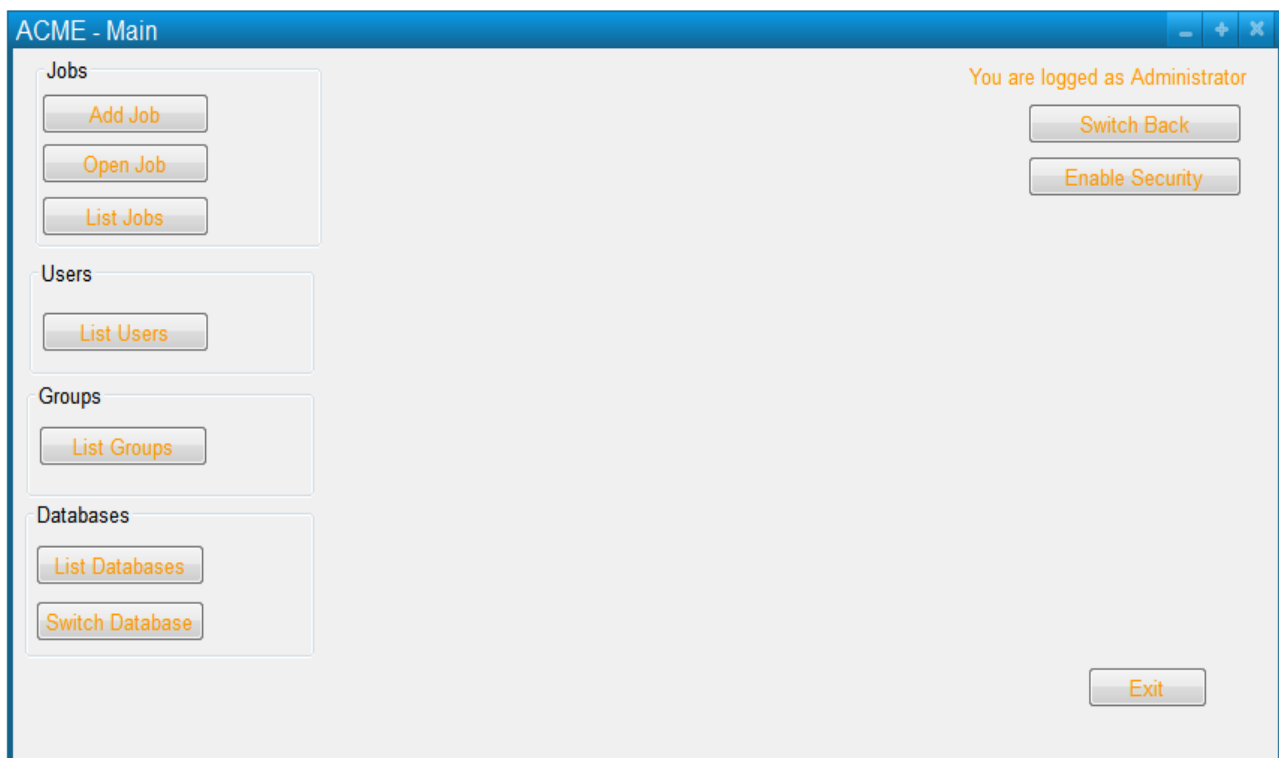
Priority 1

Frequency exceptionally

Primary Actor Administrator

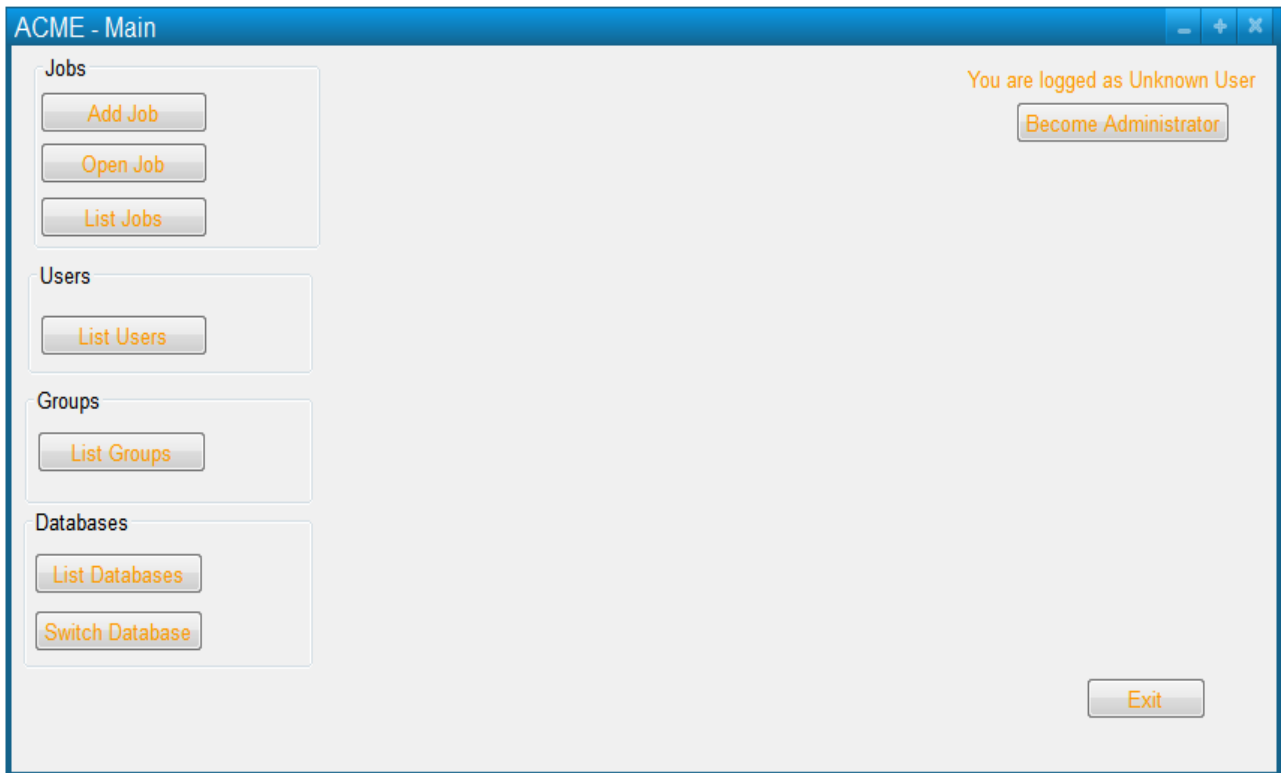
Preconditions Administrator_Logged* is true, and Security_Enabled* is false

Main Success Scenario



1. The Administrator requests to switch back to his previous identity.

2. ACME confirms the change; Upgraded User* is added to Logged Users*, and Upgraded User* becomes undefined.



Use Case: Create Job

Goal This use case allows an User to create a new job.

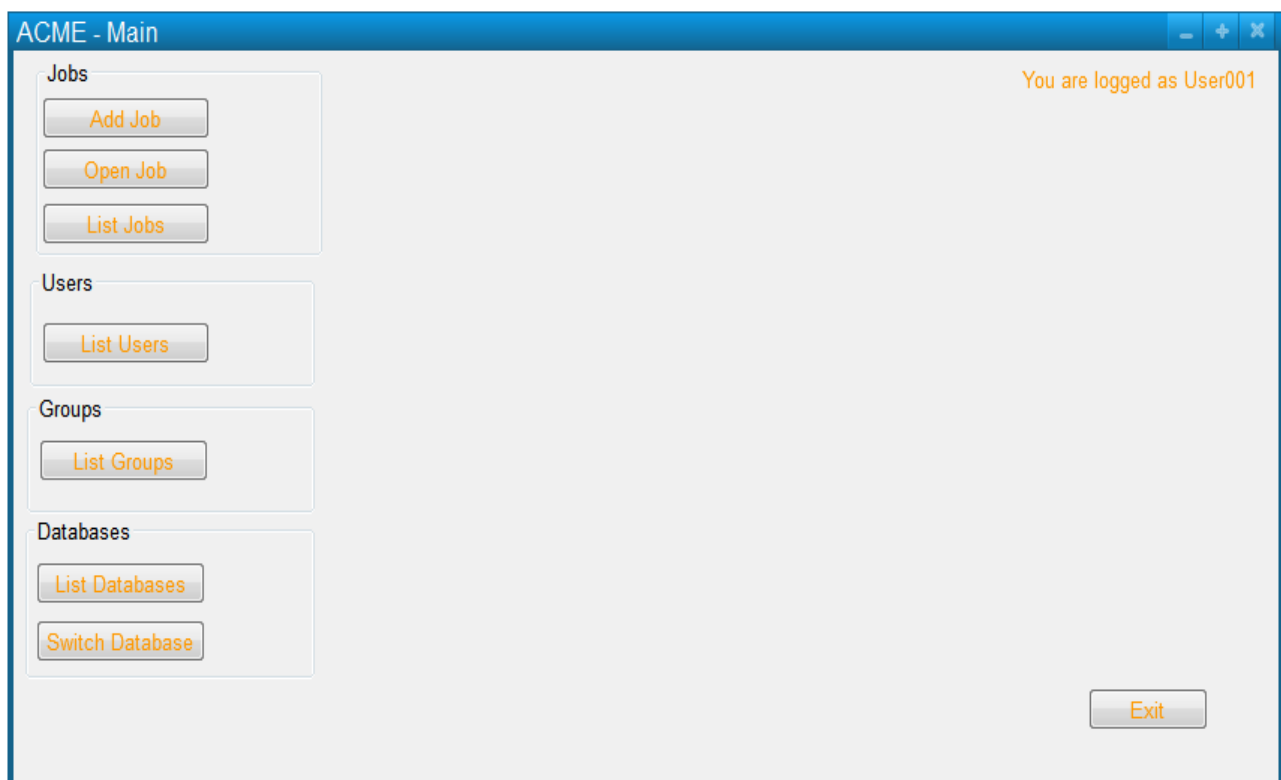
Level userGoal

Priority 1

Frequency continuously

Primary Actor User

Main success scenario



1. The User requests to add a new job.
2. ACME prompts the User for the Job_Info*.

ACME - Create Job

You are logged as User001

Insert the following job's info

Short Description

Long Description

Title

Issue

Type

Module

Add Job Clear Back

Detailed description: This screenshot shows the 'ACME - Create Job' window. The title bar is blue with standard window controls. The main area has a light gray background. At the top right, it says 'You are logged as User001'. Below the title, it says 'Insert the following job's info'. There are two columns of input fields. The left column has a 'Short Description' text box and a larger 'Long Description' text area. The right column has a 'Title' text box, an 'Issue' text box, a 'Type' dropdown menu, and a 'Module' dropdown menu. At the bottom, there are three buttons: 'Add Job' (highlighted in orange), 'Clear', and 'Back'.

3. The User enters the Job_Info* and confirms.

ACME - Create Job

You are logged as User001

Insert the following job's info

Job 001

Job 001 Title

none

this is a job

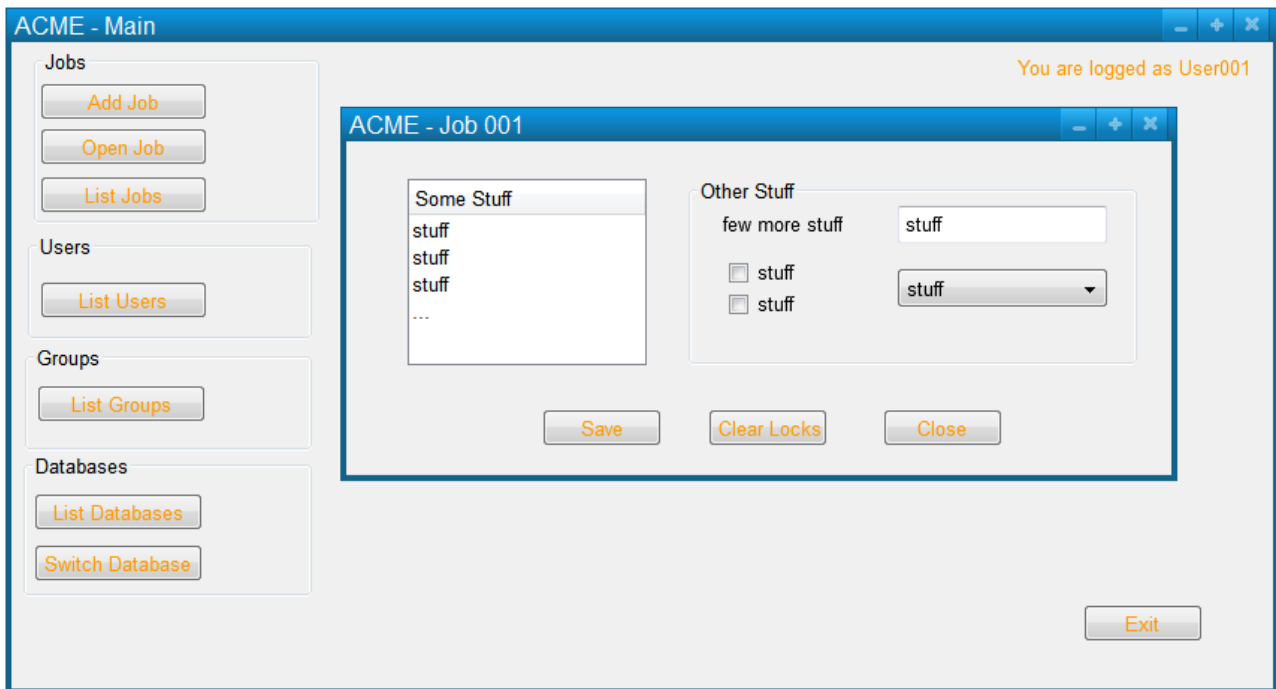
planned

module 1

Add Job Clear Back

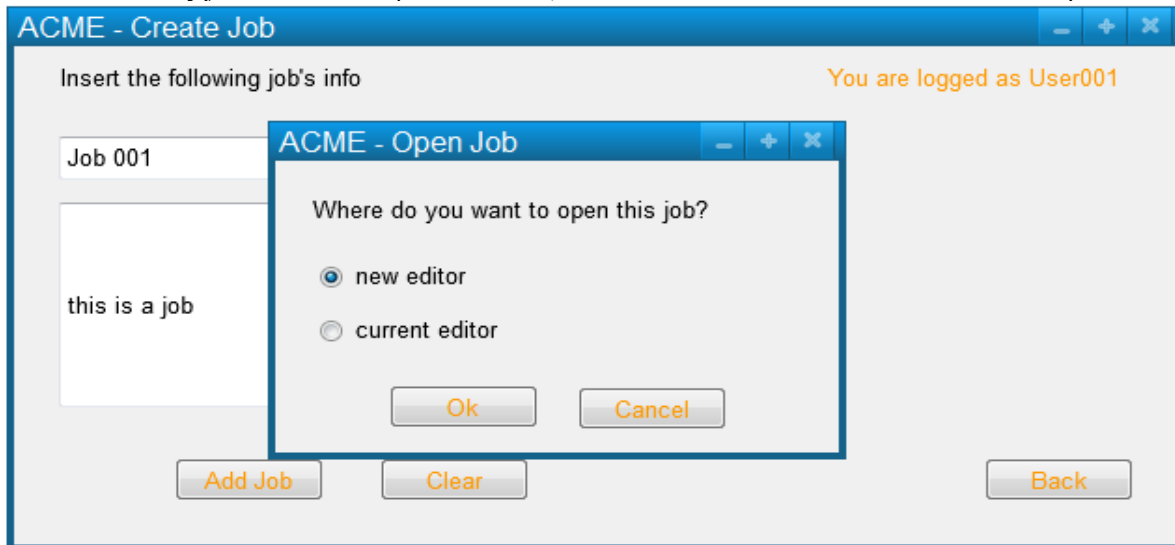
Detailed description: This screenshot shows the same 'ACME - Create Job' window, but now the input fields are filled with data. The 'Short Description' text box contains 'Job 001'. The 'Long Description' text area contains 'this is a job'. The 'Title' text box contains 'Job 001 Title'. The 'Issue' text box contains 'none'. The 'Type' dropdown menu is set to 'planned'. The 'Module' dropdown menu is set to 'module 1'. The 'Add Job' button is still highlighted in orange.

4. If there is no job in Open In Editor*, then ACME opens the new job in the editor; an entry for this job characterized by the Job_Info* is added in Jobs*



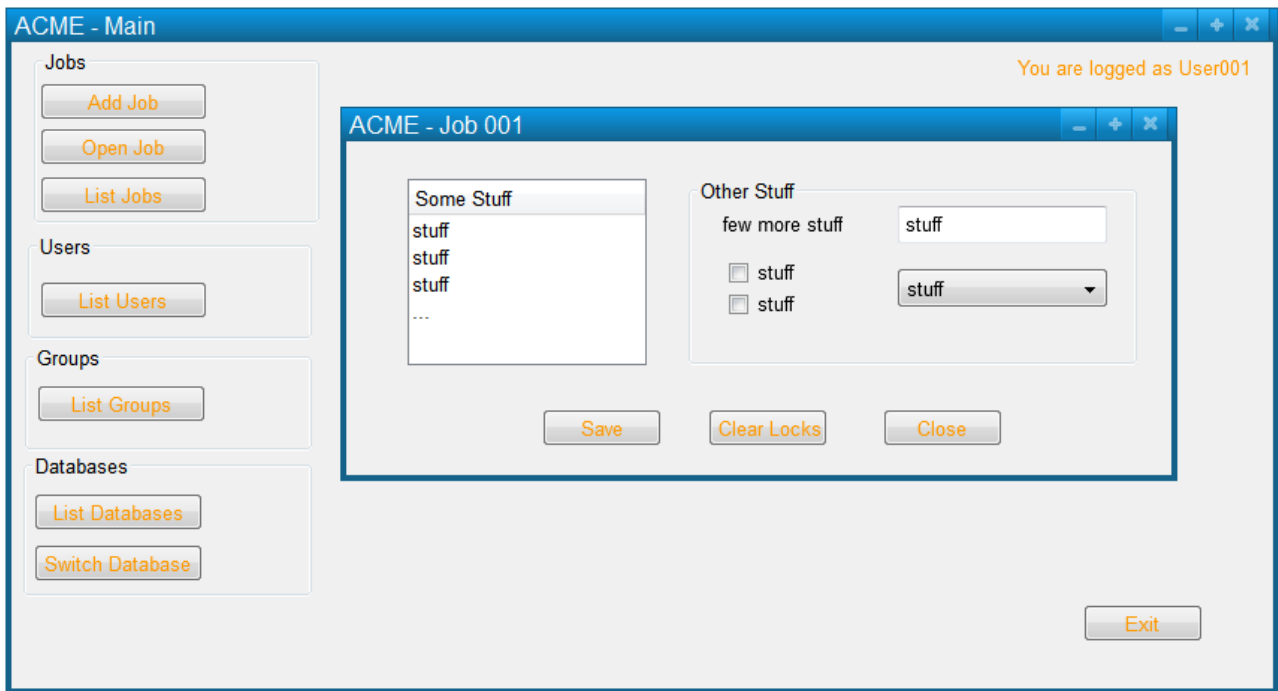
Extensions

4a.1. If a non dirty job in Jobs* is Open In Editor*, then ACME asks the User to choose where to open the new job.



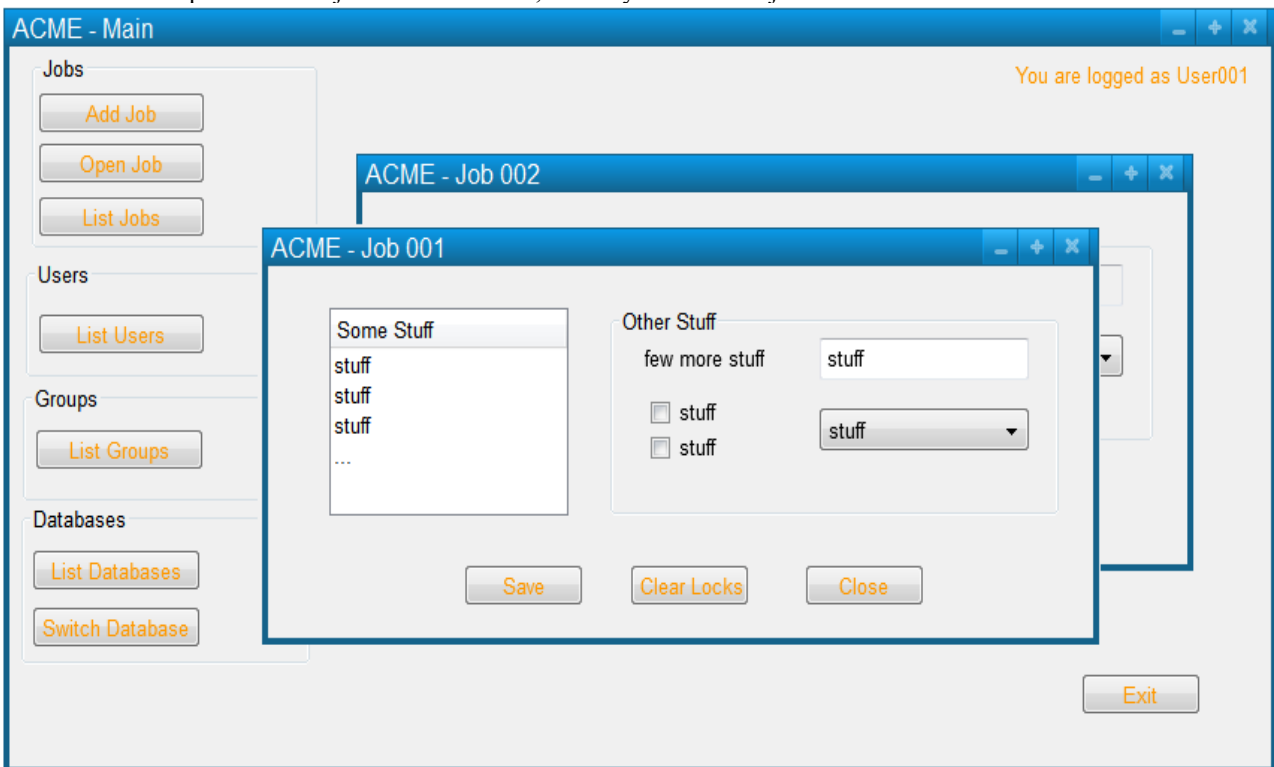
4a.2. The User answers to open the job in the current editor.

4a.3. ACME opens the new job in the current editor and closes the old non dirty-Jobs*; an entry for the new job is added in Jobs*.



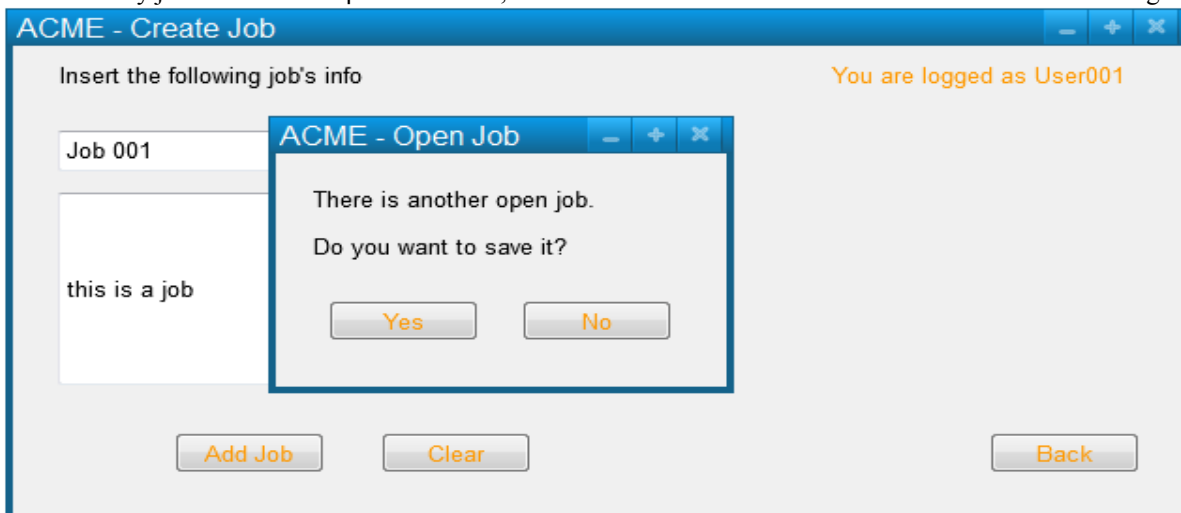
4a.2.a.1. The User answers to open the job in a new editor.

4a.2.a.2. ACME opens the new job in a new editor; an entry for the new job is added in Jobs*.



4a.2.b.1. The User aborts the operation. The use case fails.

4b.1. If a dirty job in Jobs* is Open In Editor*, then ACME asks the User if he wants to save unsaved changes.



4b.2. The User answers to save the changes.

4b.3 ACME asks the User to choose where to open the new job; the changes of the old dirty job in Jobs* are saved. The use case continues at step 4a.2.

4b.2.a.1 The User refuses to save the changes.

4b.2.a.2 ACME asks the User to choose where to open the new job; the changes of the old dirty job in Jobs* are discarded. The use case continues at step 4a.2

4b.2.b.1. The User aborts the operation. The use case fails.

Use Case: Administrator Search/List Jobs

Goal This use case allows the Administrator to search for jobs that are currently on the system.

Level userGoal

Priority 1

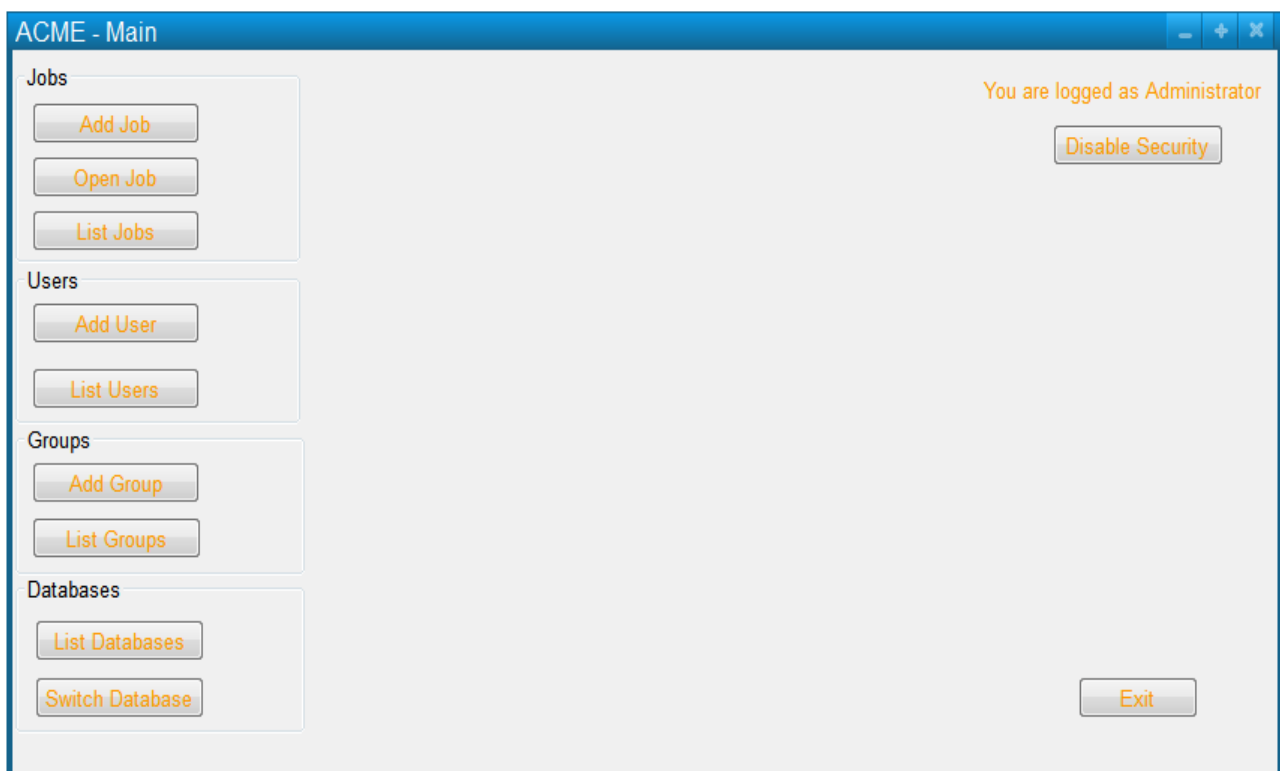
Frequency frequently

Stakeholders The company that intend to sell the software and the future users.

Primary Actor Administrator

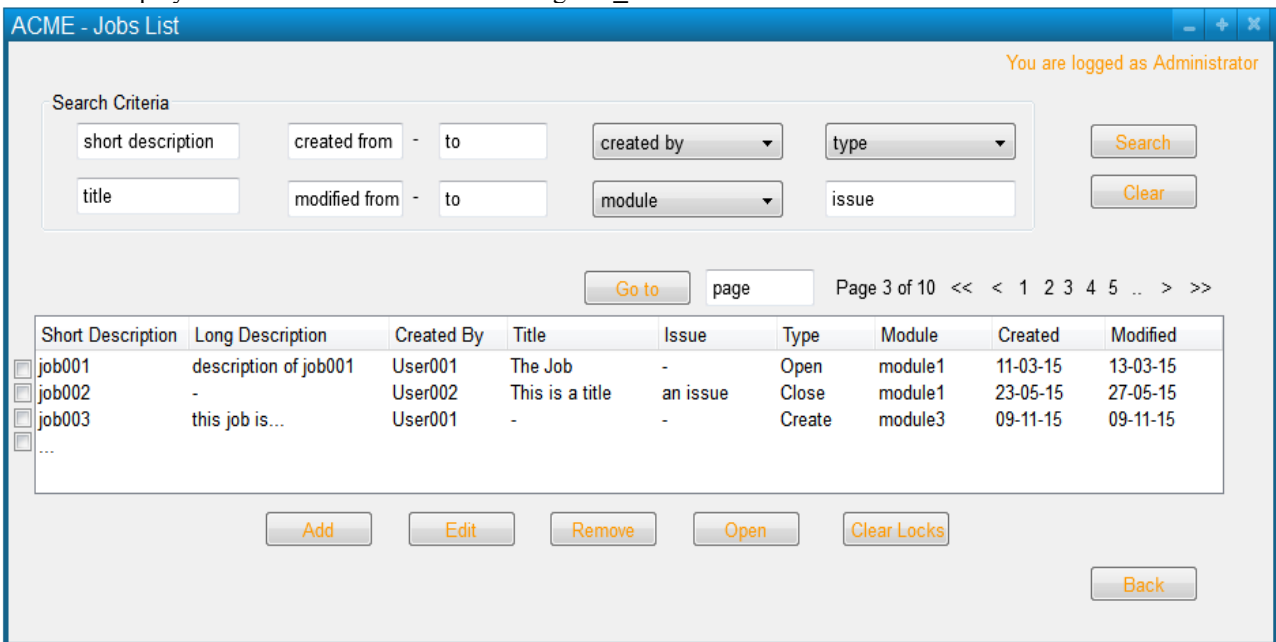
Preconditions Administrator_Logged* is true

Main success scenario

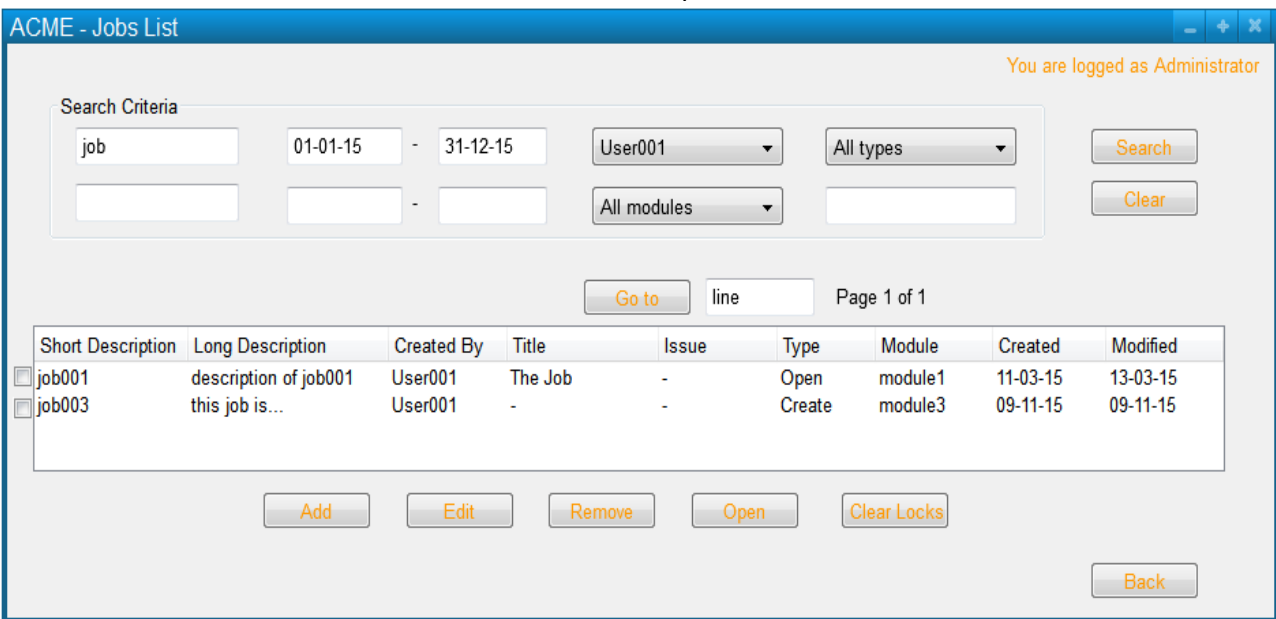


1. The Administrator requests to list the jobs.

2. ACME displays all Jobs* with their characterizing Job_Info*.



3. The Administrator enters the desired Search Criteria* and requests to search.



4. If there is at least one job satisfying the Search Criteria*, then ACME lists all jobs matching it.

Extensions:

4a.1 If there is no job satisfying the Search Criteria*, then ACME informs the Administrator that there is no job matching the inserted criteria. The use case fails.

Search Criteria

job

01-01-15

- 31-12-15

User001

All types

Search

title#123

-

All modules

Clear



No results found.

Go to

line

Page 1 of 1

Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
-------------------	------------------	------------	-------	-------	------	--------	---------	----------

Add

Edit

Remove

Open

Clear Locks

Back

Use Case: External User Search/List Jobs

Goal This use case lets the External User search for jobs that are currently on ACME.

Level userGoal

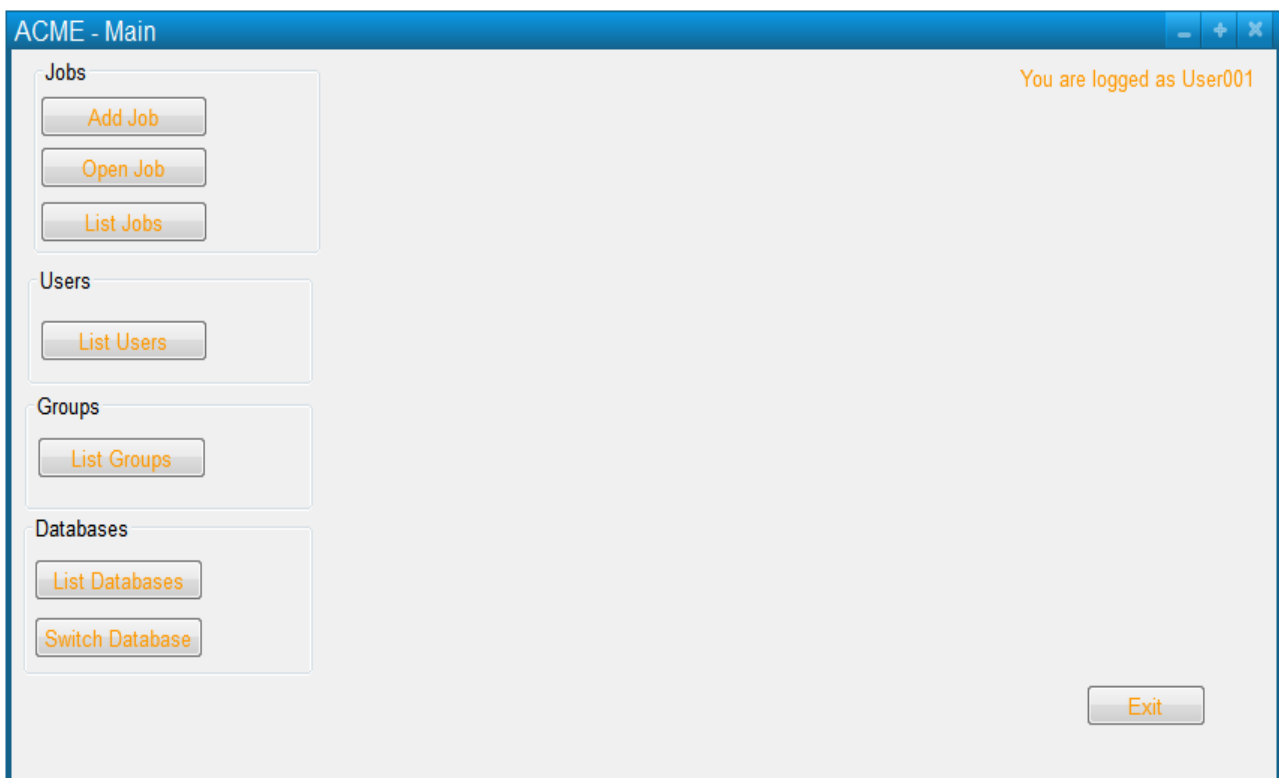
Priority 1

Frequency frequently

Primary Actor External User

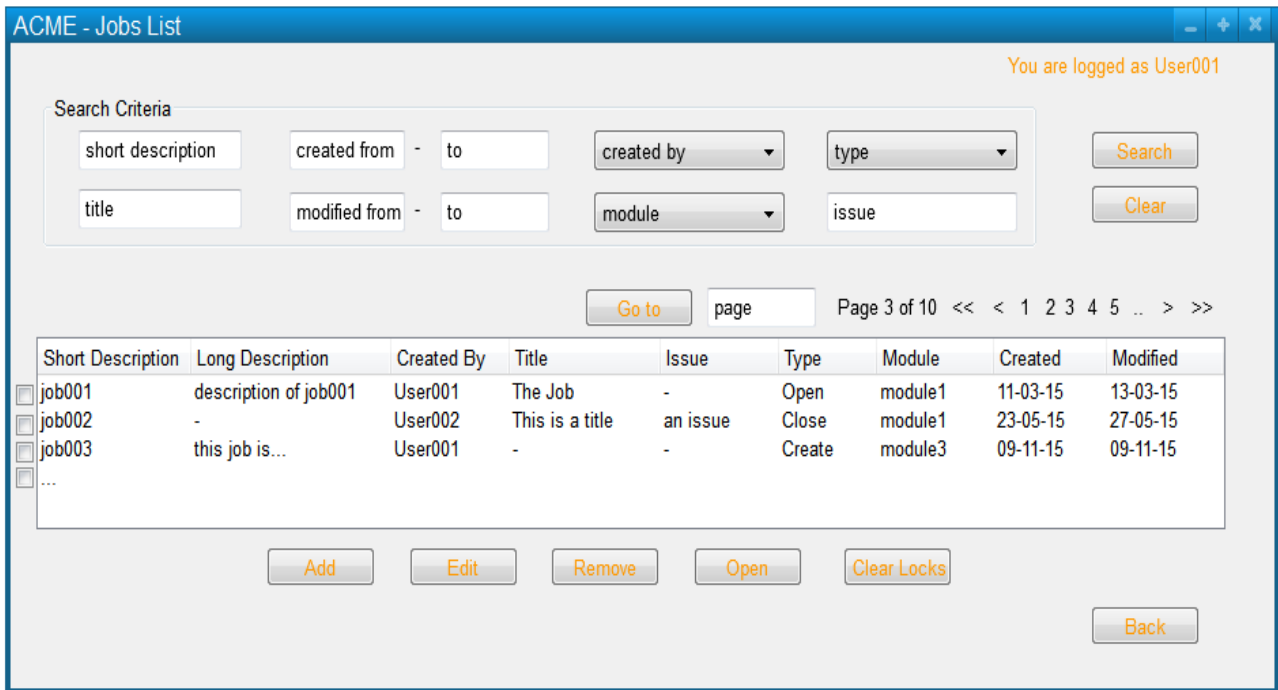
Preconditions The External User belongs to Logged Users*

Main success scenario



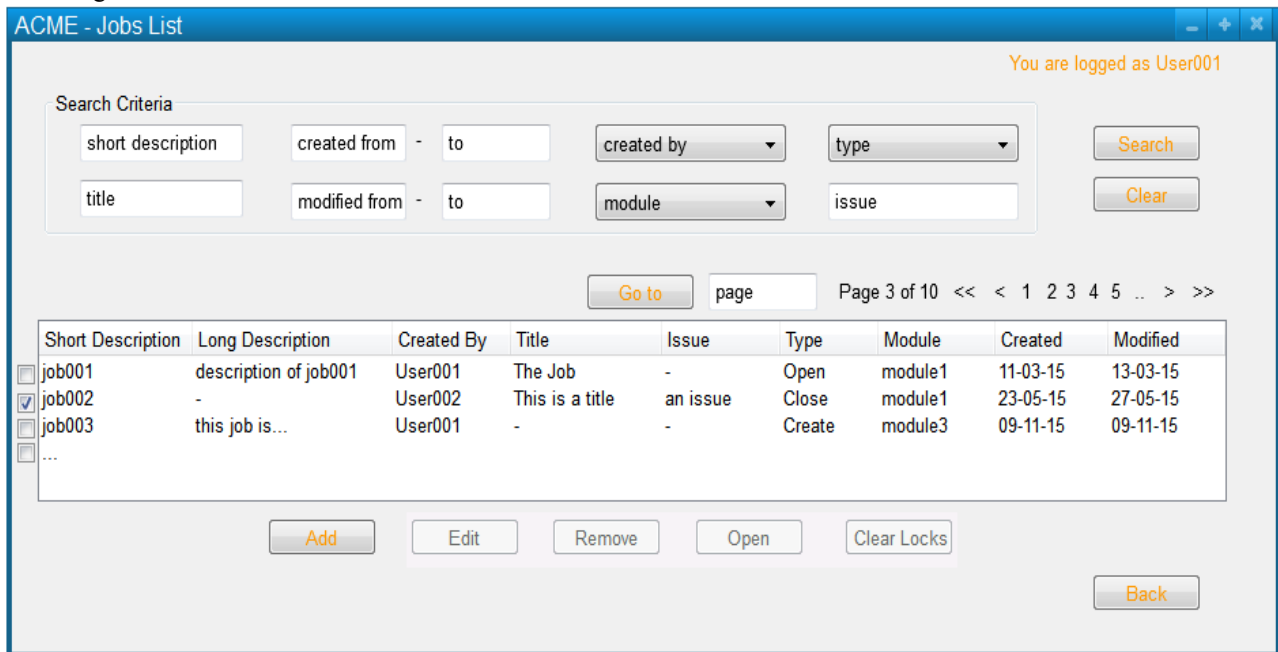
1. The External User requests to list the jobs.

2. ACME displays all jobs in Jobs* with their characterizing Job_Info*.



3. The External User enters the desired Search Criteria* and requests to search.

4. If Security_Enabled* is true, then ACME lists all jobs matching the Search Criteria* and CRUD options are shown according to Access Level*.



Extensions

4a.1 If Security_Enabled* is false, then ACME lists all jobs matching the Search Criteria* and all job CRUD options are available

ACME - Jobs List

You are logged as Unknown User

Search Criteria

job 01-01-15 - 31-12-15 User01 All types Search Clear

Go to line Page 1 of 1

Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15

Add Edit Remove Open Clear Locks Back

Use Case: Open Job

Goal This use case allows a User to open a job

Notes

- Users can have multiple ACME sessions open per PC
- Users can be logged into more than one PC
- A Job and its modules may be locked by another user
- A Job may be opened by another user, but no modules may be currently locked by that user

Level userGoal

Priority 1

Frequency continuously

Primary Actor User

Actor User

Preconditions A Job List* is displayed.

Main success scenario

ACME - Jobs List

You are logged as User001

Search Criteria

short description created from - to created by type Search

title modified from - to module issue Clear

Go to page Page 3 of 10 << < 1 2 3 4 5 .. > >>

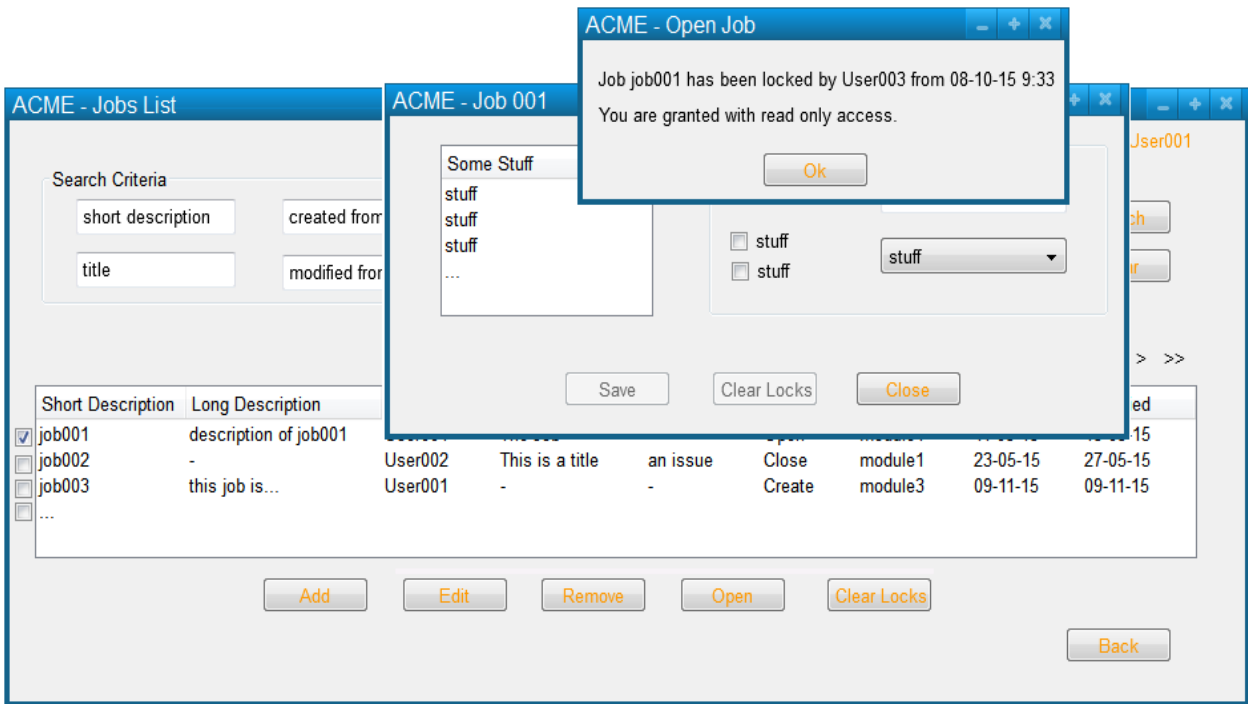
Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/> ...								

Add Edit Remove Open Clear Locks

Back

1. If no job is Open In Editor*, the User requests to open a job.

2. If the job or one of its module is locked by another user in Jobs*, then ACME warns the User providing the lock information and opens the job in the current editor in read-only mode.



Extensions

2a.1. If another user has opened the job but neither the job nor its modules are locked in Jobs*, then ACME warns the User which other users are editing this job at this time and opens the job in the current editor; the user that opened the job is recorded in Open Jobs* at this date/time, and the lock on the job in Jobs* is updated with the current user.

ACME - Jobs List

Search Criteria

short description

title

ACME - Job 001

Some Stuff

stuff
stuff
stuff
...

ACME - Open Job

The following users are currently editing this job:

- User005

Now you own the lock for this job.

Ok

Save Clear Locks Close

Search Clear

2 3 4 5 .. > >>

Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input checked="" type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/> ...								

Add Edit Remove Open Clear Locks

Back

re logged as User001

2b.1. If the job is not currently locked by another user, then ACME opens the job in the current editor; the user that opened the job is recorded in **Open Jobs*** at this date/time, and the lock on the job in **Jobs*** is updated with the current user.

The screenshot displays the ACME software interface. A modal window titled "ACME - Job 001" is open, showing two text areas: "Some Stuff" and "Other Stuff". The "Some Stuff" area contains the text "stuff", "stuff", "stuff", and "...". The "Other Stuff" area contains the text "few more stuff" and "stuff", followed by two checkboxes labeled "stuff" and a dropdown menu with "stuff" selected. Below these areas are buttons for "Save", "Clear Locks", and "Close".

The background window, titled "ACME - Jobs List", shows a search criteria section with "short description" and "title" input fields. Below this is a table with the following data:

	Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input checked="" type="checkbox"/>	job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/>	job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/>	job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/>	...								

Below the table are buttons for "Add", "Edit", "Remove", "Open", "Clear Locks", and "Back".

Use Case: Edit Job

Goal This use case allows a User to edit a job

Level userGoal

Priority 1

Frequency frequently

Primary Actor User

Preconditions A Job List* is displayed.

Main success scenario

ACME - Jobs List

You are logged as User001

Search Criteria

short description created from - to created by type Search

title modified from - to module issue Clear

Go to page Page 3 of 10 << < 1 2 3 4 5 .. > >>

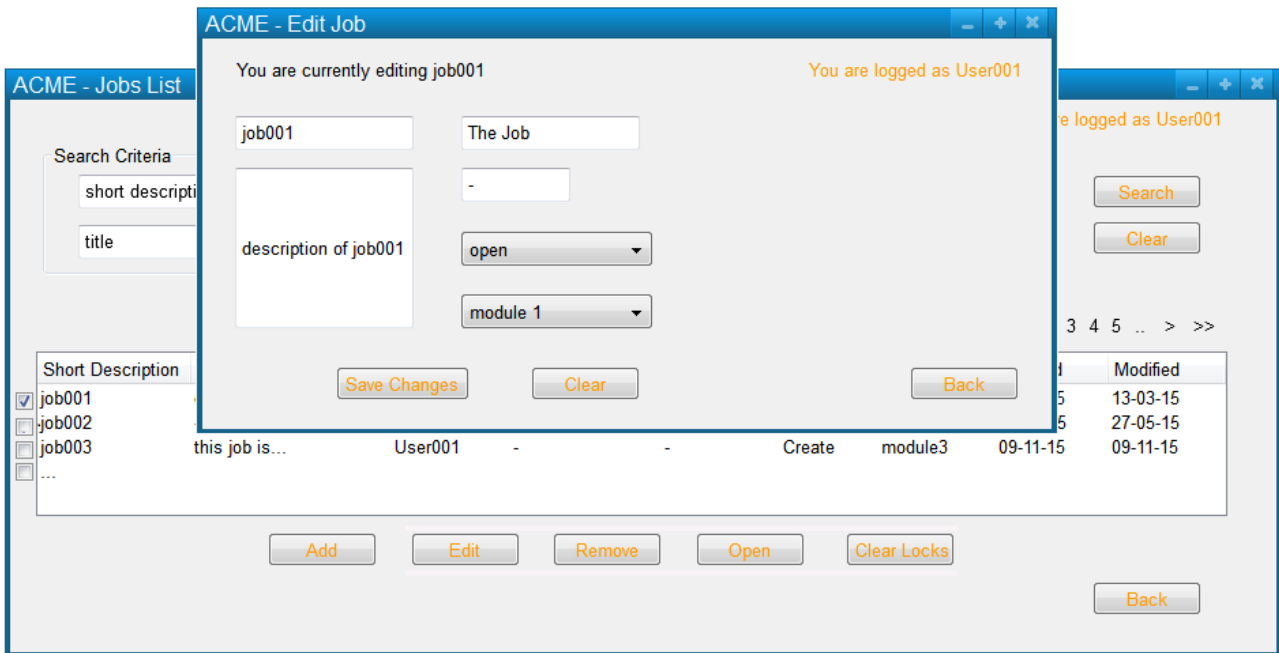
Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/> ...								

Add Edit Remove Open Clear Locks

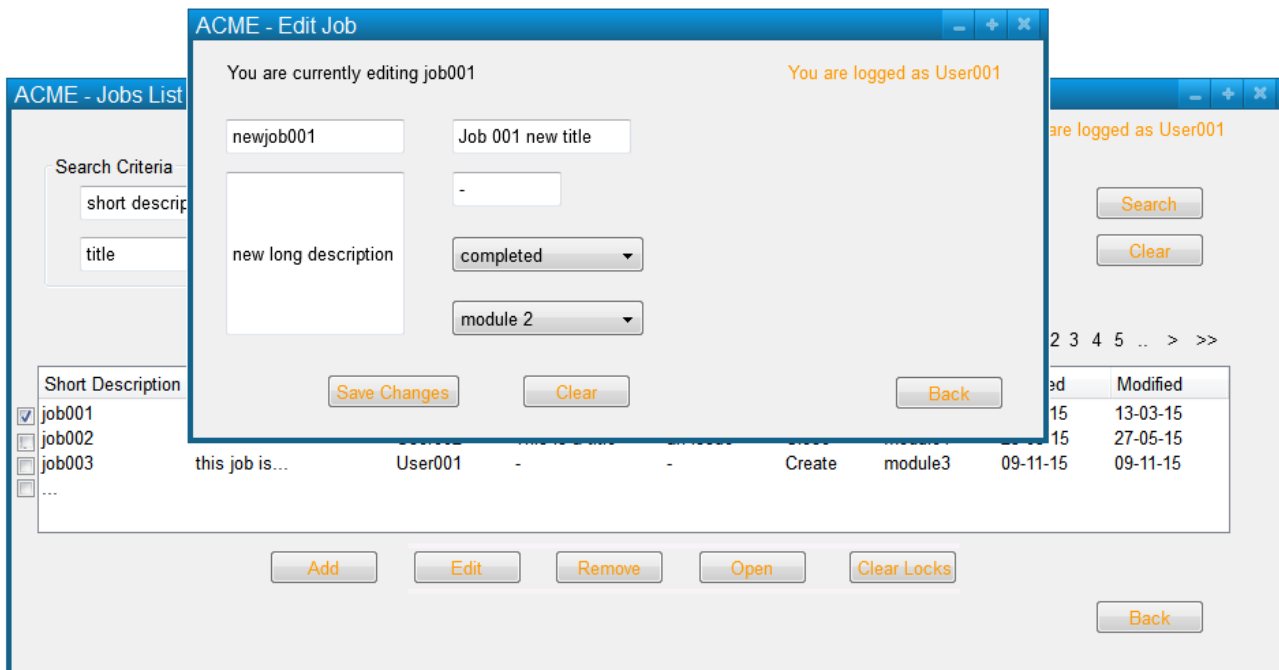
Back

1. The User requests to edit a job in accordance with his rights.

2. If the User has write permission and the job is not locked by another user in Jobs*, then ACME displays the Job Info*; a lock for the job by that user in is updated in Jobs*.



3. The User enters the Job Info* and confirms the changes.



4. ACME unlocks the job in Jobs* and prompt the User back to Job List*, where the changes have been applied.

Search Criteria

short description

created from

to

created by

type

Search

title

modified from

to

module

issue

Clear

Go to

page

Page 3 of 10

<< < 1 2 3 4 5 .. > >>

Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input type="checkbox"/> newjob001	new long description	User001	Job 001 new title	-	Completed	module2	11-03-15	15-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/> ...								

Add

Edit

Remove

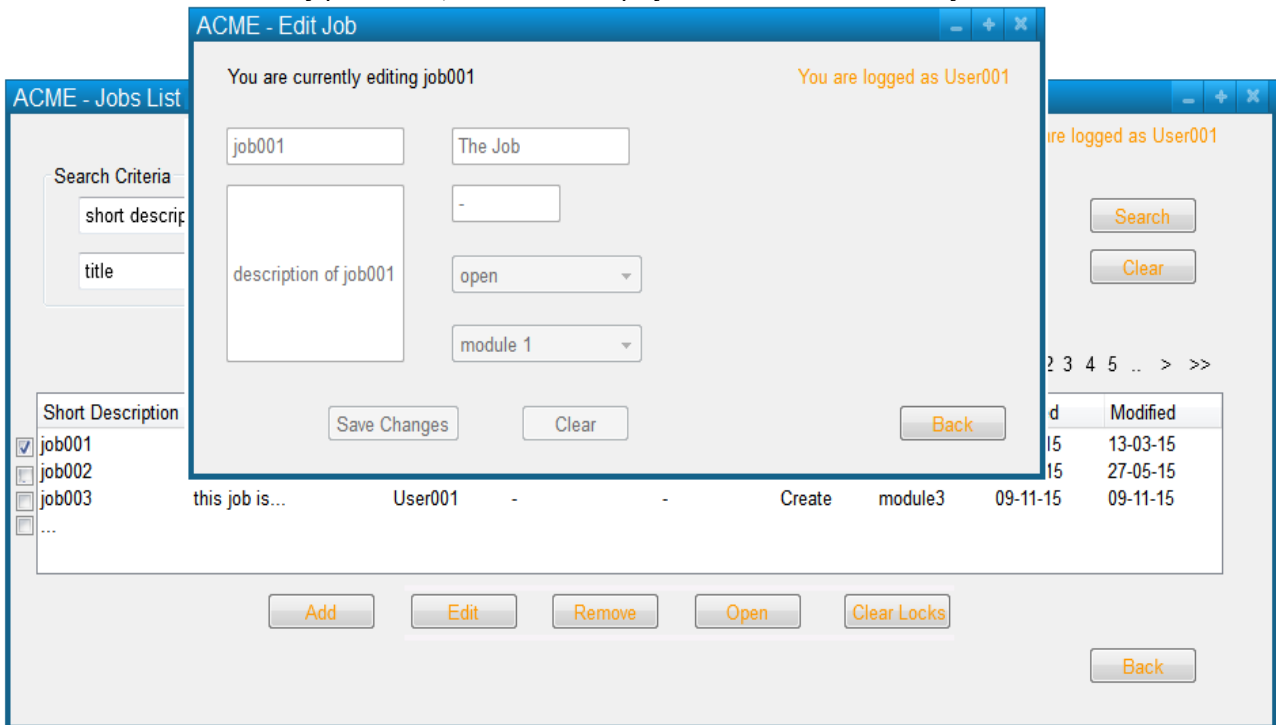
Open

Clear Locks

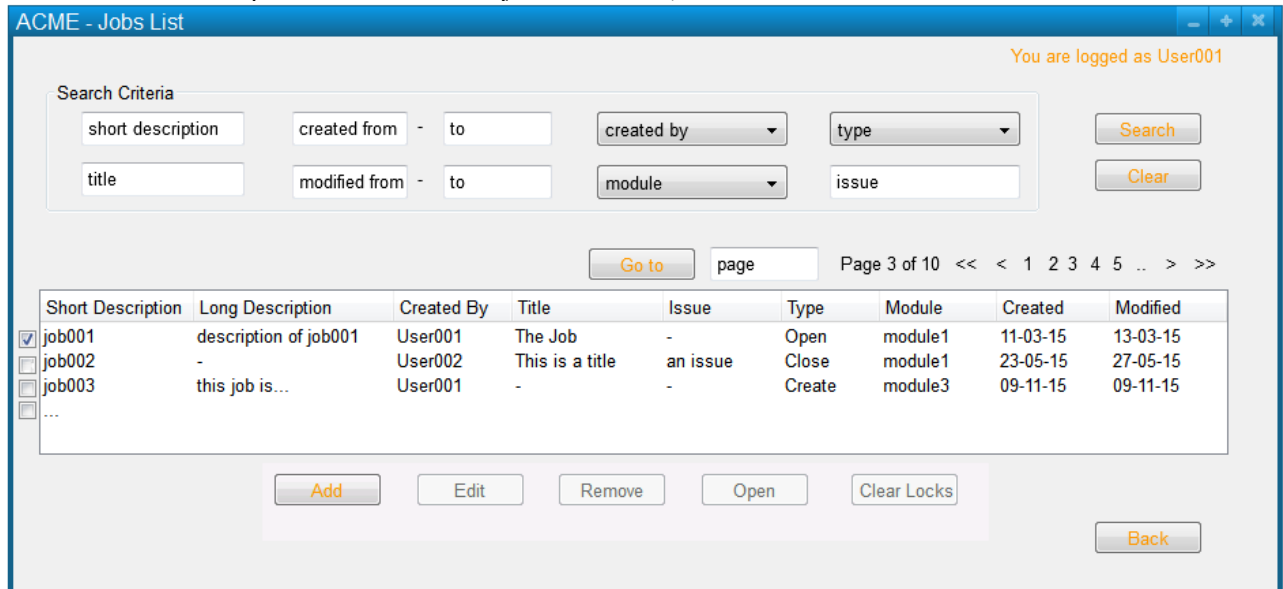
Back

Extensions

2a.1 If the User has read-only permission, then ACME displays the Job Info* in read only mode The use case fails.



2b.1 If the User has no permission to access to job information, then the use case fails



Use Case: Save Job

Goal This use case allows a User to save a job

Level userGoal

Priority 1

Frequency continuously

Primary Actor User

Pre-Conditions The User has a Dirty Job* open in read-write mode

Main success scenario

The screenshot displays the 'ACME - Jobs List' application interface. A modal dialog box titled 'ACME - Job 001*' is open in the foreground, allowing the user to save a job. The dialog contains two text areas: 'Some Stuff' with the text 'stuff', 'stuff', 'stuff', and '...'; and 'Other Stuff' with a text input field containing 'stuff' and a dropdown menu also set to 'stuff'. Below these fields are three buttons: 'Save', 'Clear Locks', and 'Close'. In the background, the 'ACME - Jobs List' window is visible, featuring search criteria for 'short description' and 'title', a table of job entries, and a 'Back' button. The table data is as follows:

	Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input checked="" type="checkbox"/>	job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/>	job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/>	job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/>	...								

1. The User requests to save a job.
2. ACME confirms the operation; job details are saved into Job_Info* and job state in Jobs* is changed into clean.

ACME - Jobs List

Search Criteria

short description

title

ACME - Job 001

Some Stuff

stuff

stuff

stuff

...

Other Stuff

stuff

stuff

stuff

are logged as User001

Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input checked="" type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/> ...								

2 3 4 5 .. > >>

Use Case: Delete Job

Goal This use case allows a user to delete one or more jobs

Level userGoal

Priority 1

Frequency periodically

Primary Actor User

Pre-Conditions A Job List* is displayed.

Main success scenario

ACME - Jobs List

You are logged as User001

Search Criteria

short description created from - to created by type Search

title modified from - to module issue Clear

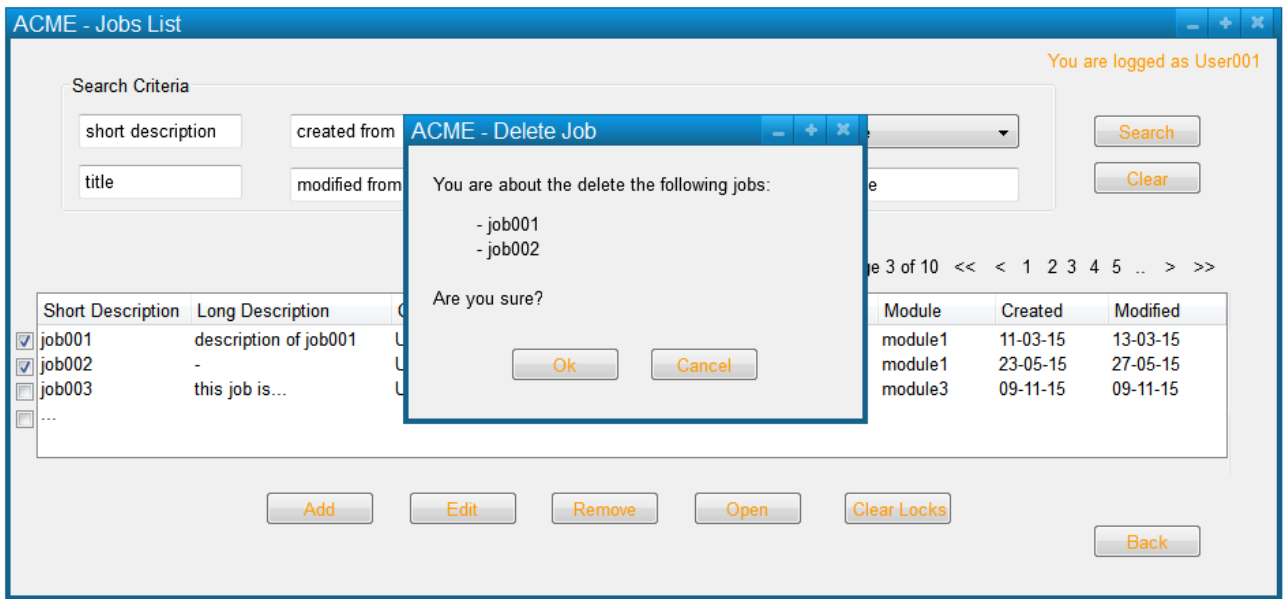
Go to page Page 3 of 10 << < 1 2 3 4 5 .. > >>

Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/> ...								

Add Edit Remove Open Clear Locks

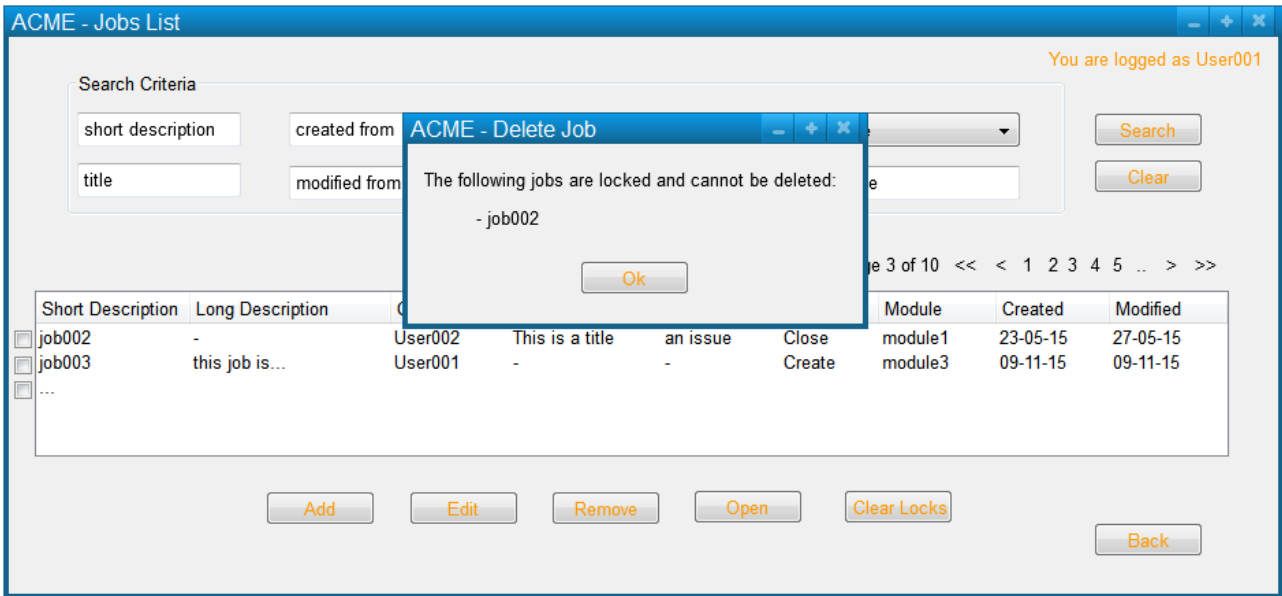
Back

1. If the User has Delete permission* on some Jobs*, then he asks to delete some of them.
2. ACME asks for a confirmation.



3. The User confirms the intention to delete the selected jobs.

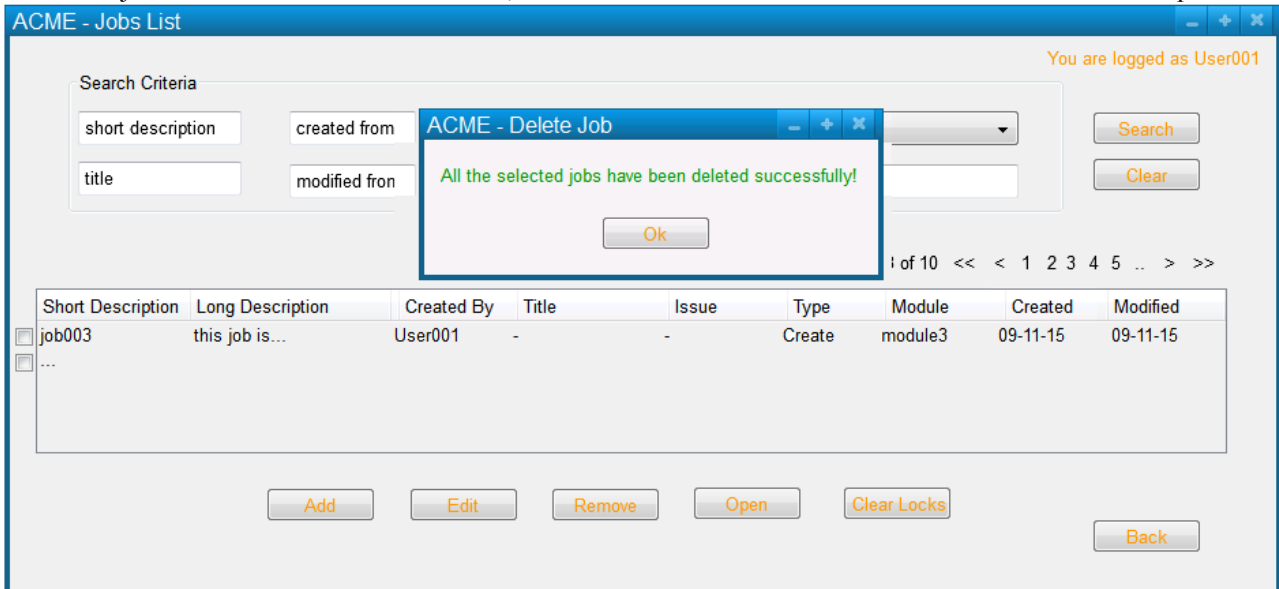
4. If there are jobs or modules locked in **Jobs***, then ACME deletes all the jobs where no job or module locks in **Jobs*** exists and informs the User that some of them are locked and cannot be deleted, providing **Lock_Info***; the **Job List*** is updated.



Extensions

3a.1. The External User aborts the operation. The use case fails.

4a.1. If no jobs or modules are locked in **Jobs***, then ACME deletes all of them from **Jobs***; the **Job List*** is updated.



Use case: Administrator Clear Job Lock

Goal This use case allows the administrator to clear a lock on a job

Notes

- This can potentially occur when a job is open by a user, and then something happens to that user's computer, such as the computer losing power.
- The Administrator can clear a lock on any job

Level userGoal

Priority 1

Frequency frequently

Primary Actor Administrator

Pre-Conditions Administrator_Logged* is true, a Job List* is displayed

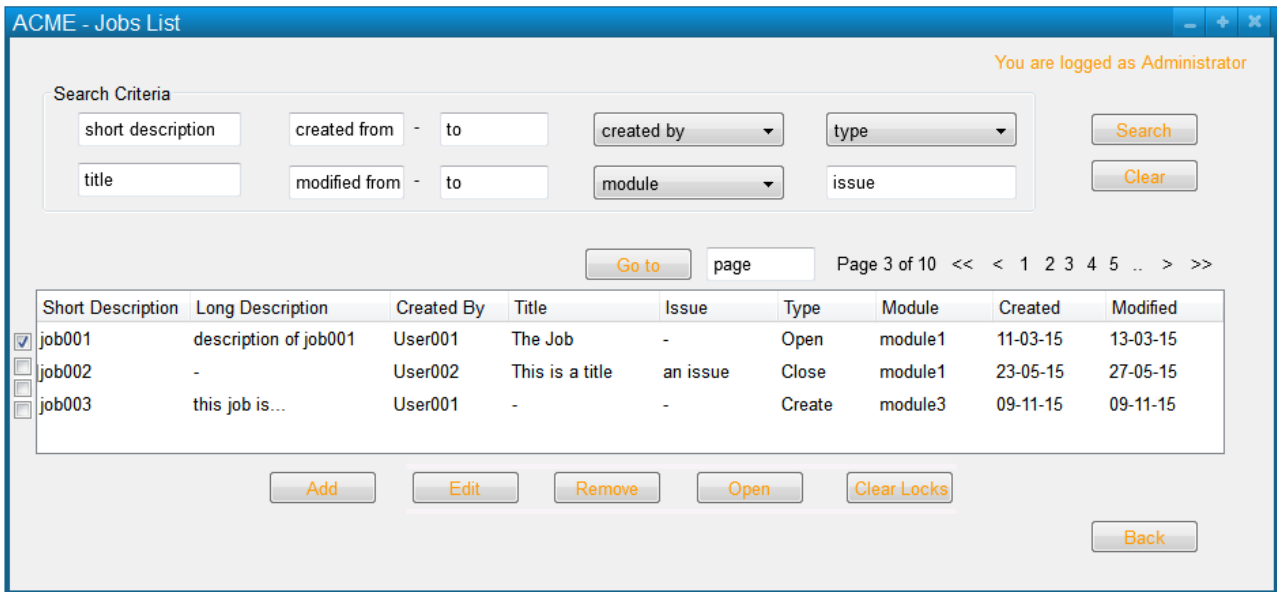
Main success scenario

The screenshot shows a web application window titled "ACME - Jobs List". In the top right corner, it says "You are logged as Administrator". Below this is a search criteria section with two rows of input fields and dropdown menus. The first row includes "short description", "created from - to", "created by", and "type". The second row includes "title", "modified from - to", "module", and "issue". There are "Search" and "Clear" buttons to the right of these fields. Below the search section is a "Go to" button, a "page" input field, and pagination information: "Page 3 of 10" with navigation arrows. The main part of the interface is a table with the following data:

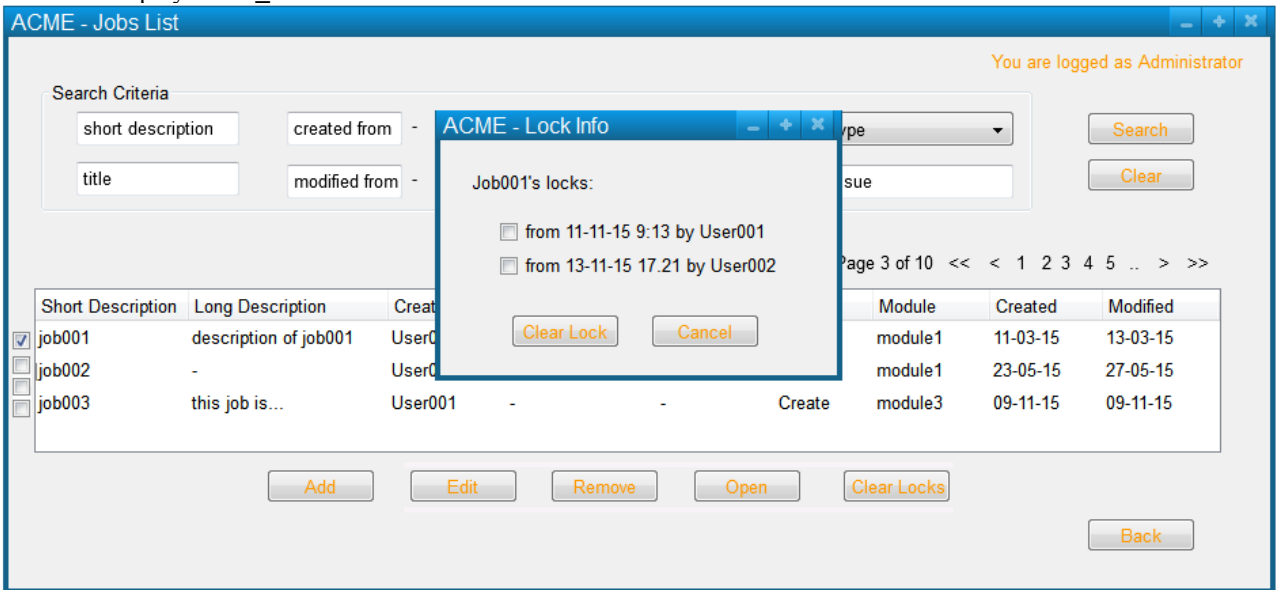
Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15

At the bottom of the interface, there are several action buttons: "Add", "Edit", "Remove", "Open", "Clear Locks", and "Back".

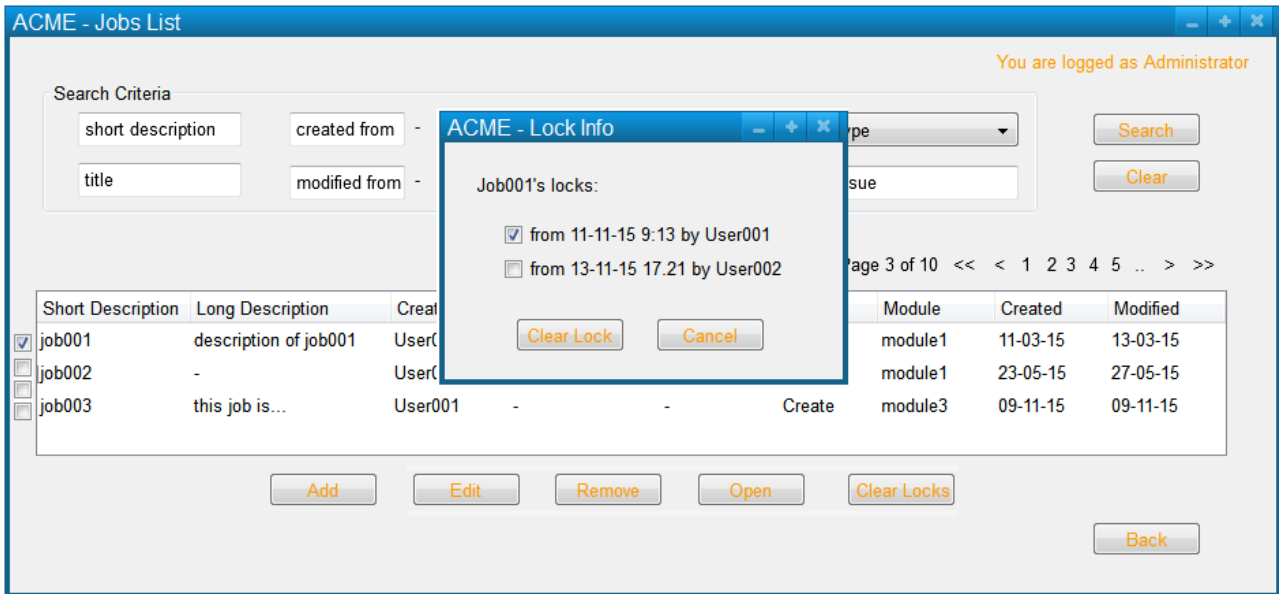
1. The Administrator requests to clear the locks on a job.



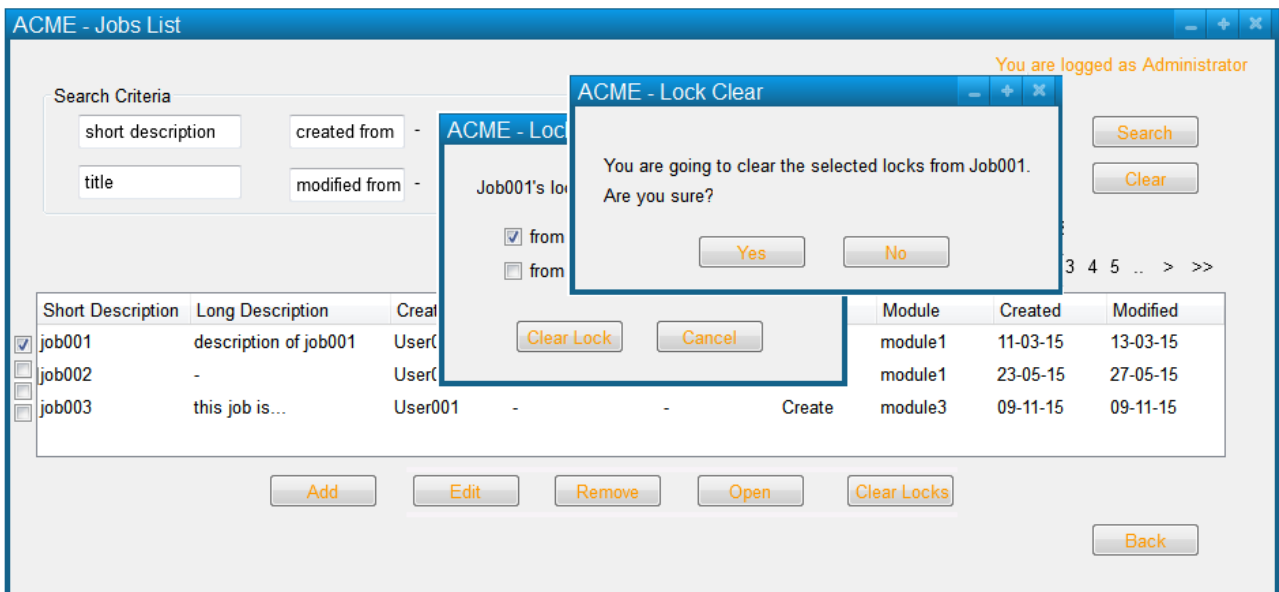
2. ACME displays Lock_Info* recovered from Jobs*.



3. The Administrator selects the locks he wants to clear.

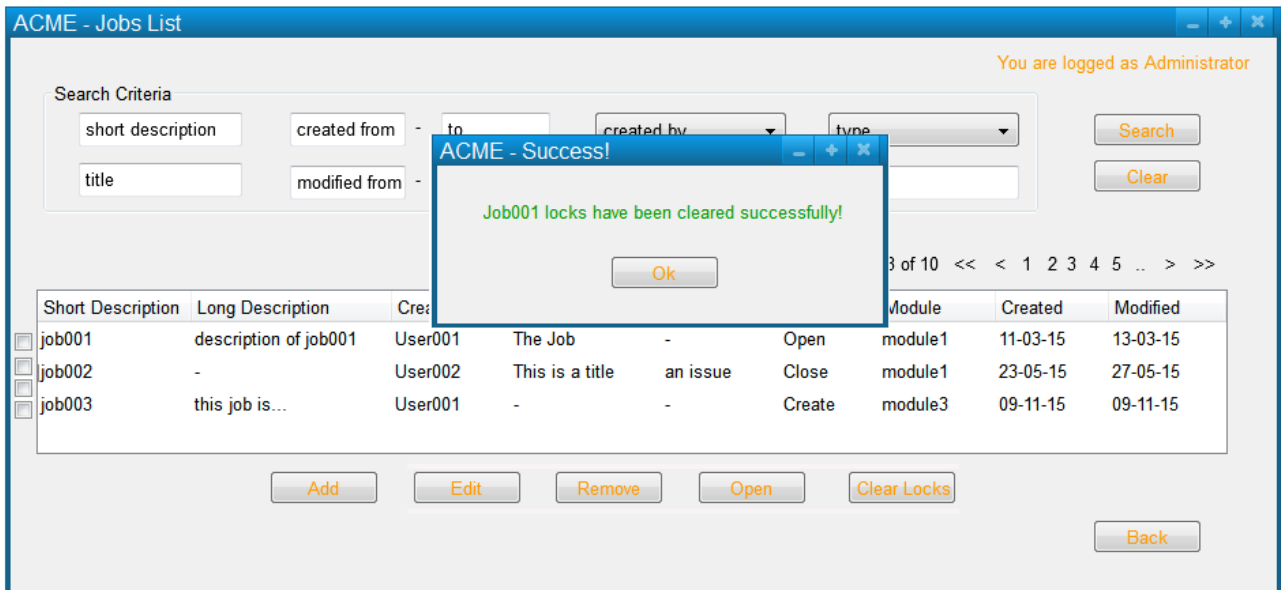


4. ACME asks for a confirmation.



5. The Administrator confirms the operation.

6. ACME confirms the removal; the selected locks in Jobs* are removed and Lock_Info* is updated.



Extensions

5a.1. The Administrator aborts the operation. The use case fails.

Use case: External User Clear Job Lock

Goal This use case allows an external user to lock a job.

Notes

- This can potentially occur when a job is open by a user, and then something happens to that user's computer, such as the computer losing power.
- In single-user mode, a user can clear a lock on any job
- In multi-user mode, other users can clear locks on their jobs

Level userGoal

Priority 1

Frequency frequently

Primary Actor External User

Pre-Conditions A Job List* is displayed

Main success scenario

ACME - Jobs List

You are logged as User001

Search Criteria

short description created from - to created by type Search

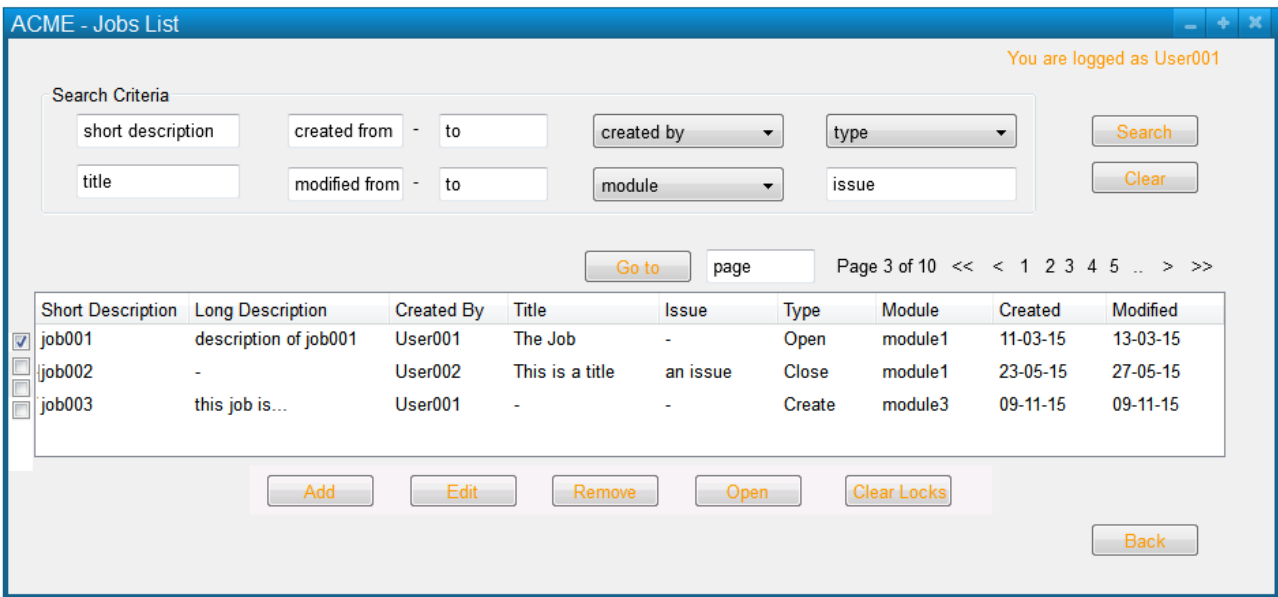
title modified from - to module issue Clear

Go to page Page 3 of 10 << < 1 2 3 4 5 .. > >>

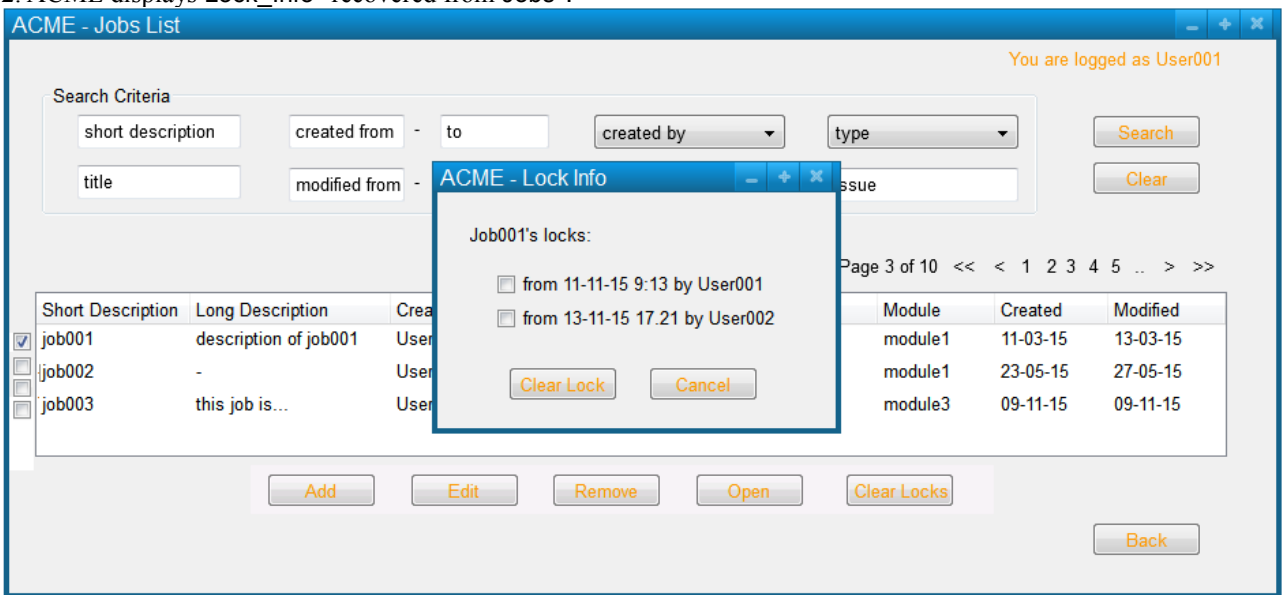
Short Description	Long Description	Created By	Title	Issue	Type	Module	Created	Modified
<input type="checkbox"/> job001	description of job001	User001	The Job	-	Open	module1	11-03-15	13-03-15
<input type="checkbox"/> job002	-	User002	This is a title	an issue	Close	module1	23-05-15	27-05-15
<input type="checkbox"/> job003	this job is...	User001	-	-	Create	module3	09-11-15	09-11-15
<input type="checkbox"/> ...								

Add Edit Remove Open Clear Locks Back

1. The External User requests to clear the locks on a job.



2. ACME displays Lock_Info* recovered from Jobs*.



3. If Security_Enabled* is true and the locks on the job are owned by the External User, then the External User selects the locks he wants to clear.

Search Criteria

short description

created from

-

to

created by

type

Search

title

modified from

issue

Clear

ACME - Lock Info

Job001's locks:

- from 11-11-15 9:13 by User001
- from 13-11-15 17:21 by User002

Clear Lock Cancel

Page 3 of 10 << < 1 2 3 4 5 .. > >>

Short Description	Long Description	Created
job001	description of job001	User001
job002	-	User001
job003	this job is...	User001

Module	Created	Modified
module1	11-03-15	13-03-15
module1	23-05-15	27-05-15
module3	09-11-15	09-11-15

Add

Edit

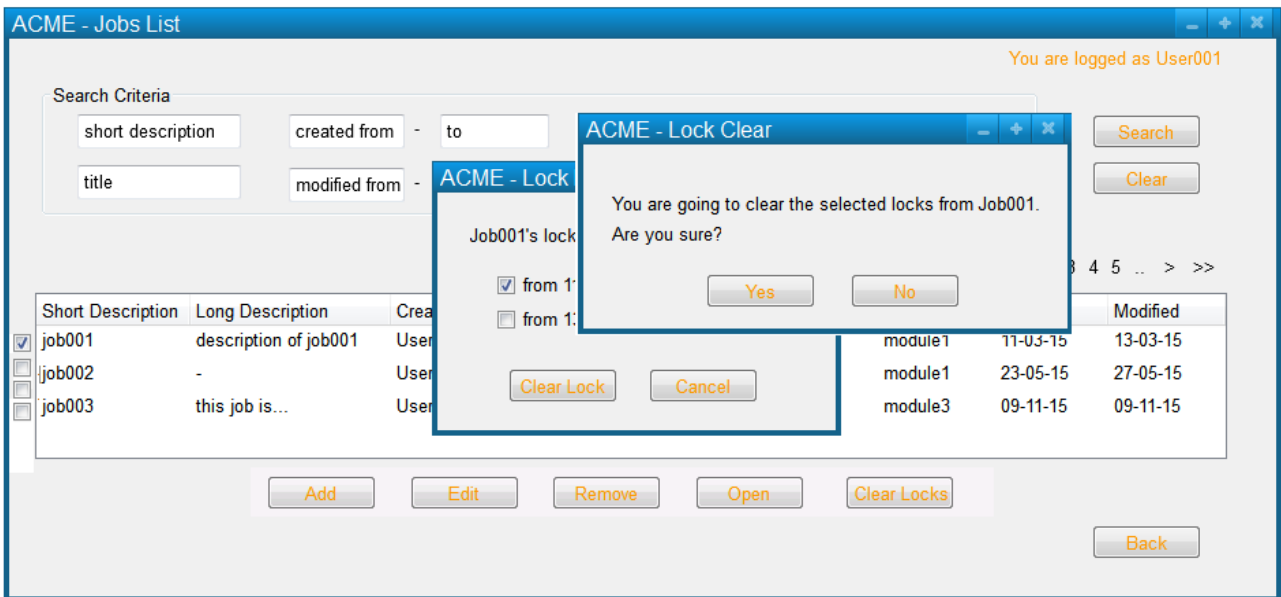
Remove

Open

Clear Locks

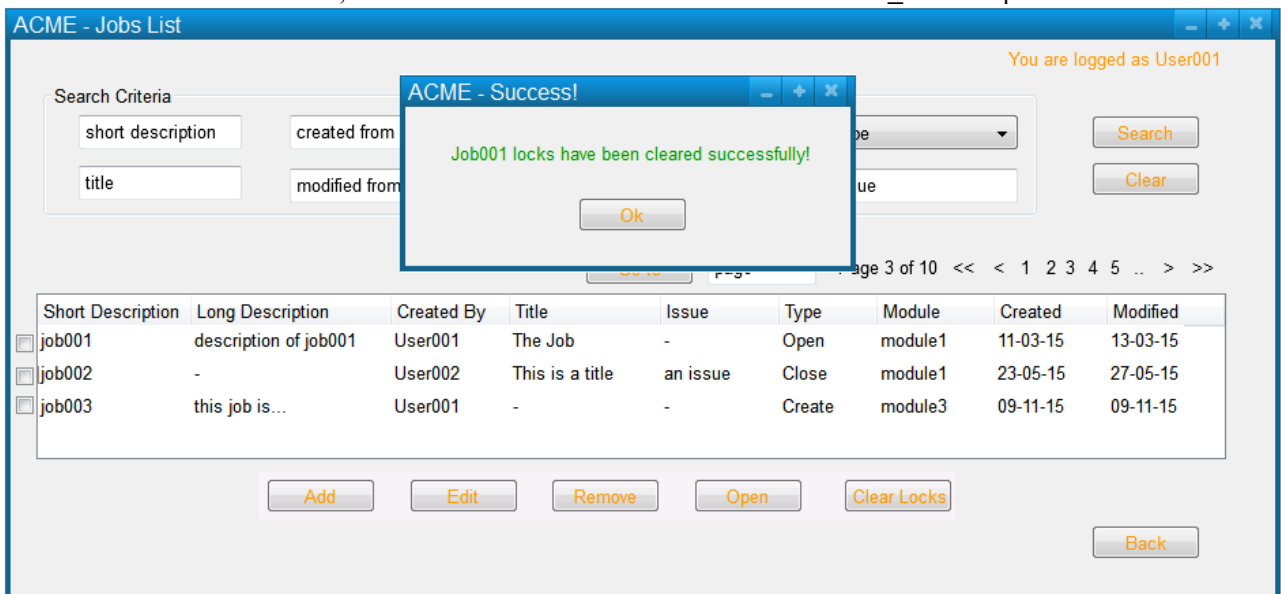
Back

4. ACME asks for a confirmation



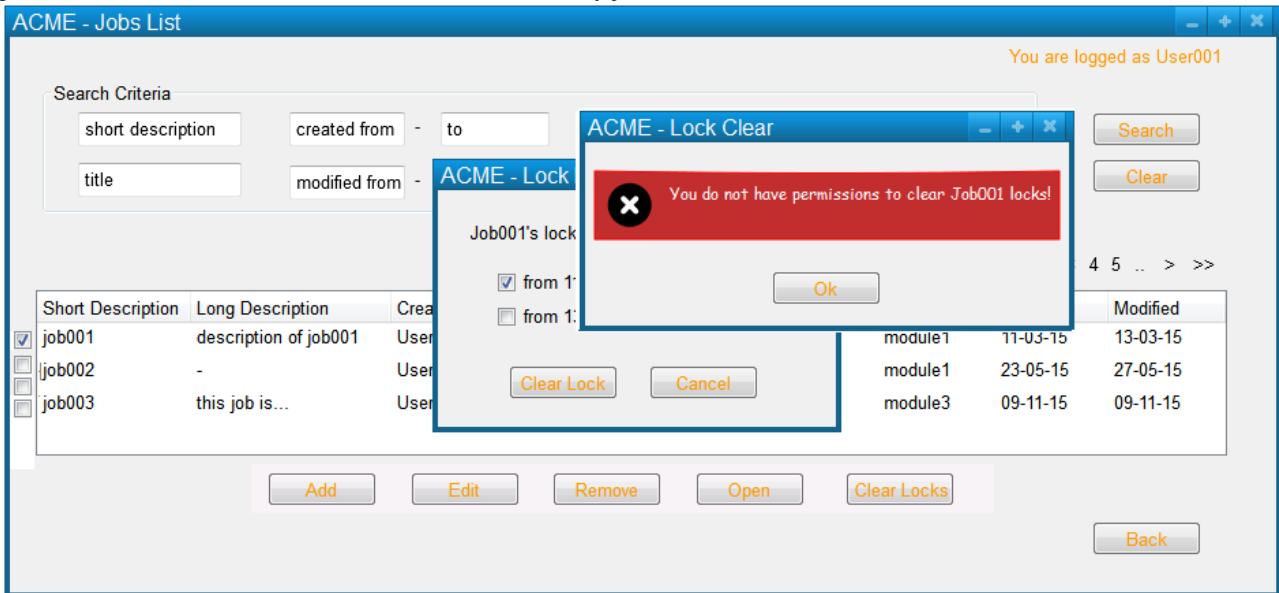
5. The External User confirms the operation.

6. ACME confirms the removal; the selected locks in Jobs* are removed and Lock_Info* is updated.



Extensions

3a.1. If the locks on the job are by a user different from Unknown User and the External User does not have edit permission, then ACME informs that he cannot clear any job lock. The use case fails.



3b.1. If the locks on the job are by a user different from Unknown User and the External User has edit permission, then the External User selects the locks he wants to clear. The use case continues at step 4.

3c.1. If Security_Enabled* is false, then the External User selects the locks he wants to clear. The use case continues at step 4.

5a.1. The External User aborts the operation. The use case fails.

Use case: Exit Application

Goal This use case allows the user to exit from the application by closing the editor

Note The User may have one or more ACME sessions running in separate editors

Level userGoal

Priority 1

Frequency exceptionally

Primary Actor User

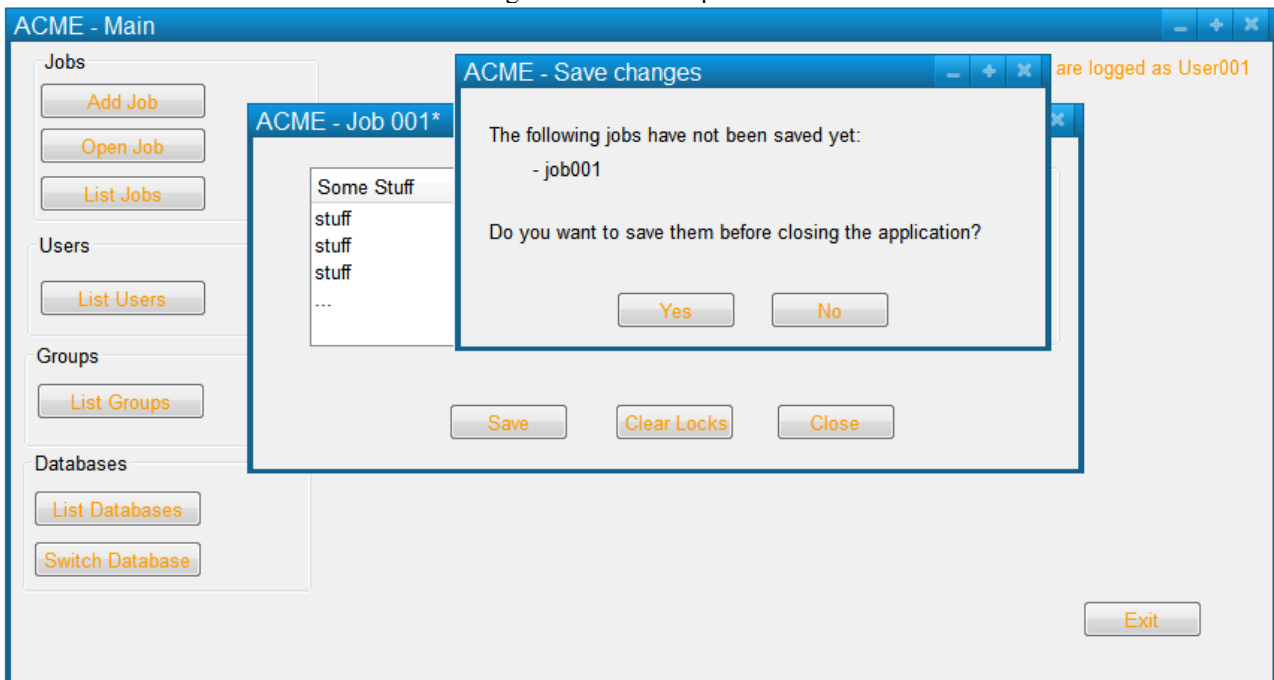
Main success scenario

1. The User closes the editor.
2. If there is no job in the editor, then ACME shuts itself down.

Extensions

2a.1. If there is a non Dirty Job* in Jobs* in the editor, then ACME closes the editor and shuts itself down; any locks of the User from Open Jobs* is removed.

2b.1. If there is a Dirty Job* in Jobs* in the editor, then ACME informs the User that there is a job that has been modified and asks whether save or not the changes or aborts the operation.



2b.2. The User answers to save the changes.

2b.3. ACME closes the editor and shuts itself down; the Dirty Job* is saved, and any locks of the User from Open Jobs* is removed.

2b.2.a.1. The Users answers not to save the changes.

2b.2.a.2. ACME closes the editor and shuts itself down; any locks of the User from Open Jobs* is removed.

2b.2.b.1. The User aborts the operation. The use case fails.

Use Case: Add Database

Goal This use case allows a user to add a ACME database to the list of known databases

Notes

- Any user can create a new database alias
- Users do not share database aliases; they are saved on their local computers under their own info
- Database definitions will be stored under the user's home directory on their computer

Level userGoal

Priority 4

Frequency exceptionally

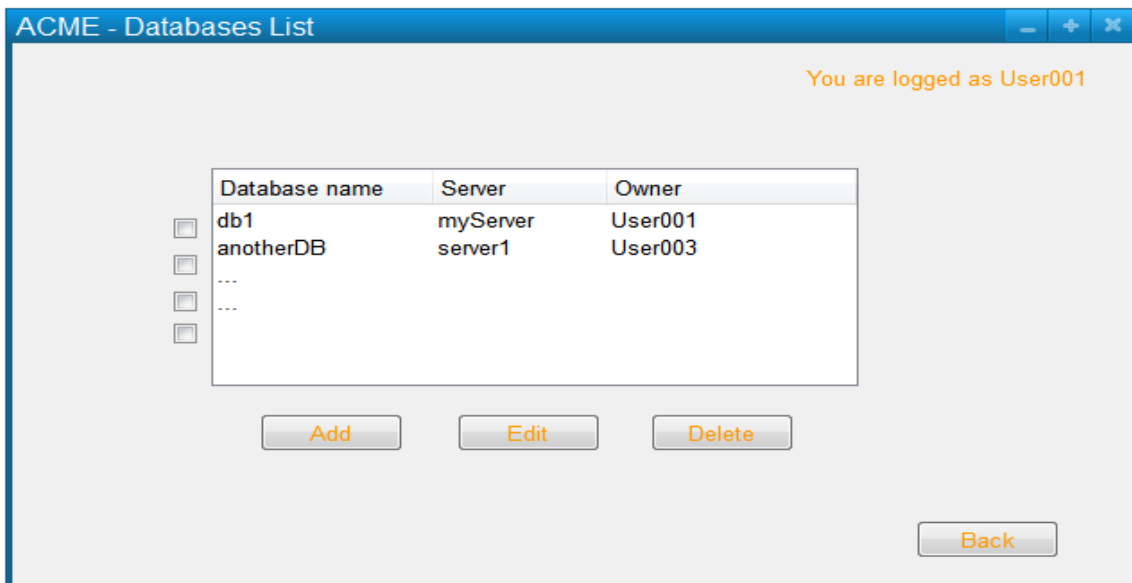
Primary Actor User

Pre-Conditions

- One or more aliases are in Database Aliases*
- The list of Database Aliases* is shown

Actor User

Main success scenario



1. The User requests to add a new database.
2. ACME prompts the User for the following Database Alias Info*.

ACME - Add Database

You are logged as User001

Insert the following database's info

name server path

admin id admin password

Test Connection

Add Database Back

3. The User enters Database Alias Info* and confirms.

ACME - Add Database

You are logged as User001

Insert the following database's info

my new database myservers

root root

Test Connection

Add Database Back

4. If Working Database Connection* is true, Database Alias Info* is complete, Database Name* is unique, Database User ID* and Database Password* are valid, then ACME confirms the creation of a new database connection; Database Alias Info* is added in Database Aliases*.

ACME - Add Database

You are logged as User001

Insert the following database's info

my new database myservers

root root

Test Connection

Add Database Back

ACME - Success!

The database connectivity has been tested positively
and the database has been successfully created.

Ok

Extensions

3a.1. The User aborts the operation. The use case fails.

4a.1. If Database Alias Info* is incomplete, then ACME informs the Administrator of which required fields are missing. The use case fails.

The screenshot shows a window titled "ACME - Add Database" with a blue header bar. In the top right corner, it says "You are logged as User001". Below the header, the text "Insert the following database's info" is displayed. There are four input fields: the first contains "my new database", the second contains "myserver", the third is empty, and the fourth contains "root". To the right of these fields is a "Test Connection" button. At the bottom, there are "Add Database" and "Back" buttons. A prominent red error box with a black 'X' icon contains the text "Incomplete information!".

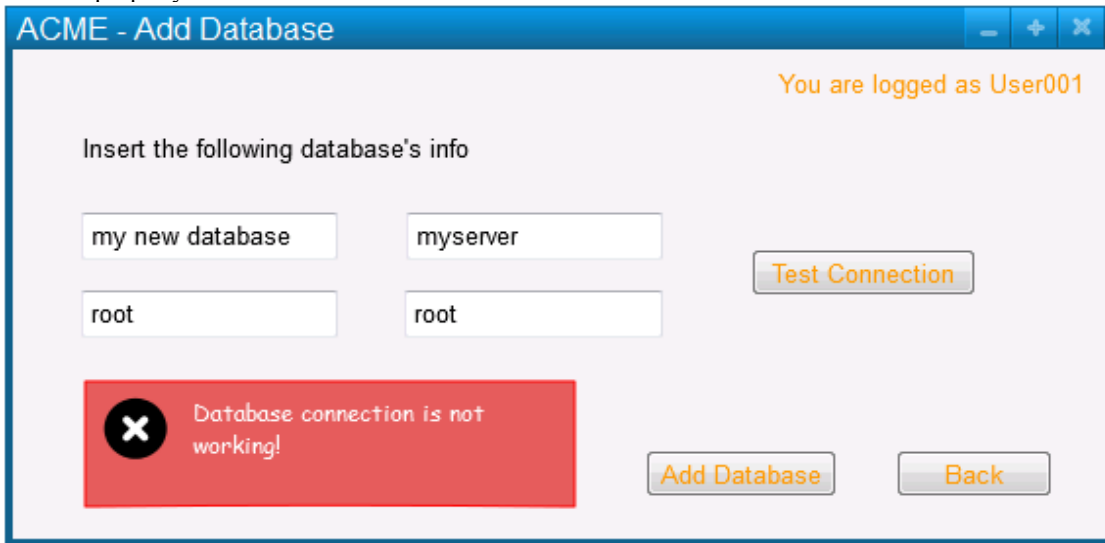
4b.1. If Database Name* is not unique, then ACME informs the Administrator that he must choose a different name. The use case fails.

The screenshot shows the same "ACME - Add Database" window. The input fields now contain "my new database", "myserver", "root", and "root". The "Test Connection" button is disabled. The red error box now displays the message "Database name is already in use. Choose a different one!".

4c.1. If Database User ID* is not valid, then ACME informs the Administrator that he must choose a different one, and valid id criteria is displayed. The use case fails.

4d.1. If Database Password* is not valid, then ACME informs the Administrator that he must choose a different one, and valid password criteria is displayed. The use case fails.

4e.1. If Working Database Connection* is false, then ACME informs the Administrator that the connection has not been set properly. The use case fails.



Post-Conditions An alias for a new database connection is created and stored.

Use Case: Delete Database

Goal This use case allows to delete a database alias

Level userGoal

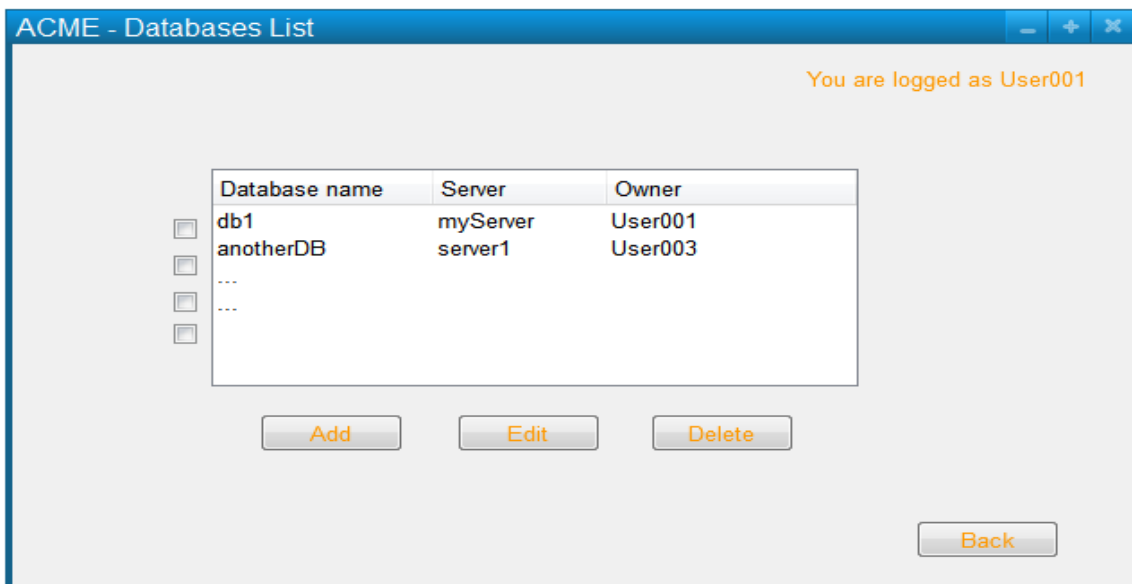
Priority 4

Frequency exceptionally

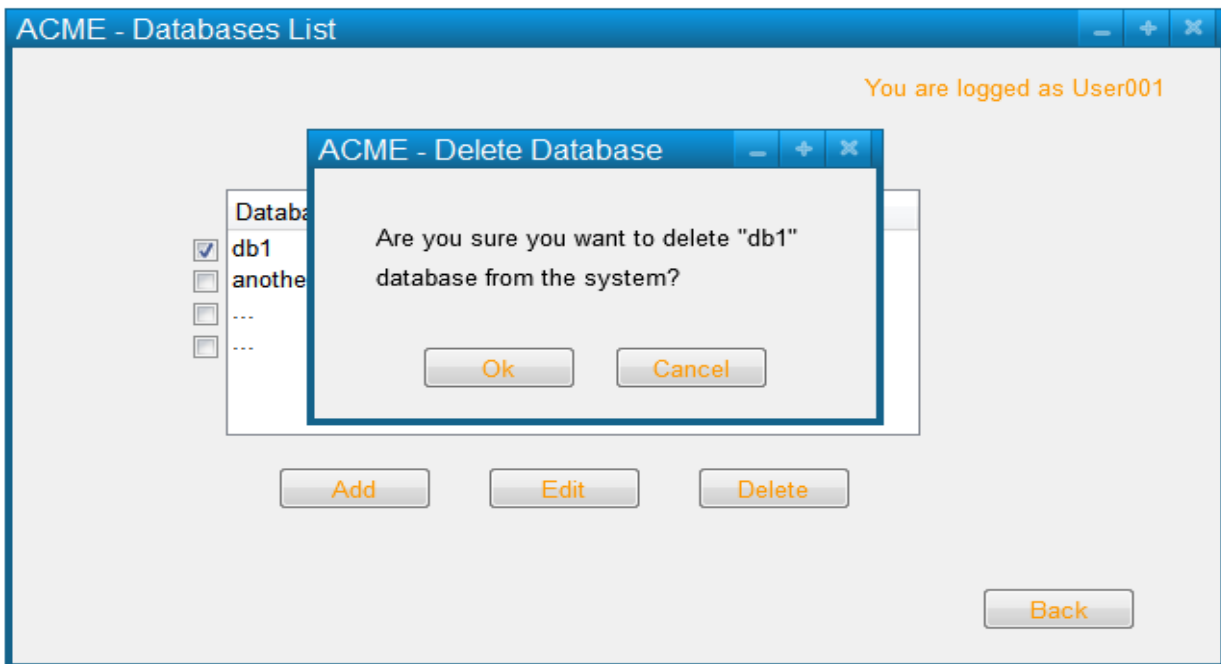
Primary Actor User

Pre-Conditions One or more aliases are in Database Aliases*, and the list of Database Aliases* is shown

Main success scenario

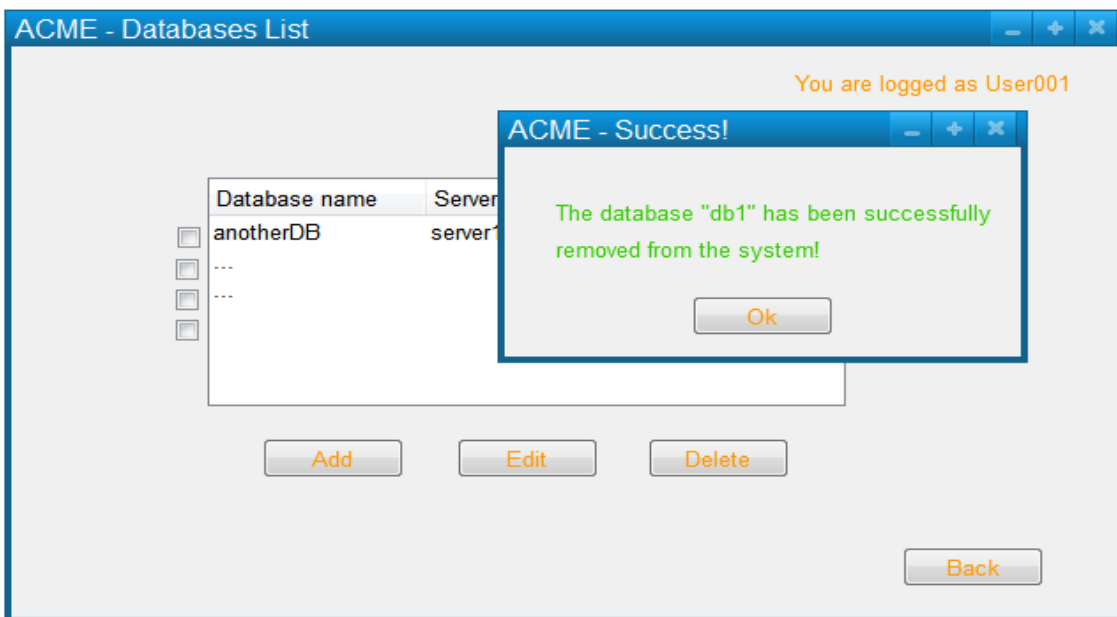


1. The User requests to delete a database.
2. If the User has the rights to delete the selected database, then ACME asks the User for a confirmation.



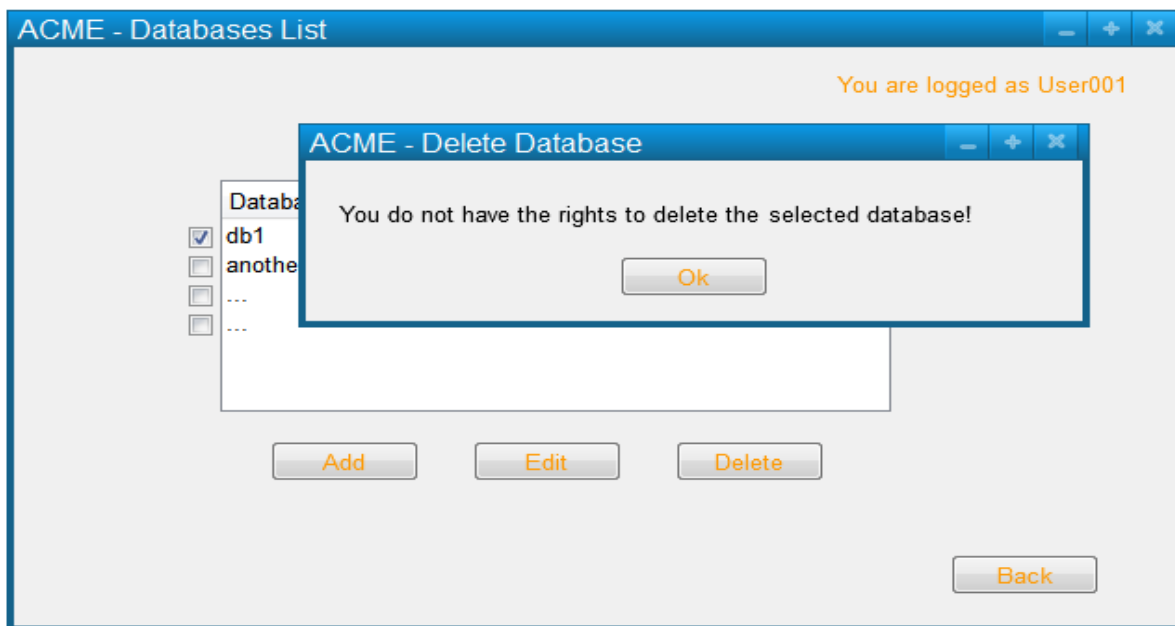
3. The User confirms the intention to delete the selected database.

4. ACME informs of the deletion; Database Alias Info* is removed from Database Aliases*



Extensions

2a.1 If the User has not the rights to delete the selected database, then ACME informs that he cannot delete the selected database. The use case fails.



3a.1. The User aborts the operation. The use case fails.

Use Case: Edit Database

This use case describes the process of editing a ACME database alias.

Level userGoal

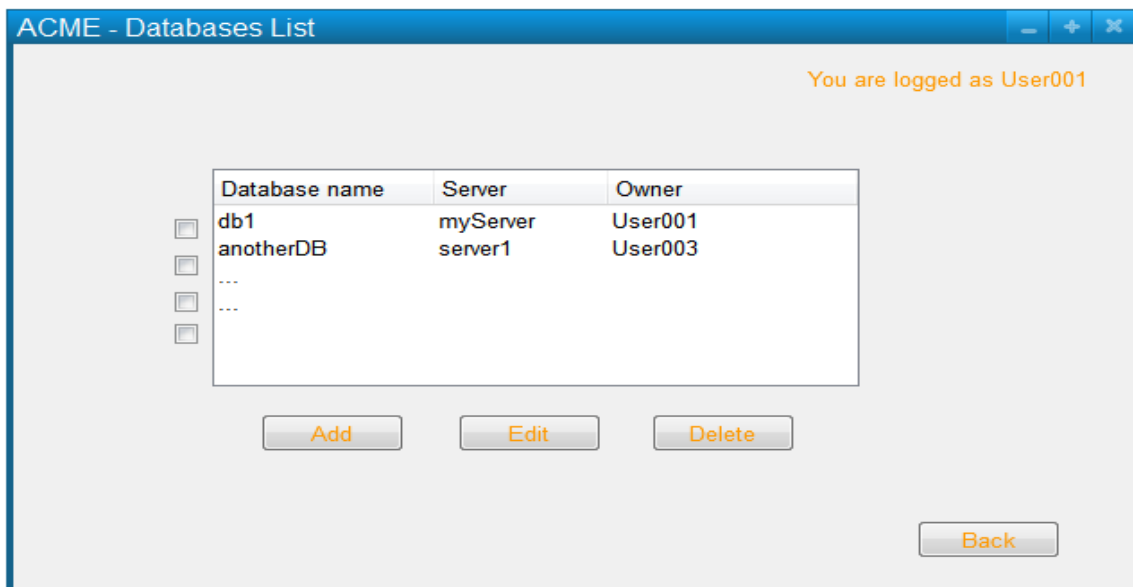
Priority 4

Frequency exceptionally

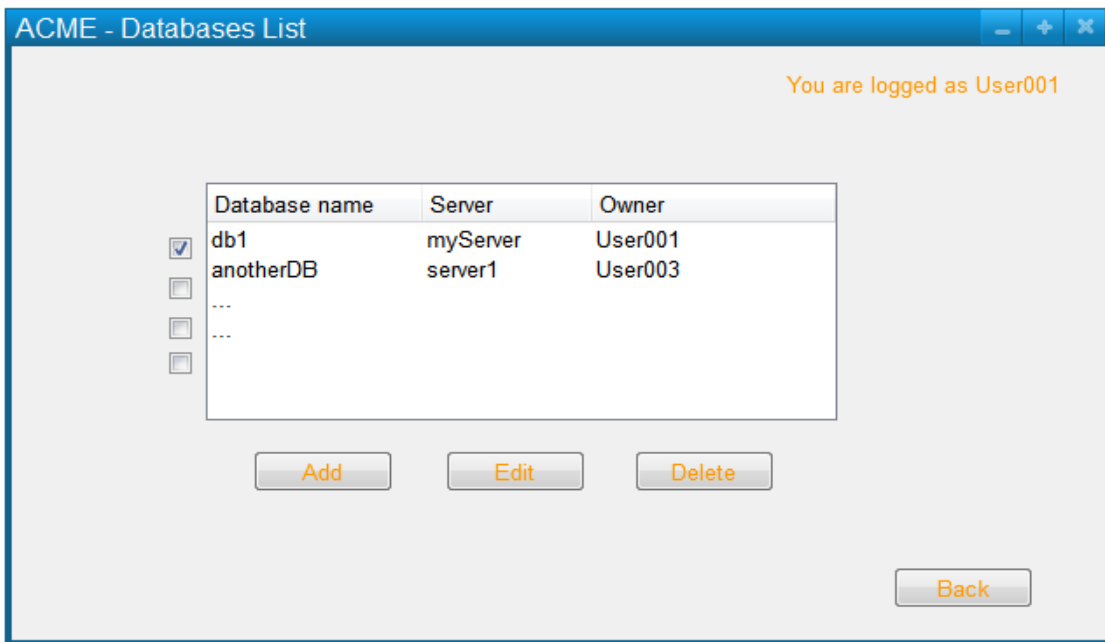
Primary Actor User

Pre-Conditions One or more aliases are in Database Aliases*, the list of Database Aliases* is shown

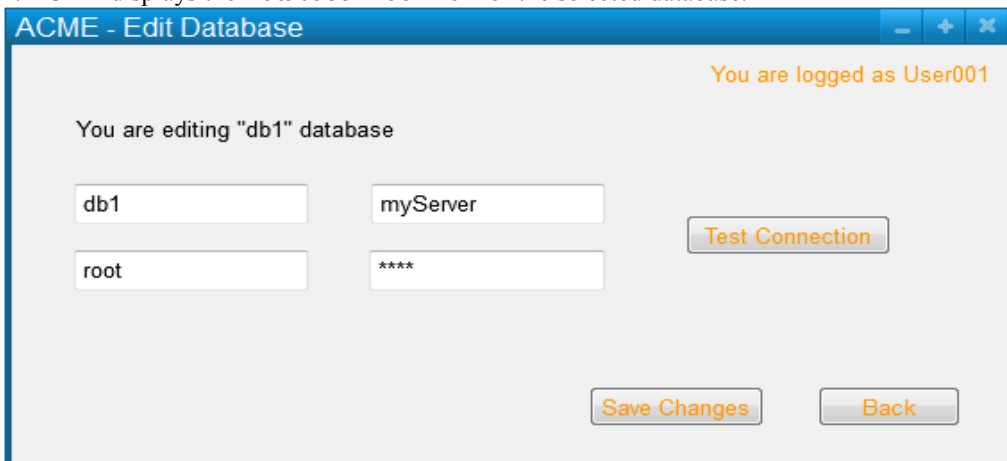
Main success scenario



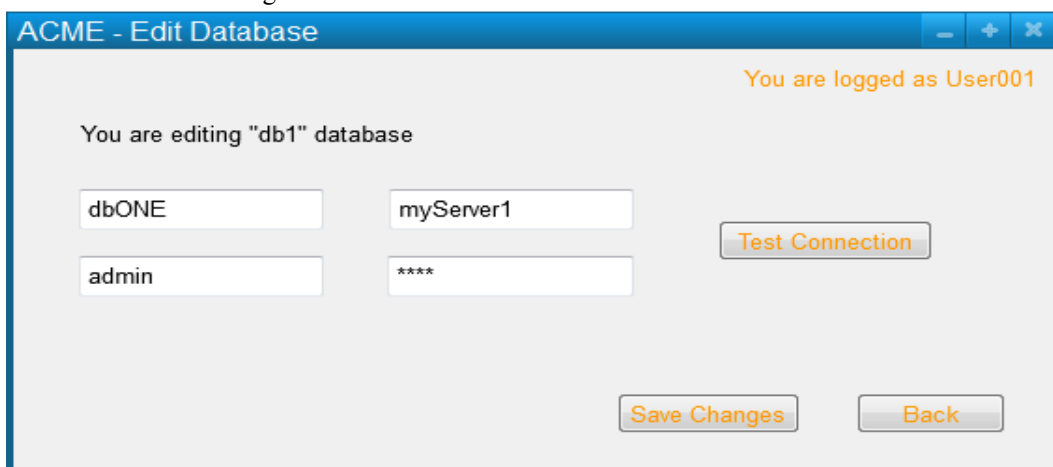
1. The User requests to edit a database.



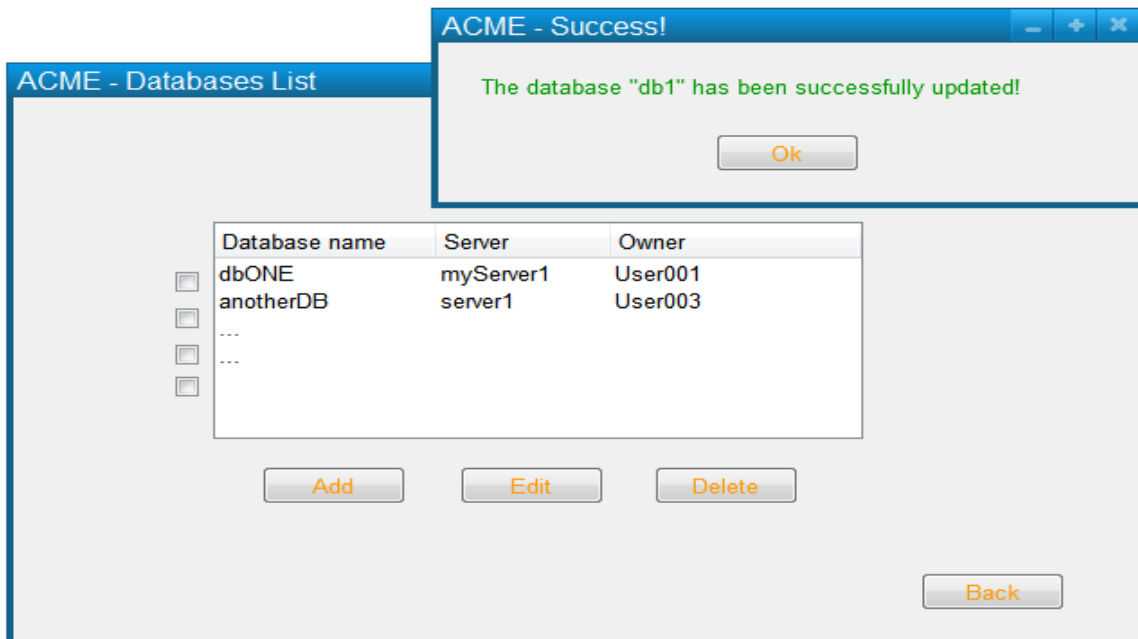
2. ACME displays the Database Alias Info* for the selected database.



3. The User makes changes to Database Alias Info* and confirms.



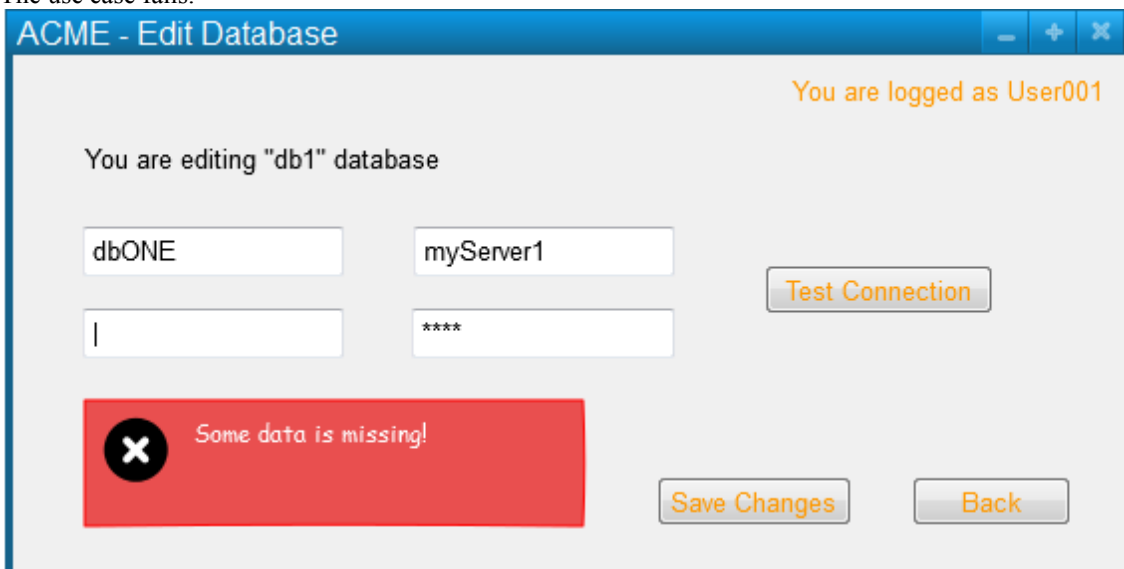
4. If Working Database Connection* is true, Database Alias Info* is complete, Database Name* is unique, Database User ID* and Database Password* are valid, then ACME informs that the database has been updated; Database Alias Info* for that database alias is updated.



Extensions

3a.1. The User aborts the operation. The use case fails.

4a.1 If Database Alias Info* is incomplete, then ACME informs the Administrator of which required fields are missing. The use case fails.



4b.1 If Database Name* is not unique, then ACME informs the Administrator that he must choose a different name. The use case fails.

4c.1 If Database User ID* is not valid, then ACME informs the Administrator that he must choose a different one, and valid id criteria is displayed. The use case fails.

4d.1 If Database Password* is not valid, then ACME informs the Administrator that he must choose a different one, and valid password criteria is displayed. The use case fails.

4e.1 If Working Database Connection* is false, then ACME informs the Administrator that the connection has not been set properly. The use case fails

Use Case: List Known Databases

Goal This use case allows a user to list the known database aliases.

Level userGoal

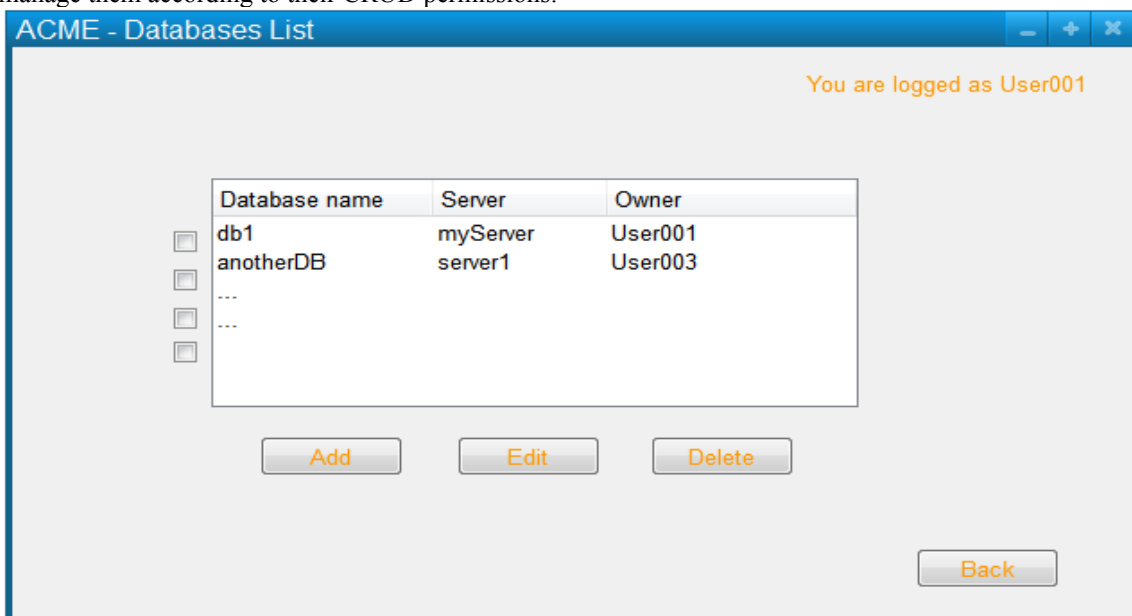
Priority 4

Frequency exceptionally

Primary Actor User

Main success scenario

1. The User requests to list all the known databases.
2. ACME displays all Database Aliases*, including the server where the database is installed and the options to manage them according to their CRUD permissions.



Post-Conditions The list of Database Aliases* is shown.

Use Case: Switch Database

Goal This use case allows a user to change the current database.

Level userGoal

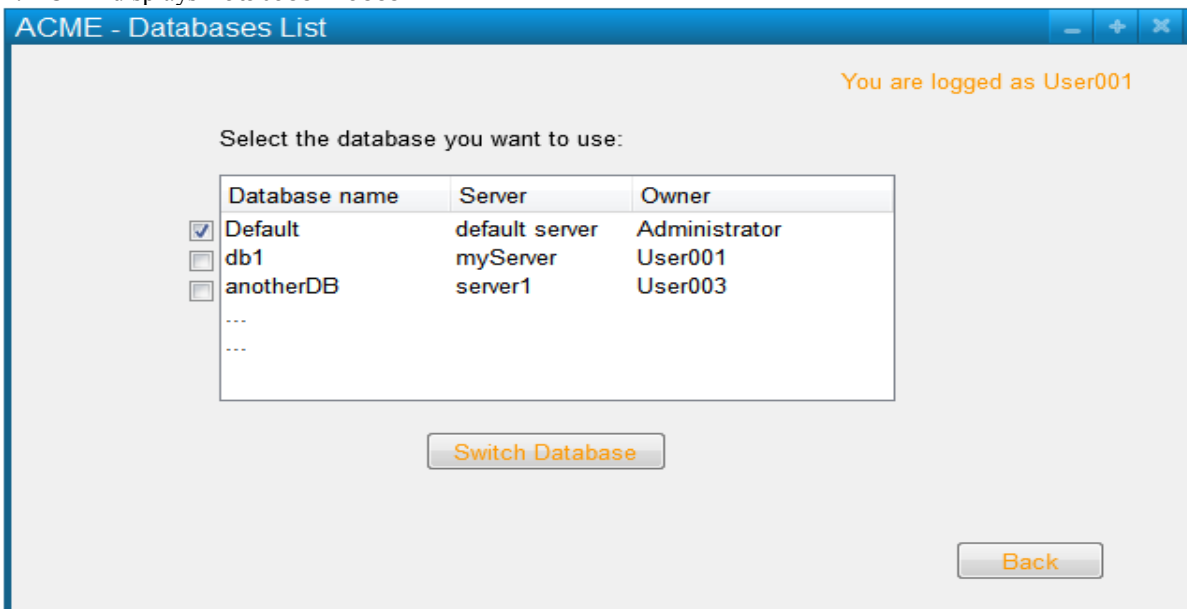
Priority 4

Frequency exceptionally

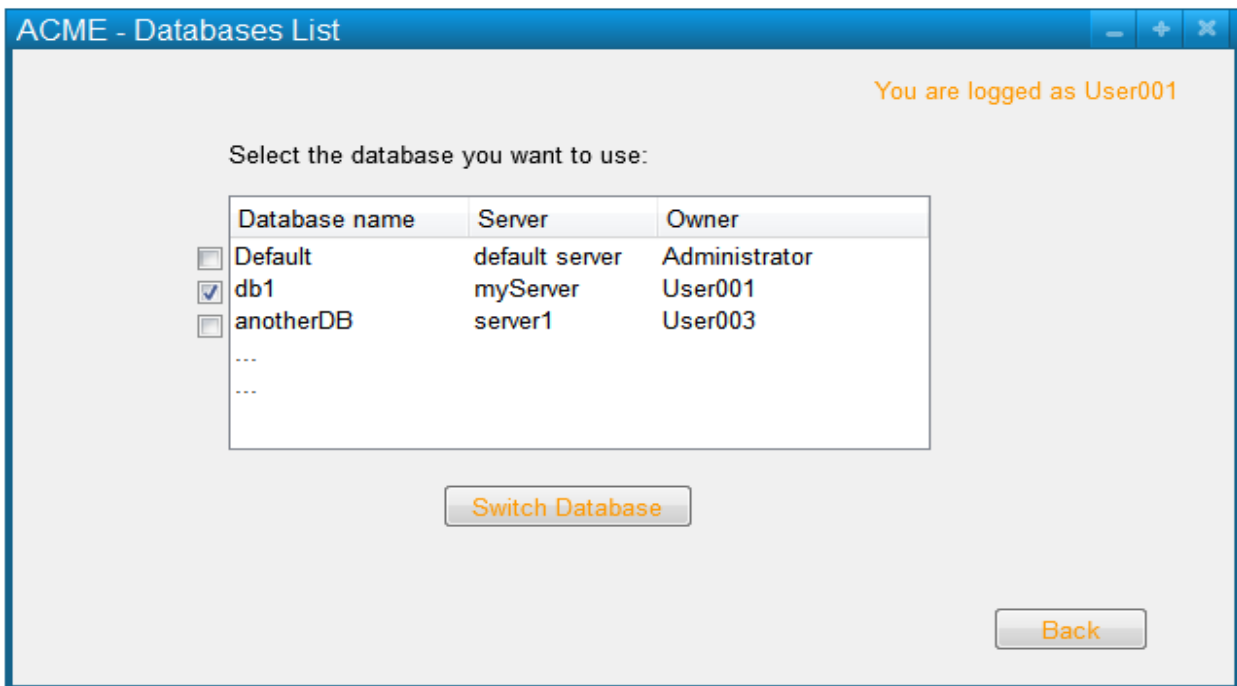
Primary Actor User

Main success scenario

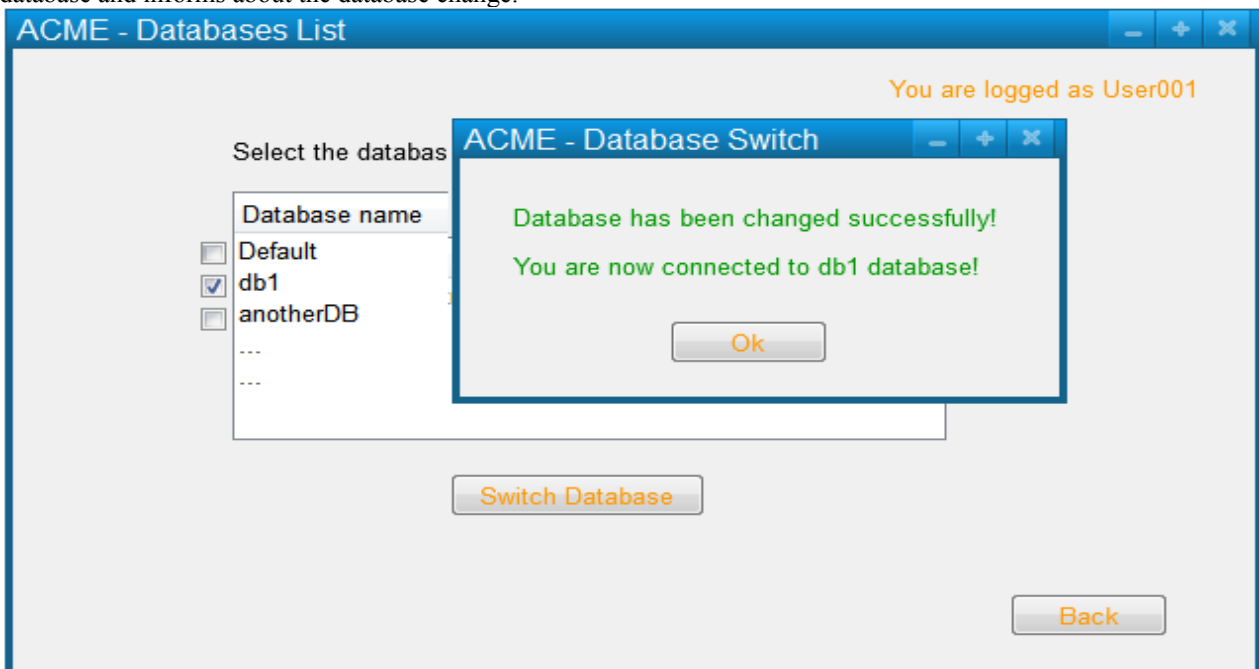
1. The User requests to switch to a different database from the current one.
2. ACME displays Database Aliases*.



3. The User selects a database he would like to connect to.

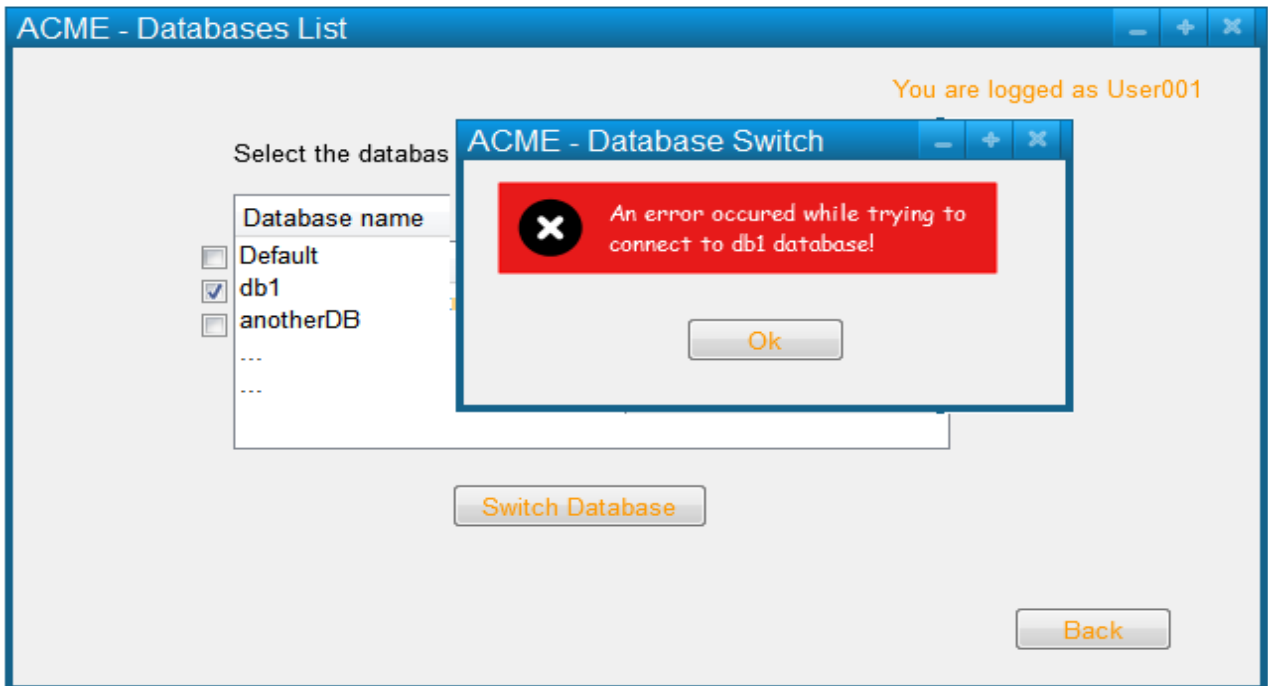


4. If no error is encountered while connecting the User to the database, then ACME connects the User to the desired database and informs about the database change.



Extensions

4a.1. If an error is encountered while connecting the User to the database, then ACME displays an error message to the User. The use case fails



Glossary

Data Entry

Access Level: the access level to ACME Functional Areas*. It can be *hidden* for a not accessible functional area, *view* for a read-only one, and *edit* for read-write permissions

Functional Areas: functionalities or actions or operations whose access may be different for different groups of users

Group Info:

- Group Name*
- a total mapping from Functional Areas* into Access Level*

The special group Administrator has specific access levels (they should be edit for every one), whereas the special group Unknown Group has full access to all functional areas other than User and Group management.

Group Name: Valid group names are identifiers built out of A-Z, a-z, 0-9, _, -, and blank spaces, long up to 32 characters, that begin with a character or number.

Lock_Info: locking user identity (or name) and lock date/time

Job_Info:

- Short Job Description (*)
- Customer name (*)
- Long job description
- Title
- Issue
- “date created” and “date modified”
- the “last maintained by” information:
- Job type (Unplanned)
- Modules not further detailed, each one may be optionally locked, i.e. it may have associated a Lock_Info*
- Lock (optional) of type Lock_Info* (a job is locked iff this field is not undefined)

Job List: the list of all Job_Info*

Dirty Job: a job is dirty iff it is in a “dirty” state (i.e., the Job Characteristics have been changed)

Password: The passwords can be up to 20 characters, can contain any ASCII character and are not case sensitive

Search Criteria: the information given by a user to ACME to find some jobs

- Short Job Description
- Created from/to
- Modified from/to
- Created By
- Type
- Title
- Issue
- Module

User_Info: (the parts marked by * are optional)

- Username (Username*)
- Password (Password*)
- GroupId, the identity of the group that the user is assigned

- First name (*)
- Middle Initial (*)
- Last name (*)

Username: Valid user names are unique identifiers built out of A-Z, a-z, 0-9, _, -, and blank spaces, long up to 20 characters, that begin with a character or number. Username are not case sensitive.

Database alias info:

- Database name/alias (**Database Name***)
- Server/system where the database is installed
- Database user id (**Database User ID***)
- Database password (**Database Password***)

Database Name: Valid database name are unique identifiers built out of A-Z, a-z, 0-9, _, -, and blank spaces, long up to 20 characters, that begin with a character or number. Username are not case sensitive.

Database Password: The database passwords can be up to 20 characters, can contain any ASCII character and are not case sensitive

Database User ID: Valid user ID are unique identifiers built out of A-Z, a-z, 0-9, _, -, and blank spaces, long up to 20 characters, that begin with a character or number. User ID are not case sensitive.

System (Abstract State)

Administrator_Logged: (true or false). There can be more users with administrator power but just one logged at the same time.

Groups: “all the existing groups”, a mapping from GroupId (not further specified) into Group Info*. It always contains at least two groups, named “Administrator” and “Unknown”

Logged Users: (the set of the identities of the users UserId currently logged)

Upgraded User: of type UserId it may be undefined (the user that is now currently logged as administrator)

Jobs: “all the existing jobs” characterized by Job_Info*

Open Jobs: the identities of the jobs currently open (with opening data/time and user that has opened the job)

Open In Editor: of type JobId it may be undefined, the identity of the job open in the current editor

Security_Enabled: (true or false)

Users: “all existing users” a mapping from UserId (not further specified) into User_Info*.

Users always contains at least the users “Unknown User” and “Administrator”, and cannot contain two users with the same name.

Database Aliases: list of database aliases

Working Database Connection: (true or false) based on Database alias info*

Delete permission:

1. With Security_Enabled* false all users have “Delete Job” permission
2. With Security_Enabled* true, certain users will have “Delete Job” permission. Those users can delete jobs:
 - Created by themselves
 - Created by default users such as Margaret, Ralph, etc.
 - Created by “Unknown User”