

Participants

213586 - Samuel Latella
213634 - Antonio Sisinni
213533 - Francesco Copelli
210438 - Stefan Yoshovski



ANALYSIS OF USER REQUIREMENTS	5
Purpose of the product	5
Product functionality	5
Figure 1.2	6
SYSTEM REQUIREMENTS ANALYSIS	8
Use Case Diagram	8
almost d'use	9
Use case UC1: Manage User	9
Use case UC2: Manage Component	13
UC3 Use Case: Interact with Posts	16
UC4 Use Case: Manage Posts	18
UC5 Use Case: Manage Configuration	21
Domain model	24
System Sequence Diagrams (SSD)	25
SSD - UC1	25
Alternate flow Edit User	25
Figure 1.3	25
Basic flow User ban	26
Figure 1.4	26
Alternate flow Delete User	27
Figure 1.5	27
SSD - UC2	28
Alternate flow Add Component	28
Figure 1.6	28
Basic Flow Remove Component	29
Figure 1.7	29
Alternate flow Edit Components	30
Figure 1.8	30
SSD - UC3	31
Basic Flow Likes	31
Figure 1.9	31
Alternate Flow Respond to Survey	32
Figure 2.1	32
SSD - UC4	33
Basic Flow Create Post	33
Figure 2.2	33
Alternate stream Edit Post	34
Figure 2.3	34



Alternate Flow Delete Post	35
Figure 2.4	35
SSD - UC5	36
Basic Flow Create Configuration	36
Figure 2.5	36
Alternate Flow Remove Configuration	37
Figure 2.6	37
Alternate Flow Change Configuration	38
Figure 2.7	38
Operations Contracts	39
User Operations Contracts	39
CO Delete user	39
CO Select User to Remove	39
CO User Selection to Ban	39
CO Ban user	39
CO Confirm Change User	39
CO Modify user	40
Contracts Operations Components	40
CO Select Component To Remove	40
CO Remove Components	40
CO Confirm Change	40
CO Modification Component	41
c.specifications has become specifications	41
CO Enter Component data	41
CO Add Component	41
Contracts Transactions Interactions Post	41
CO Enter "Like"	41
CO Remove "Like"	42
CO Check if the post has a "Like" associated with it	42
CO Check if the post has an associated "Poll Vote"	42
CO Insert "Poll Voting"	42
Post Operations Contracts	43
WHAT Elimina Post	43
CO Modify Post	43
CO Publish Post	43
CO Post modification request	43
CO Request for publication Post	44
Contracts Operations Configurations	45
CO Modify Configuration	45
CO Create Configuration	45



CO Request Delete Configuration	45
CO Configuration Change Request	45
CO Create Configuration	46
CO Add Component	46
CO Remove Component	46
CO Execute Operation	46
CO Control Addition	46
Logical Architecture	47



ANALYSIS OF USER REQUIREMENTS

1. Purpose of the product

AND required the creation of a website, c.d. system, which hosts different types of users. The system presented is designed to facilitate the user in creating computer configurations with components chosen from a wide range and purchasing them thanks to a connection with a third-party e-commerce platform.

Furthermore, the system is designed to create a community in which users can share content by interacting with them.

The system must be able to provide suitable means for entering and managing the data relating to the components and users that compose it.

Furthermore, a section must be set up within the structure to help less expert users, which offers a selection of pre-made configurations divided by price ranges.

2. Product functionality

The system must allow a user to authenticate or register within it, because according to his role he will have access to various functions:

1. User (unauthenticated), c.d. guest, unauthenticated user of the system
2. User (authenticated), user of the system which is divided into:
 - a. Newbie
 - b. Nerd
 - c. Genius
3. Manager, site manager
4. Site administrator, capable of creating, viewing, modifying, removing operations.

The operations that can be carried out are listed in the [Figure 1.2](#).

The system must allow for banning a user that is, limiting access to the functions of it for a certain period of time indicated, moreover it must also be allowed to cancel this status upon request.

The system must allow authenticated users to publish content, the so-called post. Such posts they will have to contain a title and description, in addition to it they can (optionally) sections be added:

1. a picture;
2. a survey in which you can express a preference between two options.

The system must allow users to interact with posts, the types of interaction that can be carried out are:

1. inserting or removing a like;
2. apply a choice to a survey.

The system must make it possible to store the data and specifications of various hardware components, arranging interfaces that allow some users to "suggest" new components which will then have to be reviewed by the site managers

With the components present in the system it must be possible to create "Configurations", i.e. combinations of components to create PCs. These configurations must also be able to be shared with other users within the site or outside it through public links that do not require authentication on the site.



Functionality related to	Operation	User (unauthenticated)	User (authenticated)			Controller	
		Guest (1)	Newbie (2a)	Nerd (2b)	Genius (2c)	Manager (5)	Admin (6)
User	Registration	✓	✗			✗	
	View/Edit	✗	! Only themselves			! Only themselves or inferior	✓
	Ban/Delete	✗	✗			! only themselves or inferior	✓
Component	Creates Approve Edit Remove		✗			✓	
Post	Creates View Interact	✗	✓			✓	
	Edit Remove	✗	! Only themselves			! Only themselves	
Configuration	Creates View	✓	✓			✓	
	Publish Edit Remove	✗	✓			✓	

Figure 1.2



The image displays three devices showing the Silicon Square website. The desktop screen at the top right features a large banner for the 'Configurator' with a blue and black aesthetic. Below it are three cards: 'Configurator' (with a circular icon), 'Templates' (with a circular icon), and 'Square' (with a person icon). The tablet screen on the left shows a sidebar with 'Component Categories' and a grid of component images. The smartphone screen at the bottom left shows a configurator interface with a search bar and a list of components.

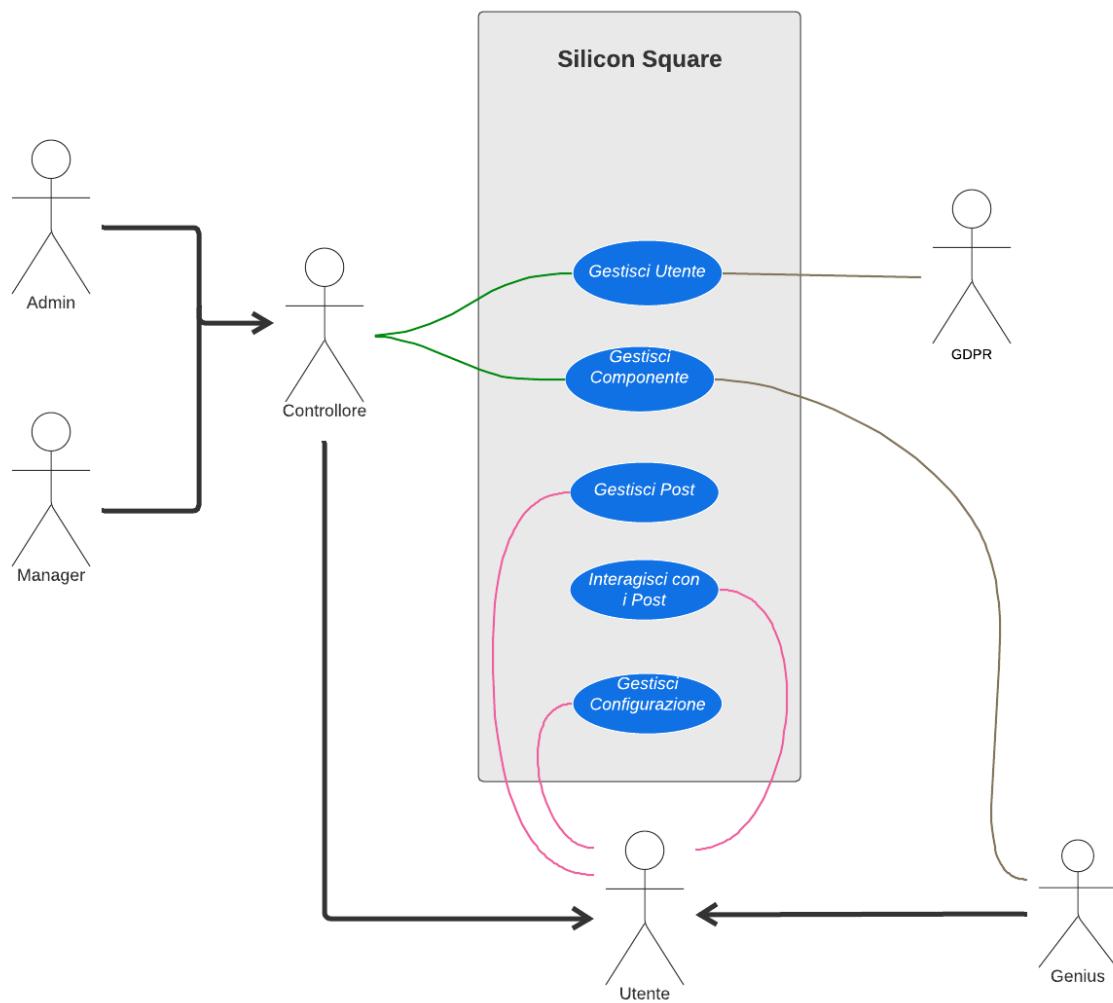
Interface that allows you to adapt to the size of the device on which you are using.

The image shows four devices illustrating the platform's responsive design across different screen sizes. The laptop on the left displays a user profile page for 'Stefko Yoshovski'. The tablet in the center shows a blog post titled 'I am Legend' with a photo of a man and his dog. The smartphone at the bottom left shows a product detail page for an 'Intel Core i9-12900HK' processor. The desktop monitor on the right shows a 'Components' page with a table listing various hardware items like CPU Coolers, Fans, and Memory modules, each with edit and delete icons.



SYSTEM REQUIREMENTS ANALYSIS

1. Use Case Diagram





2. Casi d'Uso

2.1. Use case UC1: Manage User

Scope: Silicon Square application

Level: User goal

Primary actor: Ticket inspector

Stakeholders and Interests:

- Controller: Wants to manage all user data modification, deletion and moderation operations. Also, he wants to have full control over the grades of all registered users.
- Administrator (Admin): He wants to keep the system always operational and functional.
- Manager: Wants to make sure that the System is always up and running.
- User: Wants to browse the site without interruptions and in the simplest way possible.
- GDPR: Wants personal data processing regulations to be complied with.

Pre-conditions: The Controller is identified and authenticated within the System.

Success Guarantee (or Post-Conditions):

The management operation is performed. All data connected to this operation are updated in compliance with the regulations on the processing of personal data.

Main Success Scenario (or Base Flow):

1. The controller asks the system to ban users.
2. The system returns the list of users.
3. The Controller is looking for a user.
4. The System proposes users correlated to the Controller's request.
5. The Controller indicates to the System a user to ban.
6. The system shows the list of users hitherto affected by the ban operation.
The Controller resumes at step 3 until it indicates that it is finished.
7. The Controller indicates the ban period for the affected users and asks the System to ban them.
8. The system applies the operation of ban.
9. The system generates a response.
10. The System displays the updated information of the users.

Extensions (or Alternative Streams):

*to. At any moment, the System fails:

1. The Controller restarts the system.
 - 1a. the Controller is not an administrator.
 1. The Controller contacts the administrator to ask him to restore system operation.
1. The system restarts from its last consistent state.
2. Go back to step 1 of the basic flow.



*b. At any time, the controller cancels the operation

1. The system discards the controller changes.
2. Go back to Point 9.

1a. The controller wants to delete users.

1. The controller asks the System to delete users.

1b. Controller wants to edit users.

1. The controller asks the system to modify some users.

4a. The system does not find users related to the Controller's request

1. The System notifies the Controller.
2. Go back to step 3 of the basic flow.

5a. The Controller wants to delete a user

1. The Controller selects a user to delete.

5b. The Controller wants to edit a user

1. The Controller selects a user to edit.
2. The system shows information relating to the indicated user.
3. The Controller indicates to the System the data to be modified
4. Go to step 8 of the basic flow.

5-7a. The Controller is no longer interested in banning one of the users.

1. The Controller indicates to the System the user who is no longer affected by the ban.

1a. No users are currently affected by the ban.

1. The System notifies the Controller.
2. The System asks the Controller if he wants to add new users.
3. The controller indicates to the system that he wants to select new users.
4. Go back to step 3 of the basic flow.

2. The system removes the user from the list of those affected by the ban operation

The Controller repeats from point 1 of flow 5-7a as long as there are still users in the list of those interested in the ban who should not

5-7b. The Controller is no longer interested in deleting one of the users.

1. The Controller indicates to the System the user who is no longer affected by the cancellation operation.

1a. No user is currently affected by the delete operation.

1. The System notifies the Controller of the error.



-
- 2. The System asks the Controller if he wants to add new users.
 - 3. The controller indicates to the system that he wants to select new users.
 - 4. Go back to step 3 of the basic flow.
- 2. The system removes the user from the list of those involved in the deletion operation.

The Controller repeats from point 1 of flow 5-7a as long as there are still users in the list of those interested in deletion who should not

- 6a. The controller instructed the System to delete a user.
 - 1. The system shows the list of users hitherto affected by the deletion operation.
- 7a. The Controller wants to ask the System to modify the user concerned
 - 1. The Controller asks the System to apply the modification operation of the user involved.
 - 2. Go to step 9 of the basic flow
- 7b. The Controller wants to ask the System to delete the affected users
 - 1. The Controller asks the System to apply the operation of deleting the users involved.
- 8a. The system must apply the Delete operation as requested by the controller:
 - 1. The System applies the Elimination operation.
- 8b. The system must apply the Modify operation as requested by the controller:
 - 1. The system applies the Modify operation.
 - 1a. The System detects that the role of the Controller is lower than or equal to that of the User who is trying to modify:
 - 1. The System notifies the Controller.
 - 2. The system asks the user if he wants to continue with the operation by removing the users he is not responsible for.
 - 3. The Controller confirms that he wants to continue.
 - 4. The System discards users over which it has no competence.
- 8c. One of the users has already been deleted by another controller:
 - 1. The system detects that some users have already been deleted.
 - 2. The system discards users who are no longer present
 - 2a. There is no user
 - 1. The system cancels the delete operation.
 - 2. Go to step 9 of the basic flow



-
3. The system applies the deletion of the remaining users.
 - 8d. The ban operation has already been performed by another controller:
 1. The system recognizes that some users have already been banned by another controller.
 2. The system recognizes that the new period ofban is greater than the previous one.
 - 2a. The system recognizes that the new period ofban is less than or equal to the previous one.
 1. Go to step 9 of the basic flow
 3. The System extends the period ofban of the user until the new date..
 - 8e. Among the users selected by the Controller for the ban operation, there are some with a rank equal to or higher than his:
 1. The system interrupts the ban operation.
 2. The System notifies the Controller
 3. go back to step 3 of the basic flow.
 - 8f. Among the users selected by the Controller for the modification operation, there are some with a rank equal to or higher than his:
 1. The system notifies the controller of the error.
 2. System cancels modification.
 3. go back to step 3 of the basic flow.
 - 8g. Among the users selected by the Controller for the deletion operation, there are some with a rank equal to or higher than his:
 1. The system notifies the controller of the error.
 2. The system cancels the deletion.
 3. Go back to step 3 of the basic flow.

Special Requirements:

- Interface designed with technologies that allow it to adapt to the size of the device on which the system is being used, allowing it to maintain its usability.

List of technology variants and data: Nobody

Repetition rate: Could be almost uninterrupted.

Open problems:

- How do the laws on the processing of personal data vary?
- Look into the question of the data wipe operation.
- Management of dual access of the same controller from multiple devices.

2.2. Use case UC2: Manage Component

Scope: Website (Silicon Square)



Level:User goal

Primary actor:Ticket inspector

Stakeholders and Interests:

- Controller: Wants to access a component at any time and make changes. He also wants to be sure that the data entered is correct and that the data is up to date.
- Administrator (Admin): He wants to keep the system always operational and functional.
- Contributor (Genius): Wants to increase its visibility within the System by adding updated components.

Pre-conditions: The Controller is identified and authenticated within the System.

Success Guarantee (or Post-Conditions): Approval operation is performed. All data connected to this operation is updated.

Main Success Scenario (or Base Flow):

1. The Controller asks the System to remove some components
2. The system returns the list of components.
3. The Controller is looking for a component.
4. The System proposes components correlated to the request of the Controller.
5. The Controller indicates the component on which to perform the elimination operation.
6. The system shows the list of components hitherto affected by the deletion operation.

The Controller resumes at step 3 until it indicates that it is finished.

7. The Controller asks the System to remove the affected components.
8. The system requires confirmation.
9. The Controller confirms the deletion operation.
10. The System deletes the components affected by the deletion.
11. The system generates a response.
12. The System displays the updated information of the components.

Extensions (or Alternative Streams):

*to. At any moment, the System fails:

1. The Controller restarts the System.
 - 1a. The Controller is not an administrator.
 1. The Controller contacts the administrator to ask him to restore system operation.
 1. The system restarts from its last consistent state.
 2. Go back to step 1 of the basic flow.

*b. At any time the controller wants to cancel the edit/delete operation:

1. The Controller indicates to the System that it wants to interrupt the operation.
2. The system discards all information.
 - 2a. The system has no information about the operation



-
1. Go to step 13 of the basic flow.
 3. Go to step 13 of the basic flow.
- 1a The controller wants to add a component:
1. The controller asks the system to add a component.
 2. Go to step 5 of the basic flow.
- 1b The controller wants to modify a component:
1. The controller asks the system to modify a component.
- 2a. The system returns an empty list of components:
1. The system notifies the controller that the list is empty.
 2. The use case ends.
- 4a. The system does not find components related to the request of the controller:
1. The System notifies the controller that there are no components related to the search
 2. Go back to step 3 of the basic flow.
- 3-7a. The Controller is no longer interested in deleting one of the components:
1. The Controller indicates to the System a component that is no longer affected by the delete operation.
 - 1a. No component is currently affected by the operation.
 1. The system notifies the user of the error.
 2. The System asks the Controller if he wants to insert new components.
 3. The controller indicates to the system that it wants to select new components.
 - 3a. The Controller does not want to select new components to perform operations on:
 1. The controller indicates to the system that it does not want to select new components.
 2. Go to step 12 of the basic flow.
 4. Go back to step 3 of the basic flow.

The Controller resumes at step 1 of flow 5-7a until it indicates it is finished.

- 5a The Controller wants to modify a component:
1. The Controller indicates a component on which to carry out the modification operation.
 2. The system shows the information relating to the component to be modified
 3. The controller tells the system which data it needs to change.
 4. The controller wants to approve the component
 - 4a. The controller wants to make the changes without approving the component
 1. The controller asks the system to make the changes without approving the component.



-
2. Go to step 10 of the basic flow.
 5. The Controller asks the System to modify the component, approving it.
 6. Go to step 10 of the basic flow.
- 5b The Controller wants to add a component:
1. The Controller indicates a category of components in which to carry out the addition operation.
 2. The system asks the controller to enter the information relating to the component.
 3. The Controller enters the information relating to the component to be added.
 4. The Controller asks the System to add the component.
 5. Go to step 10 of the basic flow.
- 7a. The Controller wants to ask the System to remove the affected components:
1. The Controller asks the System to apply the operation of eliminating the involved components.
- 9a. The Controller wants to confirm the Modify operation:
1. The Controller confirms the Modify operation.
 2. The system applies the Modify operation.
 3. Go to step 11 of the basic flow
- 9b. The Controller wants to confirm the Add operation:
4. The Controller confirms the Add operation.
 5. The system applies the Add operation.
 6. Go to step 11 of the basic flow
- 10a. The system detects that a component is not present because it has already been deleted from another controller:
1. The system warns the controller that one of the components has been deleted
 2. The system discards the components that are no longer present.
 - 3a. There is no component present
 1. The system cancels the approval operation.
 2. Go to step 11 of the basic flow.
 3. The system applies the approval operation.

Special Requirements:

- Interface designed with technologies that allow it to adapt to the size of the device on which the system is being used, allowing it to maintain its usability.

List of technology variants and data: Nobody

Repetition frequency: Could be almost uninterrupted.

Open problems:

- Management of dual access of the same controller from multiple devices.

2.3.UC3 Use Case: Interact with Posts

Scope:Silicon Square application



Level:User goal

Primary actor:User

Stakeholders and Interests:

- User: Wants to be able to interact with posts made by other users of the system. He wants to know the opinion of other users on the published content. He wants to know how many and which users have interacted with the post.
- Manager: Wants to make sure that the System is always up and running.

Pre-conditions: The user must be identified and authenticated within the system.

Success Guarantee (or Post-Conditions): The interaction is notified and the related data are updated and displayed correctly.

Main Success Scenario (or Base Flow):

1. The userhe asks to the system to show all posts
2. The system returns the list of posts.
3. The user indicates a post.
4. The system displays information about the post concerned.
5. The userhe asks the system to "Like" the post.
6. The system applies the "Like" operation.
7. The system shows the updated information of the post.

Extensions (or Alternative Streams):

*to. At any moment, the system fails:

1. The user waits for a solution to the problem.
 - 1a. The user is an administrator
 1. The user restarts the system
 2. The system restarts from its last consistent state.
 3. Go back to step 1 of the basic flow.
 - 1b. The user is a Manager
 1. The User contacts the administrator to ask him to restore the functioning of the System.
2. The user checks the status of the system.
 - 2a. The system still doesn't respond.
 1. Back to point *a.
 - 2a. The system does not return any posts.
 1. The system notifies the user of the lack of posts.
- 4a. The post is no longer available
 - a. The system notifies the user that the post is no longer available
 - b. The use case ends.
- 5a. The user instructs the system to respond to the "Survey" of the post:
 1. The user indicates the preference for the survey.
 - 1a. The user has already answered the survey
 1. The system does nothing.



-
- 2. The use case ends
 - 2. The system records your preference.
 - 3. Go to step 7 of the basic flow.
- 5b. The user has already “Liked” the post
- 1. The system removes the "Like" to the post
 - 2. Go to step 7 of the basic flow

Special Requirements:

- Interface designed with technologies that allow it to adapt to the size of the device on which the system is being used, allowing it to maintain its usability.

List of technology variants and data: Nobody

Repetition rate: Could be almost uninterrupted.



2.4.UC4 Use Case: Manage Posts

Scope:Silicon Square application

Level: User goal.

Primary actor: User.

Stakeholders and Interests:

- ↳ User: He wants to create, delete and edit his posts without losing data during the operations.
- ↳ Administrator (Admin): He wants to keep the system always operational and functional.
- ↳ Manager: Wants to make sure that the System is always up and running.
- ↳ System Policy (Policy): Indicates a guideline to be maintained within the system to avoid content that may offend users' sensitivity.

Pre-conditions: User and Administrator must be identified and authenticated. User must not be banned.

Success Guarantee (or Post-Conditions):The user successfully performs the desired operation on a post and all the data connected to it are saved respecting the system policy.

Main Success Scenario (or Base Flow):

1. The User requests the System to create a new post.
2. The System asks the User for basic information about the post and asks for confirmation.
3. The User enters the basic information on the post and confirms it.
4. The user tells the system that he wants to add a particular photo to the post.
5. The system adds the photo to the post.

The User continues from point 4 until he has finished inserting attachments

6. The User requests the System to publish the post.
7. The system publishes the post.
8. The system generates a response.
9. The System displays the updated list of User's posts.

Extensions (or Alternative Streams):

*to. At any moment, the system fails:

1. The user waits for a solution to the problem.
 - 1a. The user is an administrator
 1. The user restarts the system
 2. The system restarts from its last consistent state.
 3. Go back to step 1 of the basic flow.
 - 1b. The user is a Manager
 1. The User contacts the administrator to ask him to restore the functioning of the System.
 2. The user checks the status of the system.



2a. The system still doesn't respond.

1. Back to point *a.

*1-5a. At any time, the User can cancel the operation:

1. The userhe asks the system to cancel the operation.
2. The System discards all the information it had on the operation.
3. The system cancels the controller operation.
4. Go to step 8 of the basic flow.

1a. The User wants to ask the System to delete a post:

1. The User asks the System to delete a post.
2. The system displays the list of posts published by the user.
3. The User indicates the post to be deleted.
4. Go to step 8 of the basic flow.

1b. The User wants to ask the System to modify a post:

1. The User asks the System to edit a post.
2. The system displays the list of posts published by the user.
3. The User indicates the post to edit.

2b. The User is editing a post, the System will have to show the current information

1. The system shows the basic information already present for the post.

3b. The user is editing the basic information of the post

- 1a. The user changes the current basic information of the post
 1. The User does not want to change the basic information of the post.
 2. Go to step 4 of the basic flow.

4a. The User is editing the post, the System must ask the User if he wants to edit the photos/surveys/configurations of the post.

1. The system asks the user if he wants to edit photos/polls of the post.

5a. The user wants to add a poll to the post:

1. The user instructs the system to add a poll to the post.

5b. User had no intention of making edits to photos/polls of post.

1. The user indicates to the system that he does not want to make any changes to photos/surveys
2. Go to step 7 of the basic flow.

5c. The user wants to change the photo of the post:

1. The User indicates a photo to be included in the post.

5d. User wants to edit post poll:



-
1. The User indicates a survey to be included in the post.
 - 6a. The User instructed the System to add a survey to the post
 1. The system checks that there isn't already a poll in the post
 - 1a. There is already a poll in the post.
 1. The system asks the user if he wants to replace the poll in the post
 2. The user indicates to the system that he wants to replace the survey
 - 2a. The user has no intention of replacing the photo currently in the post:
 1. Go to step 4 of the basic flow.
 3. The system replaces the poll in the post.
 4. Go back to step 5 of the basic flow.
 2. The system adds the poll to the post
 - 6b. The User instructed the System to add a photo to the post when one already existed:
 1. The system has detected that there is already a photo in the post.
 2. The system asks the user if he wants to replace the photo in the post.
 3. The user tells the system that he wants to replace the photo.
 - 3a. The user has no intention of replacing the photo currently in the post:
 1. Go to step 4 of the basic flow.
 4. The system replaces the photo in the post.
 5. Go back to step 5 of the basic flow.
 - 7a. The User wants to ask the system to delete the post:
 1. The User requests the System to delete the post.
 - 8a. The user has asked the system to delete a post, the system must ask for confirmation for the deletion:
 1. The system shows the summary of the post you want to delete and asks for confirmation.
 - 9a. The user wants to indicate to the system that it is safe to carry out the cancellation:
 1. The user tells the system that he is sure he wants to delete the post.
 2. The system deletes the post.

Special Requirements:

- Interface designed with technologies that allow it to adapt to the size of the device on which the system is being used, allowing it to maintain its usability.

List of technology variants and data: Nobody



Repetition frequency: Very frequently

Open problems: Nobody

2.5. UC5 Use Case: Manage Configuration

Scope: Silicon Square application

Level: User goal.

Primary actor: User.

Stakeholders and Interests:

- User: Wants to be able to create a configuration with hardware components. He also wants to be able to create multiple different configurations that he can change later. He also wants to be able to share a configuration with other users.
- Manager: Wants to make sure that the System is always up and running.
- Administrator (Admin): He wants to keep the system always operational and functional.

Pre-conditions: None.

Success Guarantee (or Post-Conditions): The User creates a computer configuration.

Main Success Scenario (or Base Flow):

1. The User requests the System to create a new configuration.
2. The system asks for the type of component to add to the configuration.
3. The User indicates to the System the type of component to add to the configuration.
4. The system shows the list of components of the specified type
5. User searches for a component.
6. The System offers components related to the User's search.
7. The User indicates to the System a component from the list, to be added to the current configuration.
8. The system registers the component.
9. The system shows the summary of the configuration.
10. The system calculates the total cost of the components of the configuration.

The User continues from step 2 until he indicates that he is finished.

11. The User requests the System to record the configuration.
12. The system asks for confirmation.
13. The system registers the configuration.
14. The system generates the response.

Extensions (or Alternative Streams):

*to. At any moment, the System fails:

1. The User awaits a solution to the problem.
 - 1a. User is an administrator



-
- 1. The User restarts the System
 - 2. The system restarts from its last consistent state.
 - 3. Go back to step 1 of the basic flow.
- 1b. The User is a Manager
- 1. The User contacts the administrator to ask him to restore the functioning of the System.
2. The User checks the status of the System.
- 2a. The system still doesn't respond.
 - 1. Back to point *a.
- *b. At any time the User wants to cancel the addition operation:
- 1. The User indicates to the System that he wants to interrupt the add operation.
 - 2. The system discards all information.
 - 3. Go to step 15 of the basic flow.
- *c. At any time the User wants to cancel the modification operation:
- 1. The User indicates to the System that he wants to interrupt the modification operation.
 - 2. The system discards all information.
 - 3. Go to step 13 of the basic flow.
- *d. At any time the User wants to cancel the cancellation operation:
- 1. The User indicates to the System that he wants to interrupt the cancellation operation.
 - 2. The system discards all information.
 - 3. Go to step 13 of the basic flow.
- 1a. The User wants to ask the System to modify a previously created configuration:
- 1. The User requests the System to modify a previously created configuration.
 - 1. The system shows the indicated configuration.
- 1b. The User wants to ask the System to delete a previously created configuration:
- 1. The User requests the System to delete a previously created configuration.
 - 2. The system shows the list of all user configurations.
- 2a. User is not authenticated:
- 1. The System asks the User to identify himself.
 - 2. The User identifies himself.
- 2a. The User does not want to identify himself.
- 1. go to step 13 of the basic flow.
3. The system asks the user which configuration he wants to delete.
- 3a. User has never created a configuration:



-
1. Go to step 2 of flow 1b.
 4. The User indicates the configuration to be deleted.
 5. Go to step 10 of the basic flow.
- 2b. User has never created a configuration:
1. The System notifies the User that there are no configurations.
 2. The use case ends.
3. The system asks the user which configuration he wants to delete.
 4. The User indicates the configuration to be deleted.
- 2a. The User wants to remove a component of the configuration:
1. The user indicates to the system the component to remove from the configuration.
 2. The System removes the component from the current configuration.
 3. Go to step 9 of the basic flow.
- 2b. The user instructed the System to add a component type that has already been added and cannot be repeated:
1. The system does not allow the addition operation.
 2. Go to step 9 of the basic flow.
- 6a. The System does not find components related to the request from the Controller:
- a. The System notifies the User.
 - b. Go back to step 5 of the basic flow.
- 13a. The User has requested the System to register the configuration.
1. The User is identified.
 - 1a. The User is not identified:
 1. The System asks the User to identify himself.
 2. The User identifies himself.
 - 3a. The User does not identify himself:
 1. The System notifies the User.
 2. The system discards the changes.
 3. The use case ends.

Special Requirements:

- Interface designed with technologies that allow it to adapt to the size of the device on which the system is being used, allowing it to maintain its usability.

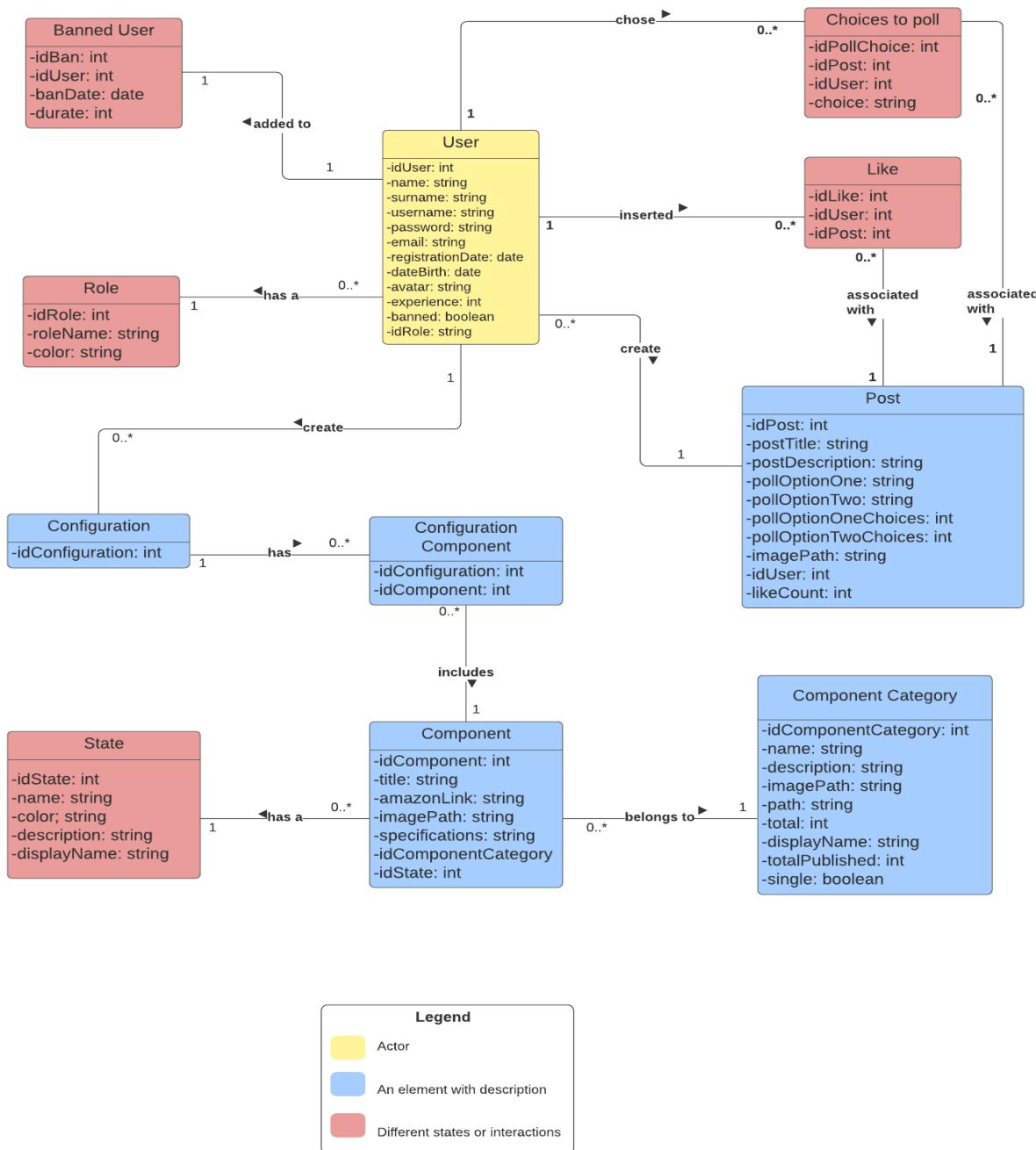
List of technology variants and data: Nobody

Repetition frequency: Very frequently.

Open problems: Nobody



3. Mdomain model





4. DSystem Sequence Diagrams (SSD)

4.1. SSD - UC1

a. Alternative stream Edit User

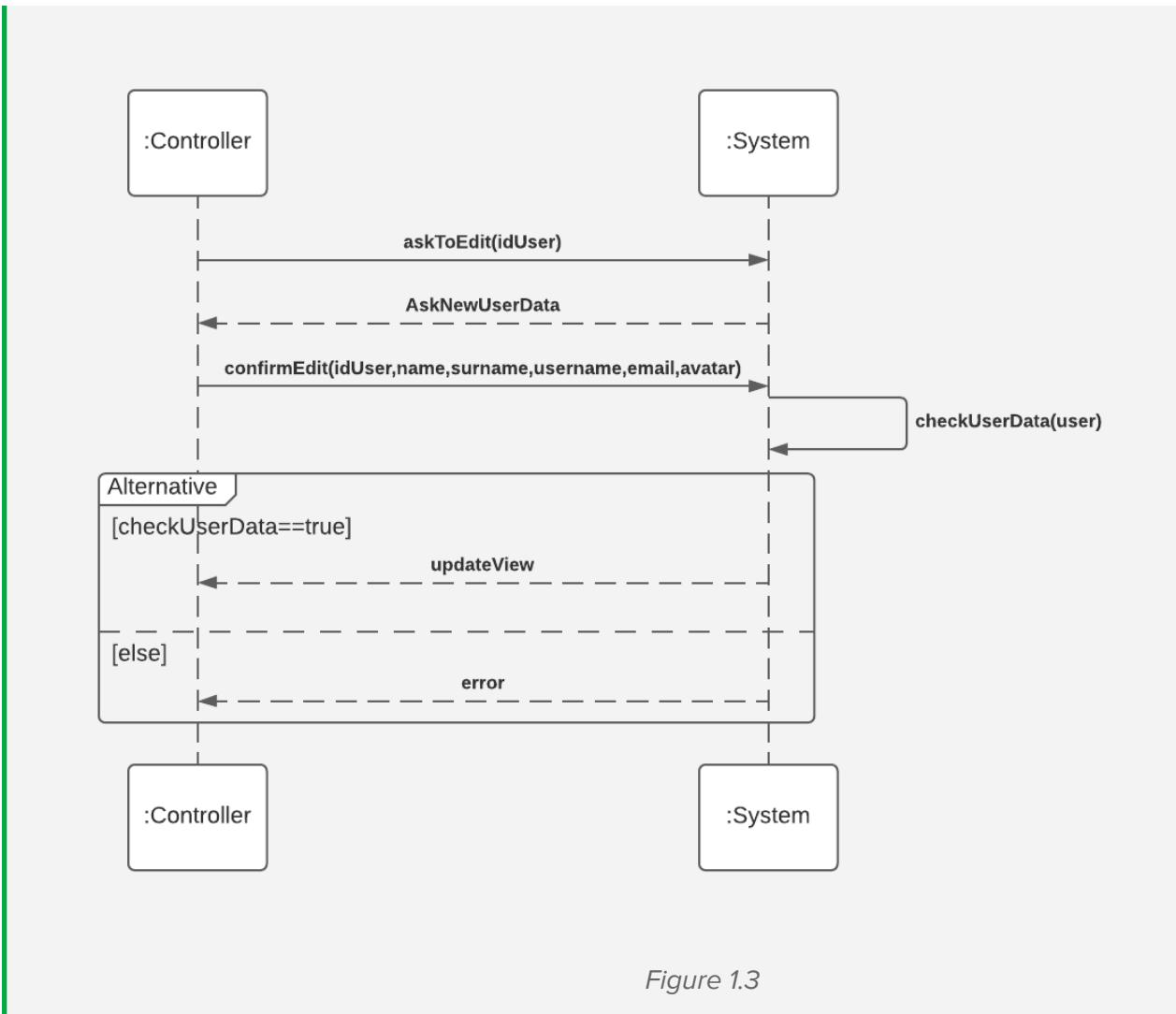


Figure 1.3



b. Fbasic luxury Banna Utente

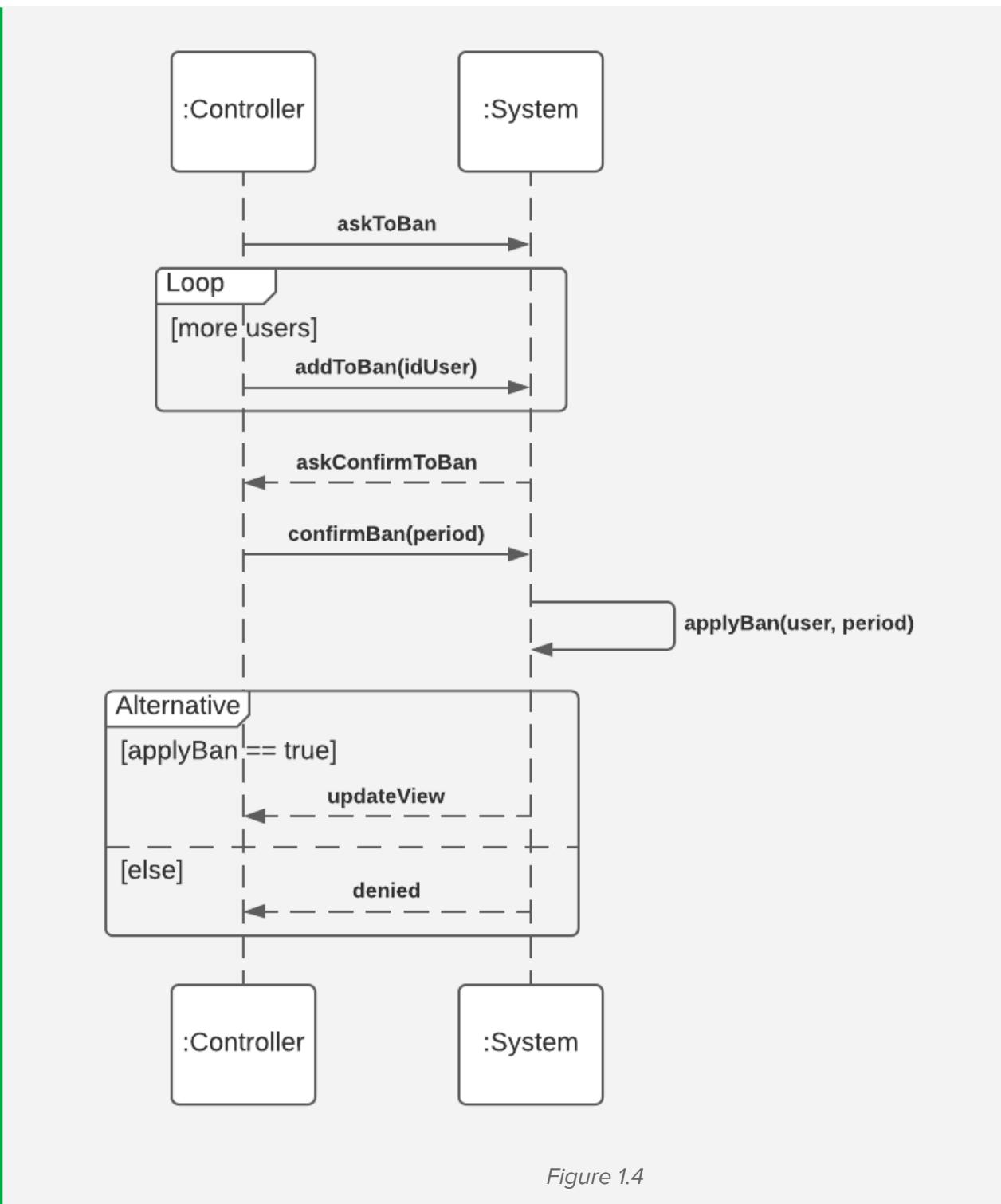


Figure 1.4



c. Alternative streamDelete User

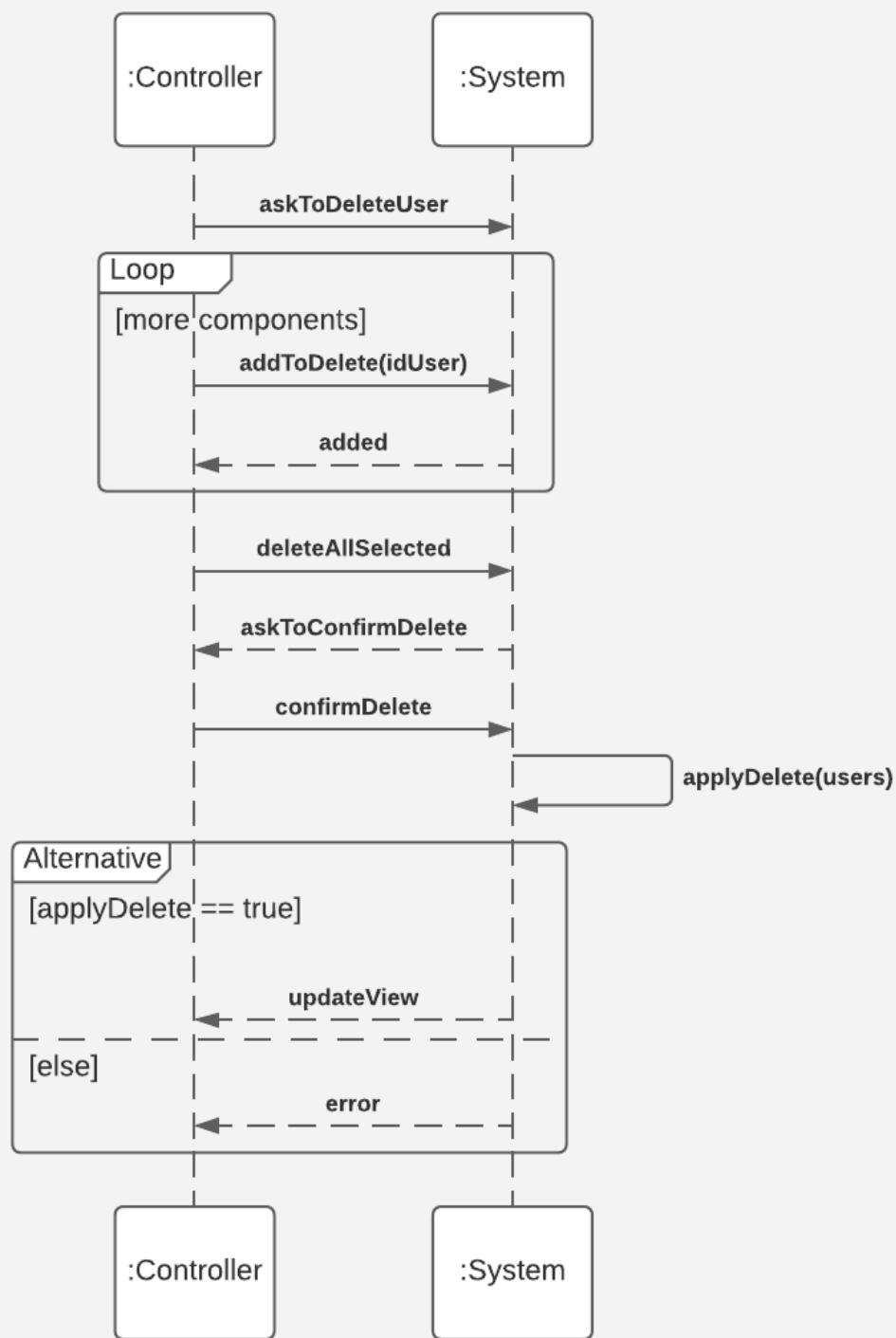
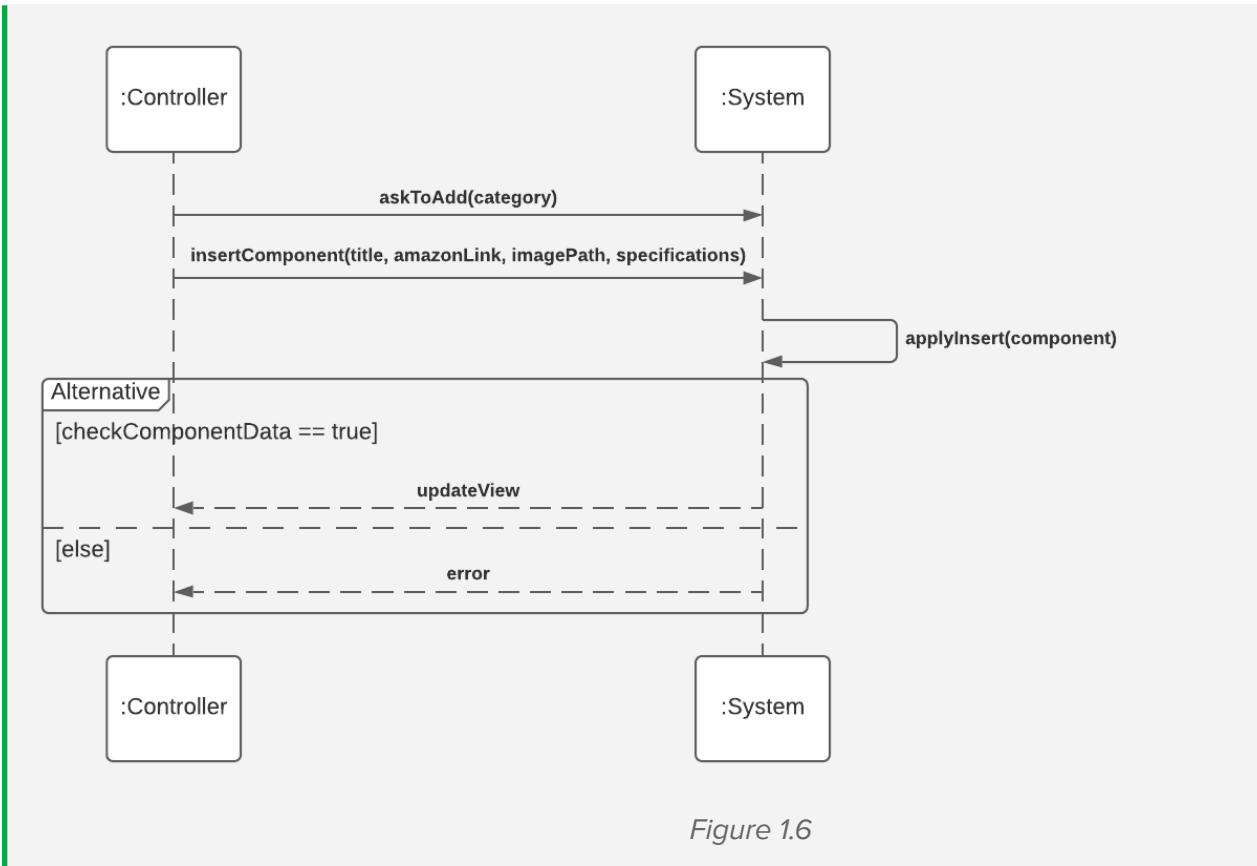


Figure 1.5



4.2. SSD - UC2

a. Alternative stream Add Component





b. Basic flow Delete Component

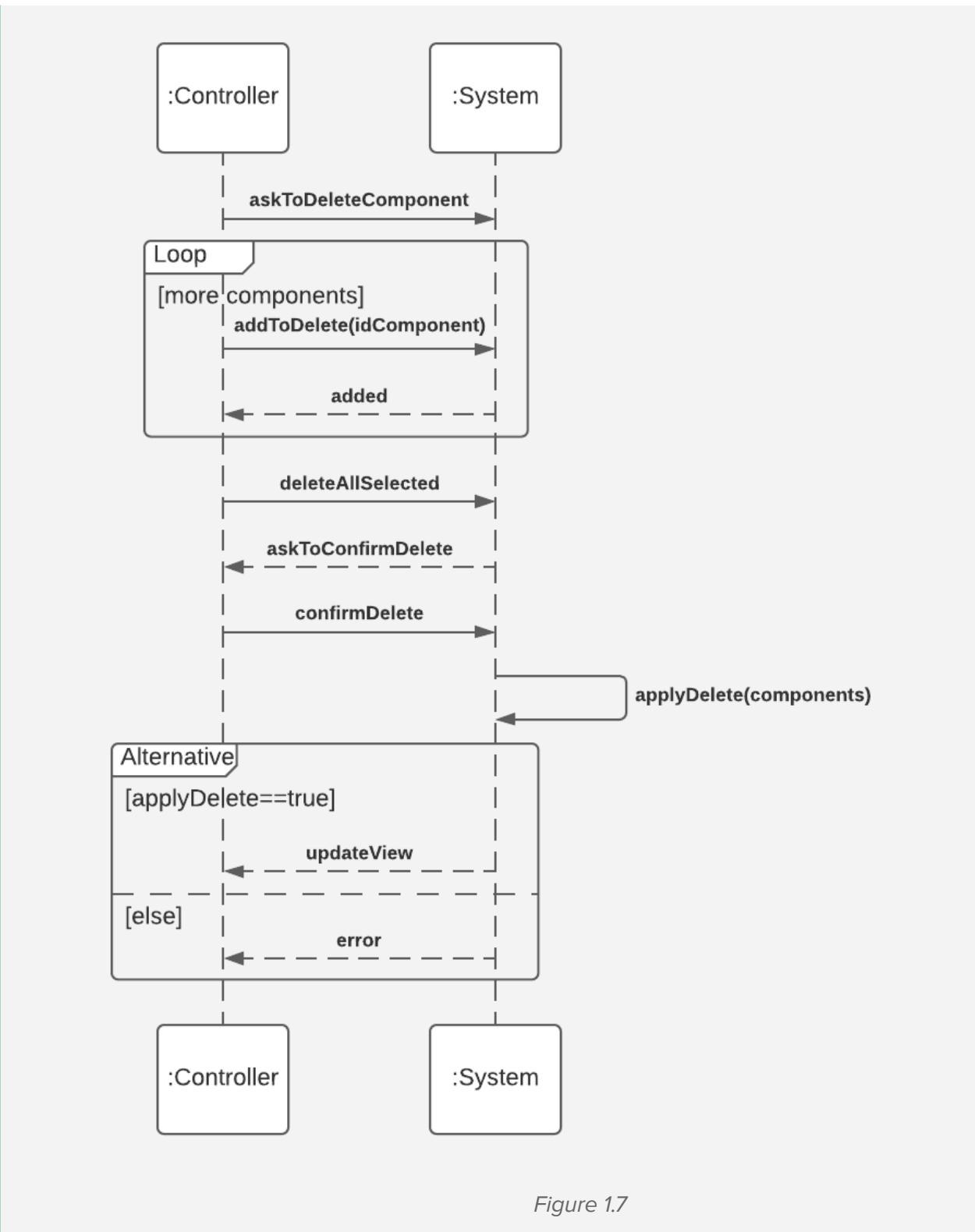


Figure 1.7



c. Alternative stream Modify Components

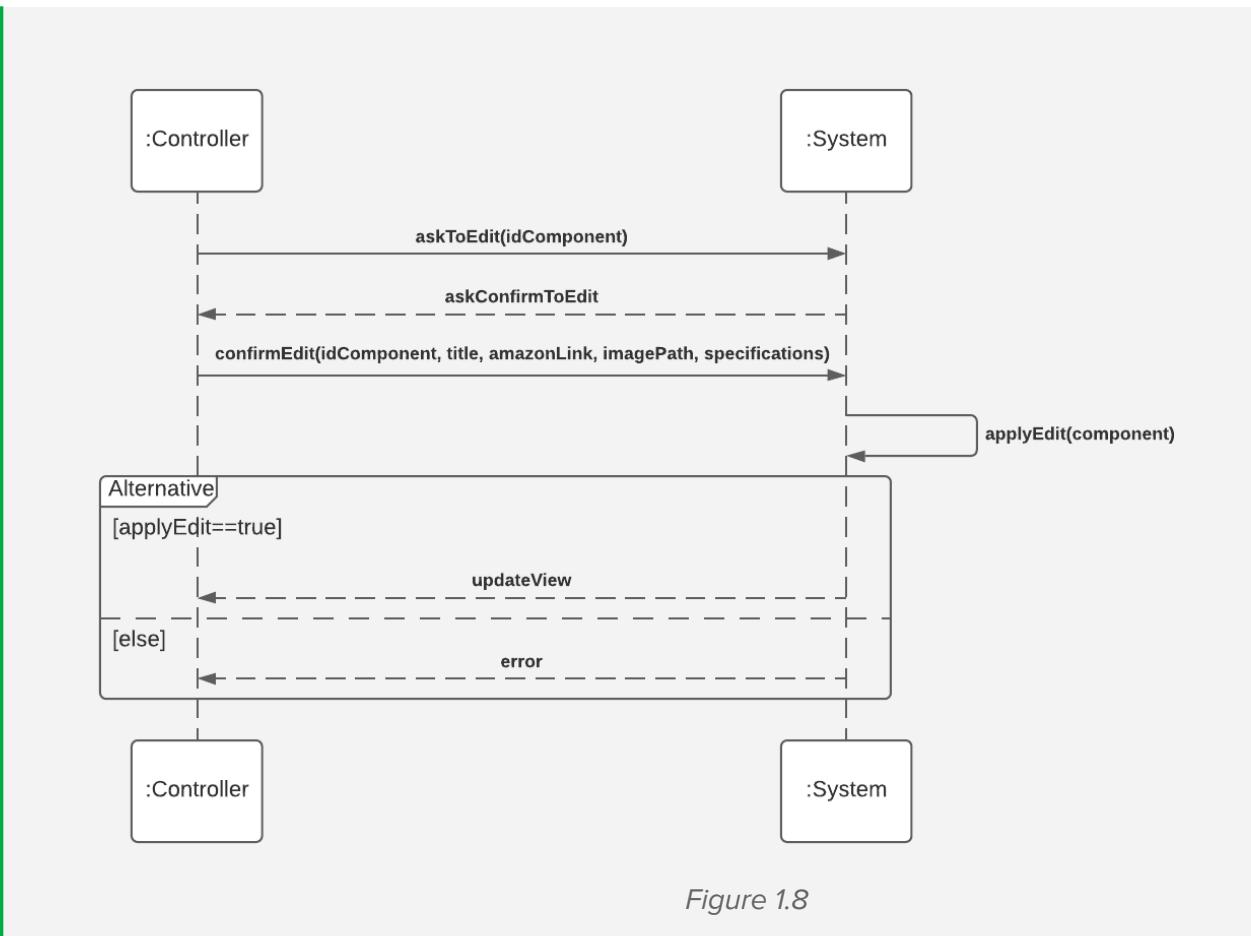


Figure 1.8



4.3. SSD - UC3

a. Basic flow *I like*

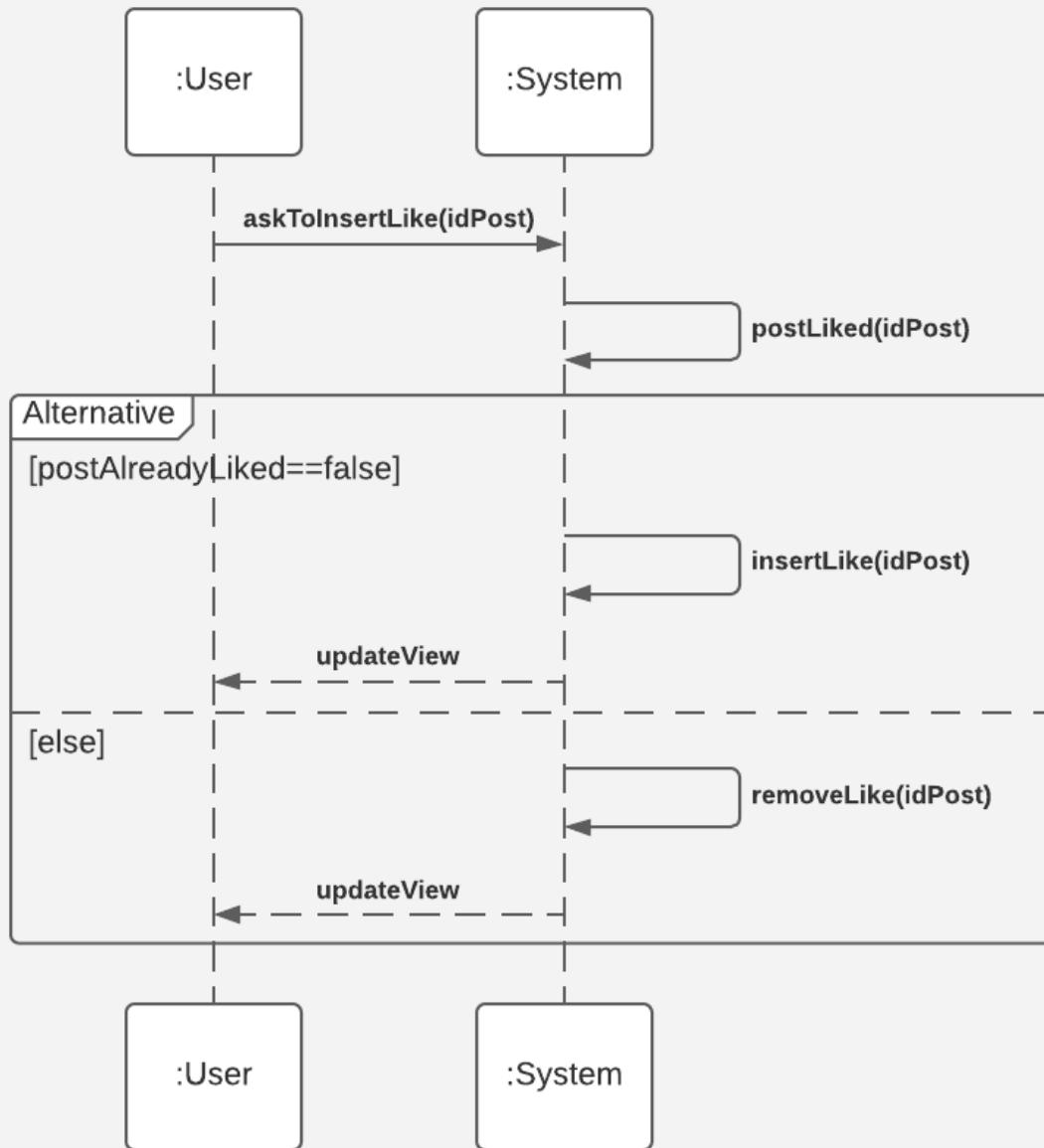


Figure 1.9



b. Alternative stream [Reply to the Survey](#)

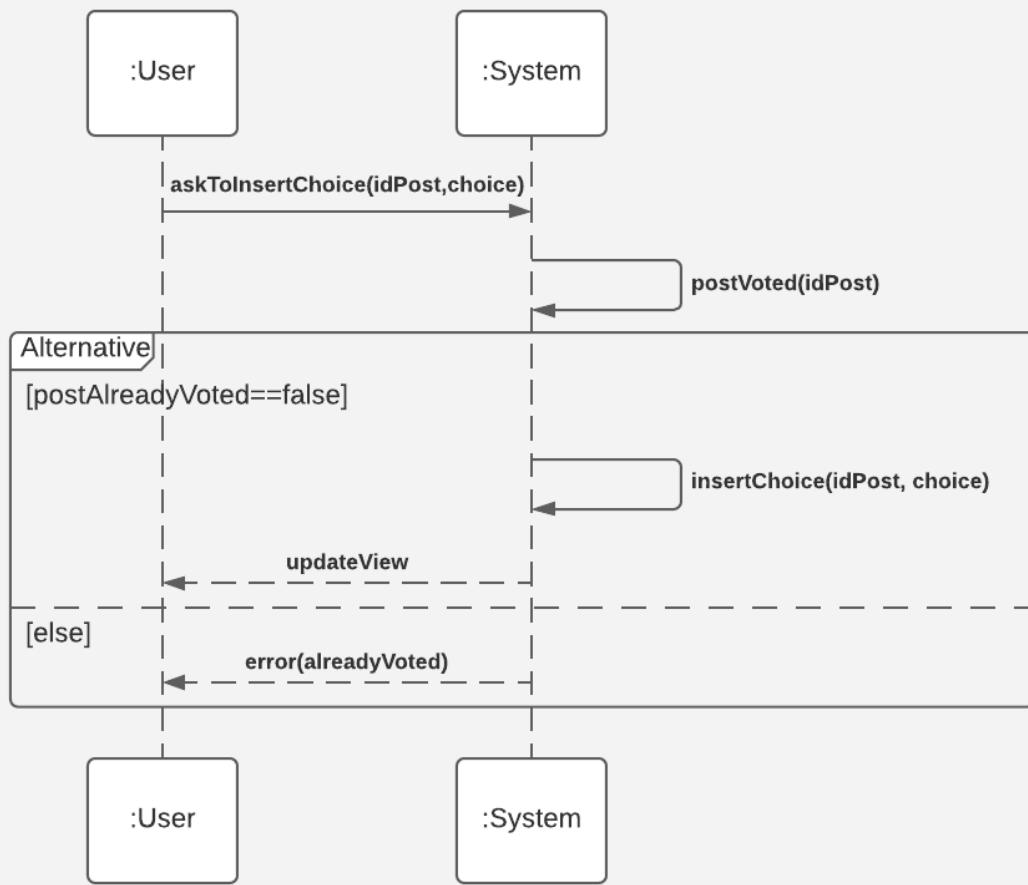


Figure 2.1



4.4. SSD - UC4

a. Basic flow Create Post

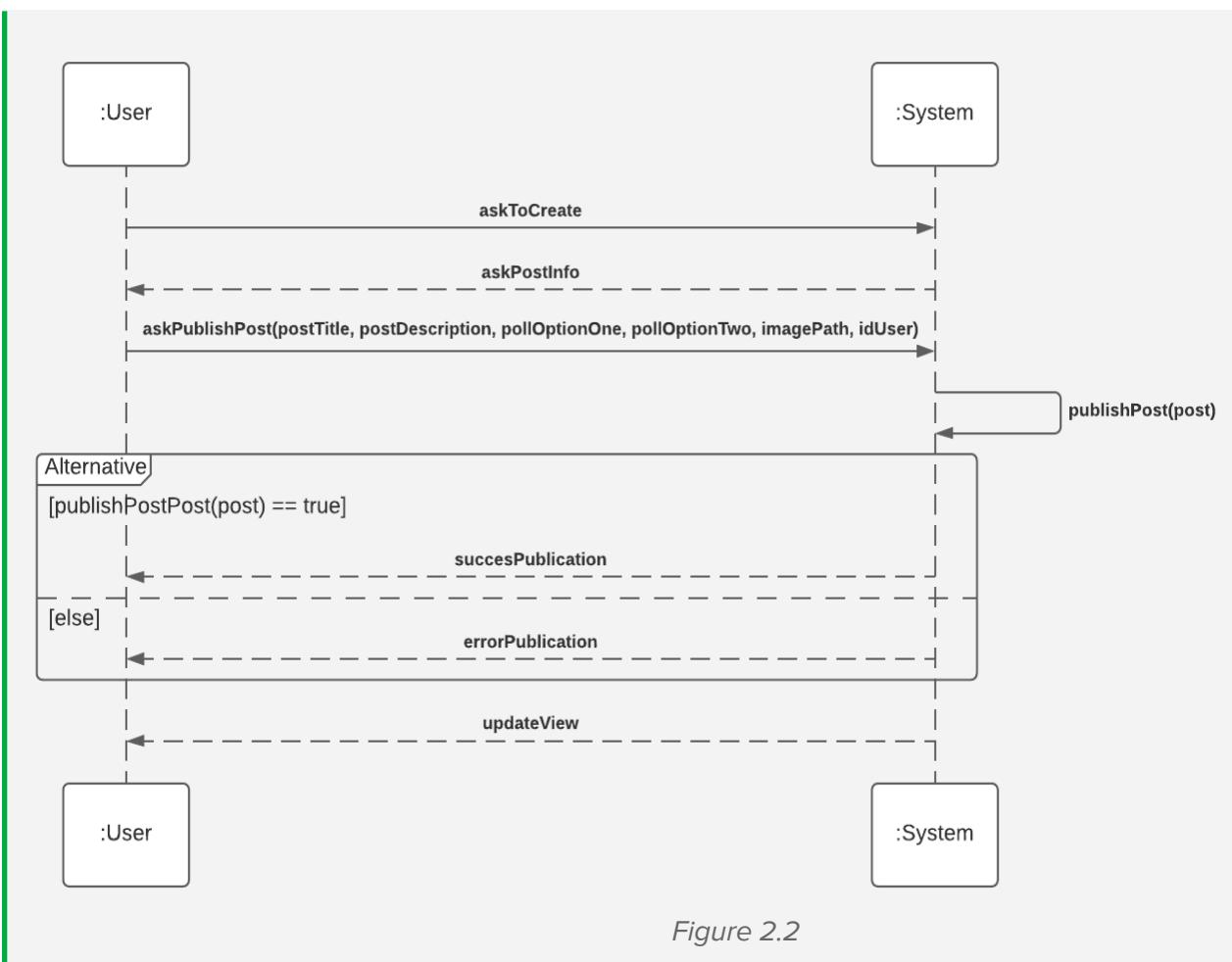


Figure 2.2



b. Alternative stream Edit Post

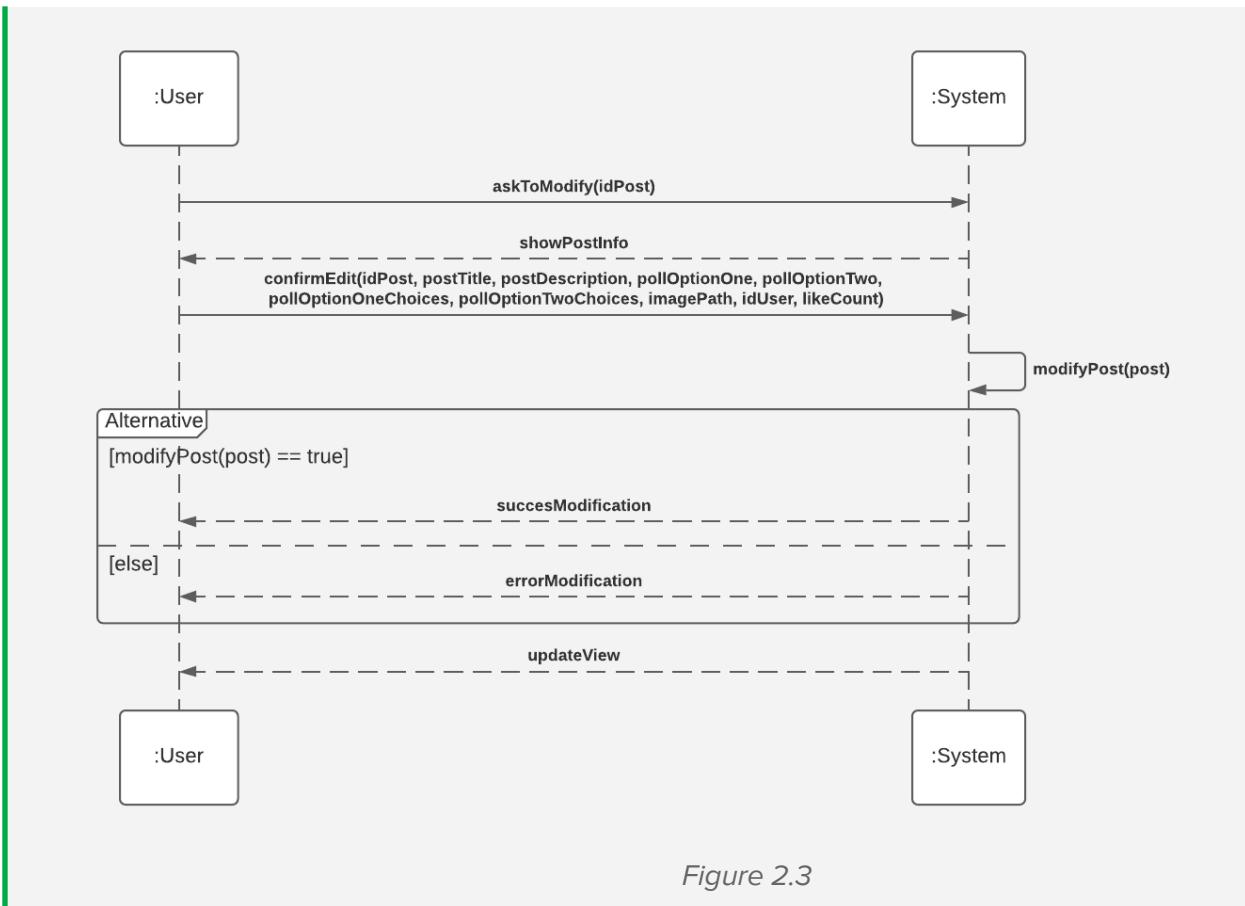


Figure 2.3



c. Alternative stream Remove Post

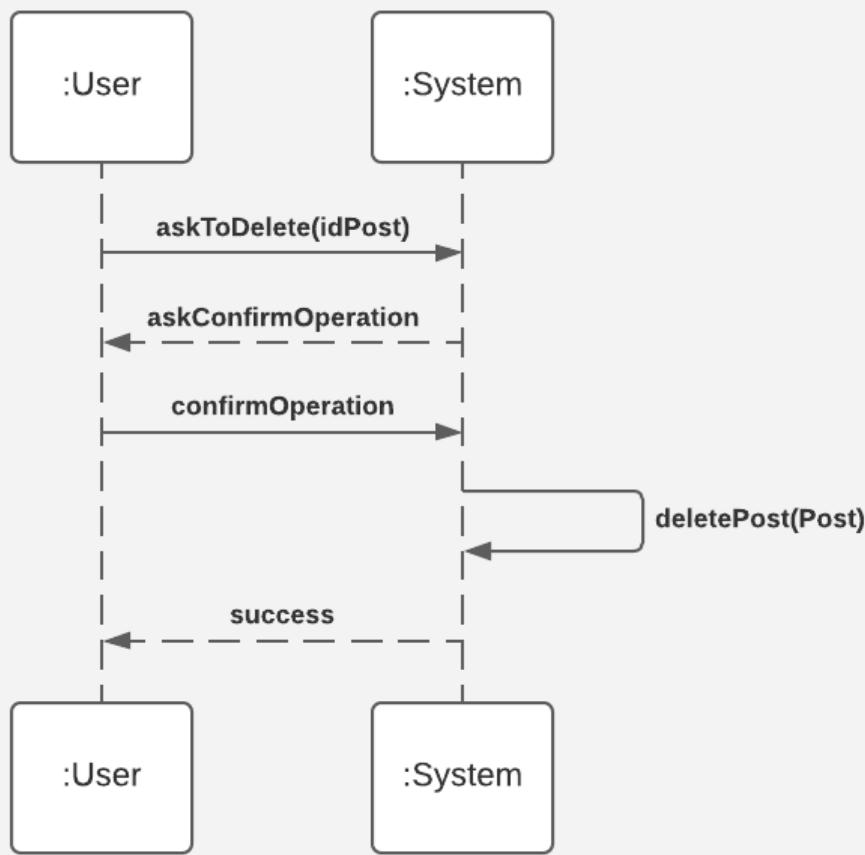


Figure 2.4



4.5. SSD - UC5

a. Basic flow Create Configuration

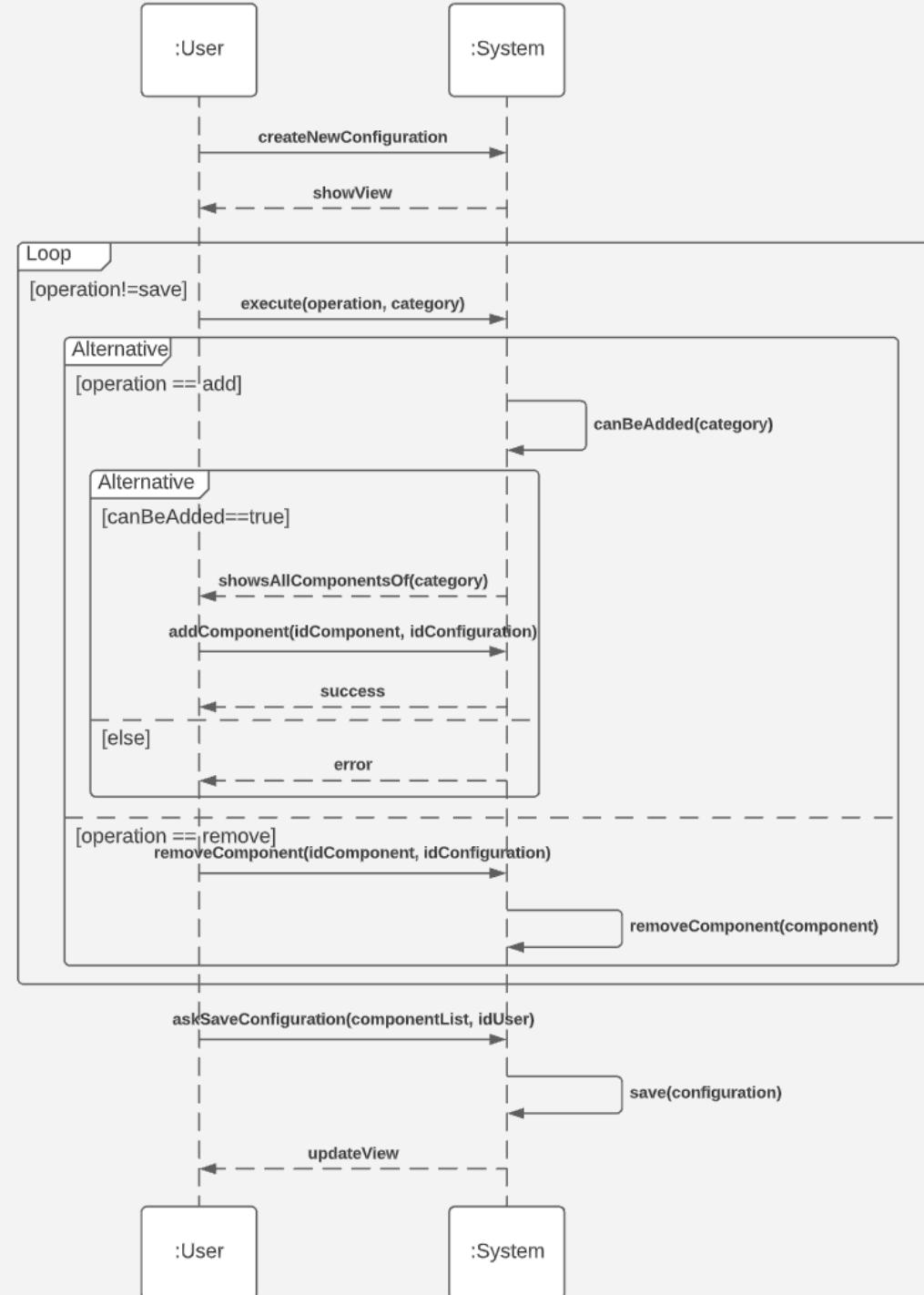


Figure 2.5



b. Alternative stream Remove Configuration

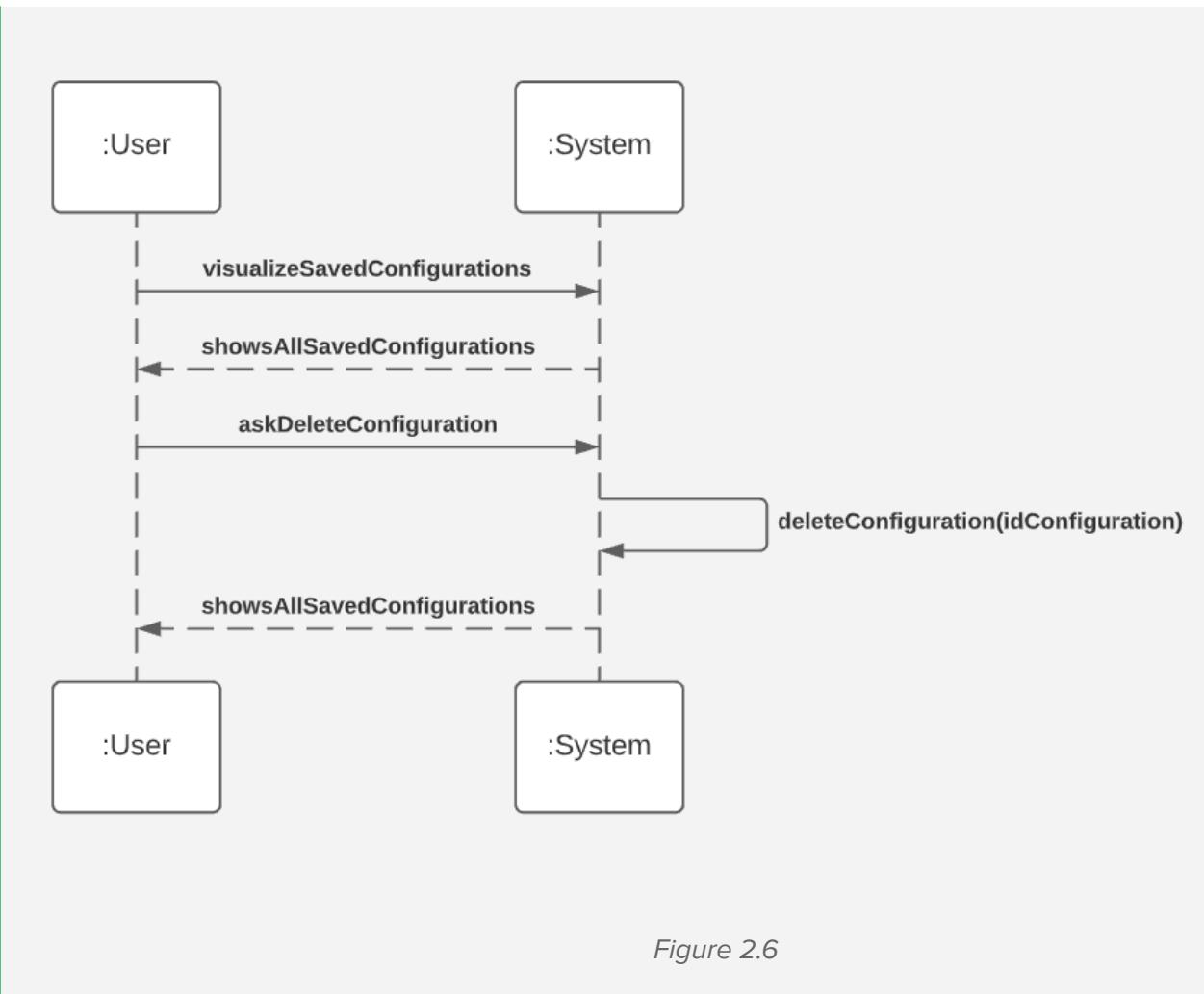


Figure 2.6



c. Alternative stream Modify Configuration

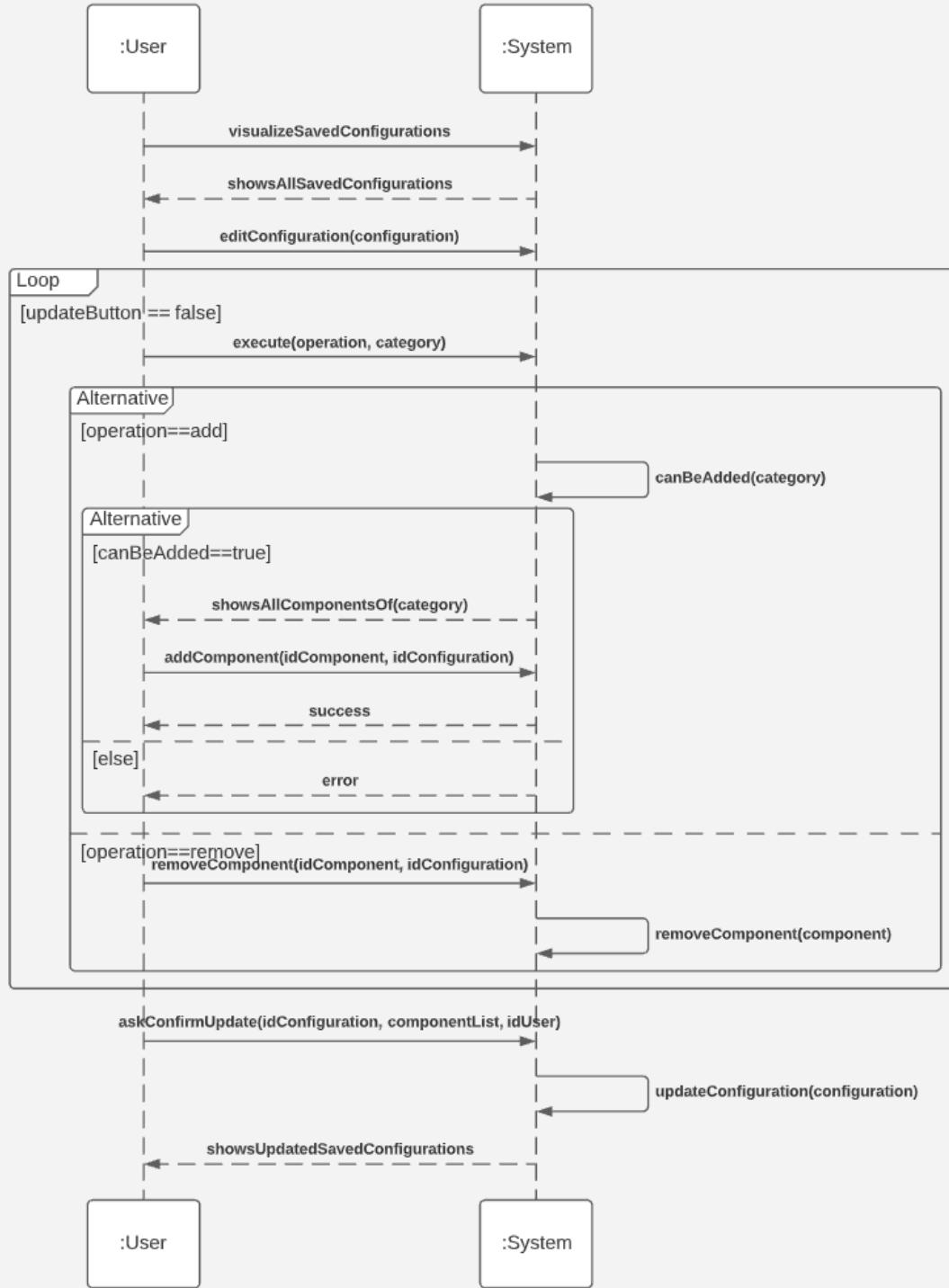


Figure 2.7



5. Contracts of Operations

5.1. User Operations Contracts

a. CO Delete user

Operation: applyDelete(users)

References:[4.1.c SSD UC1 - Delete User](#)

Pre-conditions: A controller has sent user data to be deleted.

Post-conditions: All references associated with each instance are deleted user of the list users.

b. CO SelectionINtry to giveRemove

Operation: addToDelete(idUser)

References:[4.1.c SSD UC1 - Delete User](#)

Pre-conditions: A controller has requested to delete a user.

Post-conditions: The instance is added in of User with value u.id = idUser to list of users for removal

c. CO SelectionINtry to giveBothers

Operation: addToBan(idUser)

References:[4.1.b SSD UC1 - Ban User](#)

Pre-conditions: A controller has asked to ban a user.

Post-conditions: The instance is added in of User with value u.id = idUser to list of users for the ban

d. CO Ban user

Operation: applyBan(users, period)

References:[4.1.b SSD UC1 - Ban User](#)

Pre-conditions: A controller has sent the data of the users to be banned and the period.

Post-conditions: Instances are searched user of the list users, an instance is created bannedUser of bannedUser for each user

- a. bannedUser has been associated with the user
- b. bannedUser.duration has become period

e. CO Confirm ChangeINtent

Operation: confirmEdit(idUser, name, surname, username, email, avatar)

References:[4.1.a SSD UC1 - Modify User](#)

Pre-conditions: A controller has requested to edit a user.

Post-conditions: An instance is created user in Users

- a. user.idUser becomes idUser
- b. user.name has become name
- c. user.surname became surname
- d. user.username has become username
- e. user.email has become email
- f. user.avatar has become an avatar



f. CO Edit user

Operation: applyEdit(user)

References: [4.1.a SSD UC1 - Modify User](#)

Pre-conditions: A controller has sent new user data to be changed.

Post-conditions: The instance is searchedin of User with u.id = user.id

- a. The instance data is checkedin
- b. u.idUser becomes user.idUser
- c. u.name becomes user.name
- d. u.surname diventa user.surname
- e. u.email becomes user.email
- f. u.username diventa user.username
- g. uh.avatar becomes user.avatar

5.2. Contracts Operations Components

a. CO Select Component To Remove

Operation: addToDelete(idComponent)

References: [4.2.b SSD UC2 - Remove Components](#)

Pre-conditions: A request to delete components is in progress.

Post-conditions: The instance is selectedc of Component with value c.id = idComponent to list ofcomponents for removal.

b. CO Remove Components

Operation: applyDelete(components)

References: [4.2.b. SSD UC2 - Remove Components](#)

Pre-conditions: An inspector has requested that components be removed.

Post-conditions: All references associated with each instance are deletedcomponent of the listcomponents.

c. CO Confirm Change

Operation: confirmEdit(idComponent, title, amazonLink, imagePath, specifications)

References: [4.2.c. UC2 SSD - Component Change](#)

Pre-conditions: A controller has requested to modify a component.

Post-conditions: An instance is createdc of Components.

1. c.idComponent has become idComponent
2. c.title became title
3. c.amazonLink has become amazonLink
4. c.imagePath became imagePath
5. c.specifications has become specifications

d. CO Modification Component

Operation: applyEdit(component)

References: [4.2.c. UC2 SSD - Component Change](#)



Pre-conditions: A PLC has sent the new modified data of a component.

Post-conditions: The instance is searchedc of Components withc.id = component.id

- a. The instance data is checkedc
- b. c.idComponent has become idComponent
- c. c.title became title
- d. c.amazonLink has become amazonLink
- e. c.imagePath became imagePath
- f. c.specifications has become specifications

e. CO Enter Component data

Operation: insertComponent(title, amazonLink, imagePath, specifications)

References: [4.2.a. UC2 SSD - Add Component](#)

Pre-conditions: A controller has requested to add a component.

Post-conditions: The data of is checkedcomponent.

- a. An instance is createdcomponent of Components.
- b. c.title became title
- c. c.amazonLink has become amazonLink
- d. c.imagePath became imagePath
- e. c.specifications has become specifications

f. CO Add Component

Operation: applyInsert(component)

References: [4.2.a. UC2 SSD - Add Component](#)

Pre-conditions: A controller has sent data for a component to be added.

Post-conditions: The data of is checkedcomponent.

1. The instance is savedcomponent of Components.

5.3. Contracts OperationsPost interactions

a. CO Enter "Like"

Operation: insertLike(post)

References: [4.3.a UC3 SSD - Like it](#)

Pre-conditions: A user asked to "like" a post.

Post-conditions: An instance has been createdlike of Likes

1. A like the user who inserted the like is associated.
2. A like the post selected by the user is associated.
3. The post's likes counter is incremented by one.

b. CO Remove "Like"

Operation: removeLike(post)

References: [4.3.a UC3 SSD - Like it](#)



Pre-conditions: A user has asked to remove a "like" from a post.

Post-conditions: The instance is searched like of Likes associated with the user

1. A like the user who inserted the like is disassociated.
2. A like the post selected by the user is disassociated.
3. The post's likes counter is decremented by one.
4. like is deleted

c. CO Check if the post has a “Like” associated with it

Operation: postLiked(post)

References: [4.3.a UC3 SSD - Like it](#)

Pre-conditions: A user asked to "like" a post.

Post-conditions: Looks for whether there is a Like instance associated with the instance post of Posts

d. CO Check if the post has an associated "Poll Vote"

Operation: postVoted(post)

References: [4.3.b UC3 SSD - Reply to the Survey](#)

Pre-conditions: A user asked to enter a "Vote" to a "survey".

Post-conditions: Looks for whether there is a PollOption instance associated with the instance post of Posts

e. CO Insert “Poll Voting”

Operation: insertChoice(post, choice)

References: [4.3.b UC3 SSD - Reply to the Survey](#)

Pre-conditions: A user asked to insert a "Vote" to a "Poll" in a post.

Post-conditions: An instance has been created option of PollOption containing the user choice

1. Ad option the user who entered the choice is associated.
2. Ad option the post selected by the user is associated.
3. The post choice counter is incremented by one.



5.4. Contratti OperationsPost

a. CO Remove Post

Operation: deletePost(post)

References:[4.4.c SSD UC4 - Remove Post](#)

Pre-conditions:A user has asked to delete a post.

Post-conditions: The instance is searchedpost of Posts to be deleted

1. All references belonging to are dissociatedpost
2. Thepost is deleted

b. CO Modify Post

Operation: modifyPost(post)

References:[4.4.b SSD UC4 - Modify Post](#)

Pre-conditions:A user has asked to edit his post.

Post-conditions: The one with the post is searched among the instances

Post.idPost = post.idPost

1. Post's attributes are set (if different) to those ofpost

c. CO Publish Post

Operation: publishPost(post)

References:[4.4.a SSD UC4 - Crea Post](#)

Pre-conditions:A user has created a post and wants it to be registered in the system

Post-conditions: An instance of Post is created with attributes initialized to those ofpost

1. L'idPost of the new post instance is set to be unique

d. CO Post modification request

Operation: confirmEdit(idPost, postTitle, postDescription, pollOptionOne, pollOptionTwo, pollOptionOneChoices, pollOptionTwoChoices, imagePath, idUser, likeCount)

References:[4.4.b SSD UC4 - Modify Post](#)

Pre-conditions:A user modified some fields of a Post instance (with Post.idPost = idPost).

Post-conditions: a Post instance is createdpost

1. the post.idPost attribute is initialized to idPost.
2. the post.postTitle attribute is initialized to postTitle.
3. l'attributo post.postDescription viene initializzato a postDescription
4. the post.pollOptionOne attribute is initialized to pollOptionOne
5. the post.pollOptionTwo attribute is initialized to pollOptionTwo



-
- 6. the post.pollOptionOneChoices attribute is initialized to pollOptionOneChoices
 - 7. the post.pollOptionTwoChoices attribute is initialized to pollOptionTwoChoices
 - 8. the post.imagePath attribute is initialized to imagePath
 - a. if the image is a "file" it is "registered" in the system and its path is obtained
 - b. the post.imagePath attribute is initialized with the path.
 - 9. the post.idUser attribute is initialized to idUser
 - 10. the post.likeCount attribute is initialized to likeCount

e. CO Request for publication Post

Operation: askPublishPost(postTitle, postDescription, pollOptionOne, pollOptionTwo, imagePath, idUser)

References:[4.4.a SSD UC4 - Crea Post](#)

Pre-conditions: A user has "compiled" a post and asked to publish it

Post-conditions: a Post instance is created post

- 1. the post.idPost attribute is initialized with a unique progressive
- 2. the post.postTitle attribute is initialized to postTitle.
- 3. l'attributo post.postDescription viene initializzato a postDescription
- 4. the post.pollOptionOne attribute is initialized to pollOptionOne
- 5. the post.pollOptionTwo attribute is initialized to pollOptionTwo
- 6. the post.pollOptionOneChoices attribute is initialized to 0
- 7. the post.pollOptionTwoChoices attribute is initialized to 0
- 8. the "image" is "registered" in the system and the path
 - a. the post.imagePath attribute is initialized with the path.
- 9. the post.idUser attribute is initialized to idUser
- 10. the post.likeCount attribute is initialized to 0



5.5. Contracts Operations Configurations

a. CO Modify Configuration

Operation: updateConfiguration (configuration)

References:[4.5.c UC5 SSD - Change Configuration](#)

Pre-conditions:A user has asked to edit a saved configuration.

Post-conditions: The Configuration instance with idConfiguration == is searchedconfiguration.idConfig

1. The attributes of Configuration are set (if different) to those ofconfiguration

b. CO Create Configuration

Operation: save (configuration)

References:[4.5.a UC5 SSD - Create Configuration](#)

Pre-conditions:A user asked to delete a saved configuration.

Post-conditions: An instance of Configuration is created with attributes initialized to those ofconfiguration

1. The idConfiguration of the new post instance is set to be unique

c. CO Request Delete Configuration

Operation: deleteConfiguration (configuration)

References:[4.5.b UC5 SSD - Delete Configuration](#)

Pre-conditions:A user asked to delete a saved configuration.

Post-conditions: The instance of is searchedconfiguration di Configuration in cui Configuration.idConfiguration == idConfiguration

1. All references belonging to are detachedconfiguration
2. configuration is deleted

d. CO Request Change Configuration

Operation: askConfirmUpdate(idConfiguration, componentList, idUser)

References:[4.5.c UC5 SSD - Change Configuration](#)

Pre-conditions:A user has asked to edit a saved configuration.

Post-conditions: an instance is createdconfiguration di Configuration

1. The attributeconfiguration.idConfiguration is initialized toidConfiguration
2. The attributeconfiguration.componentList is initialized tocomponentList
3. The attributeconfiguration.idUser is initialized toidUser

e. CO Create Configuration

Operation: askSaveConfiguration(componentList,idUser)

References:[4.5.a UC5 SSD - Create Configuration](#)

Pre-conditions:A user asked to delete a configuration from the saved ones.

Post-conditions: The instance is createdconfiguration di Configuration

1. The attributeconfiguration.idConfiguration is initialized with a unique progressive



-
- 2. The attribute configuration.componentList is initialized
 - 3. The attribute configuration.idUser is initialized

f. WHAT AAdd Component

Operation: addComponent (idComponent, idConfiguration)

References: [4.5.c UC5 SSD - Change Configuration](#), [4.5.a UC5 SSD - Create Configuration](#)

Pre-conditions: A user asked to add a component to a configuration

Post-conditions: The instance is searchedconfiguration of Configuration whereconfiguration.idConfiguration == idConfiguration

- 1. idComponent is added toconfiguration.componentList

g. CO Remove Component

Operation: removeComponent (component, idConfiguration)

References: [4.5.c UC5 SSD - Change Configuration](#), [4.5.a UC5 SSD - Create Configuration](#)

Pre-conditions: A user requested to remove a component from a configuration

Post-conditions: The instance is searchedconfiguration of Configuration whereconfiguration.idConfiguration == idConfiguration

- 1. show upidComponent is removed fromconfiguration.componentList

h. Co. Efollow Operation

Operation: execute (operation, category)

References: [4.5.c UC5 SSD - Change Configuration](#), [4.5.a UC5 SSD - Create Configuration](#)

Pre-conditions: A user has requested to perform an add or remove operation on a component of a configuration.

Post-conditions: The operation operation is performed on the component category category

i. CO CCheck Addition

Operation: canBeAdded (category)

References: [4.5.c UC5 SSD - Change Configuration](#), [4.5.a UC5 SSD - Create Configuration](#)

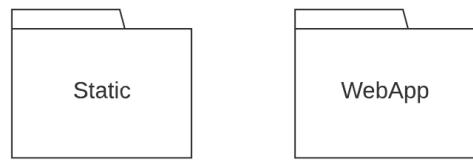
Pre-conditions: A user requested to add a component of a category to a configuration

Post-conditions: the system checks if a member of the category category can be added multiple times

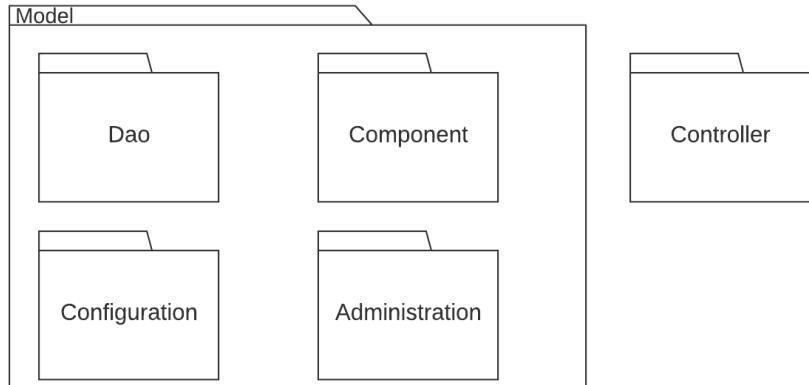
6. ALogical architecture



User Interface



Application Logic



Technical Service

