Overall user interface

Team C: Julian, Gerben, Tamar, Joanna, Levin

October 13, 2009

1. Login screen
   - Use cases
2. User management screens
   - Use cases
3. Versioning screen
   - Use cases
4. Overview pieces screen
   - Use cases
5. Discussion screen

Screen: Login page

Summary

Allows user to login using username and password.

UI Components

- Textfields for
  - username
  - password
- Button *Login* (use case 1)
- Button *Lost password* (use case 2)

Interactions

- Login using credentials.

Screen: Recover password page

Summary

Allows user to request a new password (use case 2).

UI Components

- Textfields for
  - username
  - password
  - email address
- Button *Recover Password*

Interactions

- Request new password.

Screen: Set (new) password page

Summary

Allows user to set their password (use case 2). Only accessible from email requested through *Recover password page*.

UI Components

- Textfields for
  - new password
  - (retype) new password
- Button *OK*

Interactions

- Set a new password.

Use case 1: Login user

Summary	Use login before usage of the system
Actors	Users, system
Triggers	User wants to use the system.
Flow of events	1. User goes to *login page*. 2. User fills in *user name* and *password* fields 3. When user authentication is correct, send to *overview page*.
Exceptional flow	2 a Combination of *user name* and *password* is incorrect 3 a Show error and let user stay on the *login page*.
Postconditions	If no exception ⇒ user is authenticated

Use case 2: Recover password

Summary	User wants to recreate a password
Actors	Users, system
Triggers	User cannot (re)login, lost password.
Flow of events	1. User goes to *login page* 2. User clicks *Lost password* button 3. User is redirected to *Recover password* page. 4. User has to fill in *User name* and *e-mail address* fields. 5. User gets and e-mail with a link to *Set (new) password* page, to reset the password. 6. User click link 7. User give new password and clicks on the *Ok* button. 8. User send back to *login page*

Use case 2': Recover password

Exceptional flow	4 a E-mail address won't match with user. 5 a No e-mail is received.

User management

This is the component that handles the management of the users including authentication and authorization.

There are two screens in this component:

1. User overview: list an overview of all the users
2. User details: details for viewing, editing or adding an user.

Screen: User overview

Summary

This screen allows you:

- View current users in the system
- Select an user for editing purposes or view more details
- Create a new user.

UI Components

- Datagrid with all users listed down.
- Buttons: *Change user, Select button for every user*

Interactions

- All users can be selected, but only one. Selecting a new user deselectes the other user (radiobutton behavior)
- Users cannot be removed, only disabled.

Screen: User details

Summary

This screen allows you:

- Create a new user (Use case 1).
- Or view details of an user or change properties (Use case 2).

UI Components

- Input box for every detail of an user.
- Buttons: *Save*, *Cancel*

Interactions

- The values in the input boxes can be changed. Those are saved with the *Save* button.
- Cancel exits the current screen without making any changes.

Use case 3: Create user

Summary	Create a new user
Actors	Administrator, system, new user
Preconditions	Administrator is authenticated and authorized to create users.
Triggers	A new user wants to use the system
Flow of events	1. Administrator goes *user overview* page. 2. Administrator clicks *Create user* button 3. Administrator gives details of the user. 4. Administrator clicks *Save* button. 5. User gets and e-mail with a link to create a password. 6. User click link 7. User give password and clicks on the *Ok* button.

Use case 4: Update/disable user

Summary	Update some properties of a user or disable the user.
Actors	Administrator, system
Preconditions	Administrator is authenticated and authorized to edit users.
Triggers	User needs more or less rights, user not needed anymore.
Flow of events	1. Administrator goes *user overview* page. 2. Administrator selects user. 3. Administrator clicks *change selected user* 4. Administrator changes user or disables the user. 5. Administrator clicks *Save* button.

Screen Versioning (1)

Summary

This screen allows you to among others

- Branch: *When there's a legitimate reason, a specified version can be split up and continued as separate versions.*
- Merge: *When possible, parts of one branch can be merged back into another branch if this does not cause any conflicts.*
- View History: *View the entire version history of a versioned item/branch.*
- Revert: *Rollback a versioned item to an earlier version.*
- Add Comments: *Attach a comment to a versioned item.*

Screen Versioning (2)

UI Components

- Buttons: *View History, Add Comment, Merge To, Branch, Revert, Cancel*

Interactions

- View History, Add Comment take you directly to a new page
- Merge To, Branch and Revert will give you a confirmation screen before changes are carried out.
- Cancel exits the current screen without making any changes.

Versioning system / Piece management

- History of a fragments & pieces
- Search fragments & pieces
- Branching (and add comment)
- Add piece to system

Use case 5: View fragment history

Summary	The user wants to view the history of a versioned object.
Actors	User, system
Preconditions	User is authenticated.
Triggers	User presses the *View History* button from either the piece/fragment management screen or branch information screen.
Flow of events	1. The system generates the history view including comments for each version, version number and commit author.
Postconditions	The user is presented with the history overview for the selected versioned object.

Use case 6: Branching

Summary	The user wants to create a new branch using the current branch as the basis/template to fork.
Actors	User, system
Preconditions	User is authenticated and authorized to create branches.
Triggers	User pressed the *Create New Branch* button from the branch information screen.
Flow of events	1. The user is asked to confirm the request to branch 2. The user presses confirm 3. The system creates the new branch 4. The system navigates to the branch information page 5. The user adds people who can access the branch

Use case 6': Branching

Exceptional flow	2 a The user presses cancel and is returned to the previous view.
Postconditions	A new branch is created from the selected parent and has users associated with it.

Use case 7: Add Comments

Summary	The user wants to add a comment to a particular piece or fragment.
Actors	User, system
Preconditions	User is authenticated.
Triggers	User has pressed the *Add Comment* button on the piece/fragment information screen.
Flow of events	1. The system checks if the user is allowed to post comments on the current item 2. The user enters his/her comment into the comment box 3. The user presses the "add" button 4. The system checks the comment for vulgarity and/or other restricted words.

Use case 7': Add Comments

Exceptional flow	1 a The user is not allowed to post and is given an error. 4 a The check fails and the user is presented with an error.
Postconditions	The comment is added to the object in at the current version of the object.

Overview pieces (1)

Summary

This screen allows you to among others

- Move Pieces: *Pieces of the puzzle can be dragged on an interactive field.*
- Rotate Pieces: *Pieces of the puzzle can be rotated clock- and counterclockwise.*
- Zoom In/Out: *Zooming can be used to look at the puzzle at different scales.*
- Search Neighbours: *Possible neighbours can be searched for with an analysis system.*
- Add Pieces: *New pieces can be added to the puzzle by scientists.*

Overview pieces (2)

UI Components

- Interactive puzzle field with pieces.
- Buttons: *Zoom In, Zoom Out, Rotate Clockwise, Rotate Counterclockwise, Search Neighbours, Add Piece, View History, Add Comment*

Interactions

- Pieces will move in the puzzle field when they are dragged.
- Zoom In, Zoom Out, Rotate Clockwise and Rotate Counterclockwise have direct influence on the puzzle field.
- Search Neighbours will result in a list of possible neigbours which will be displayed in the frame.
- Add piece button creates new frame where you can give in parameters and the image of a new piece.
- View History and Add Comment trigger other use cases.

Use case 8: Search possible neighbour piece

Summary	User searches a possible neigbour piece.
Actors	Users, system
Preconditions	User is authenticated.
Triggers	User wants to search for a possible neighbour of a certain piece.
Flow of events	1. User selects a piece. 2. User presses the Search Neighbours button. 3. A list of possible neighbours is presented.
Postconditions	A list of possible neighbours is showed.

Use case 9: Add piece

Summary	A piece is added to the puzzle.
Actors	Users, system
Preconditions	User is authenticated.
Triggers	User wants to add a piece to the puzzle.
Flow of events	1. User presses the Add Piece button. 2. The system checks if the user is allowed to add a piece to the puzzle. 3. User enters parameters and link to image. 4. User presses Ok button to confirm the adding of the image. 5. The system adds the new piece to the puzzle.

Use case 9': Add piece

Exceptional flow	2 a The user isn't allowed to add a piece, and is given an error. 4 a User presses Cancel button and the piece is not added.
Postconditions	If Ok button is pressed ⇒ piece is added to the puzzle.

Communication

- Add discussion comment to fragment (share ideas)
- Create discussion (static)
- Post comment to discussion.

Discussion topic browser screen

- Summary: The list of all topics created is presented also a button enabling creation of new topic is present.
- Buttons: Create new topic
- Links: To detail page for each topic
- Functionality: Adding new topic, viewing the content of exising topics

Discussion content screen

- Summary: The name of the topic with the content is displayed. Also information about the author and posting date is visible. Below is the list of all comments posted and a button that enables posing new comment by the user
- Buttons: Add new comment, Go back
- Links: To topic browser screen
- Functionality: Adding new comment

Use case 10: Create discussion

Summary	User creates new discussiojn topic
Actors	User, system
Preconditions	User is authenticated
Triggers	User wants to create new discussion topic
Flow of events	1. User clicks *new discussion* button 2. User fills in *topic name* and *content* fields 3. When button *save* is pressed user is redirected to the list of all topics and new topic appears on the list of topics
Postconditions	-

Use case 11: Post comment

Summary	User adds new comment to existing topic
Actors	User, system
Preconditions	User is authenticated
Triggers	User wants to add a comment to existing topic
Flow of events	1. User selects the topic link 2. The content of discussion is displayed 3. User fills in the new comment 4. User presses *save* and the new comment appears
Postconditions	-