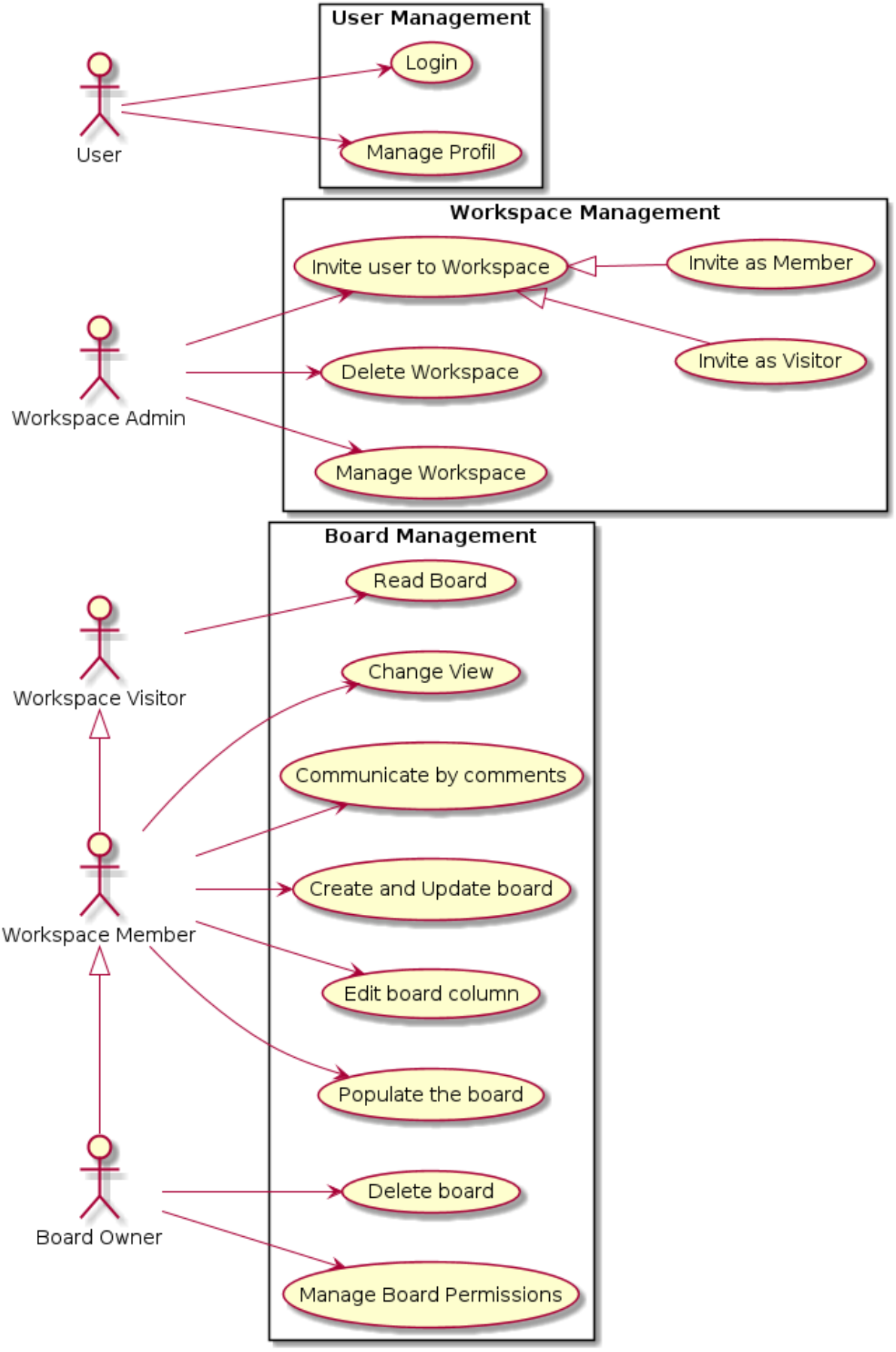
Use Case Analysis

AZHARHOUSSEN Salim
BA Birane
BOURRET Raphaël
GALOIS Nicolas

19/11/2020

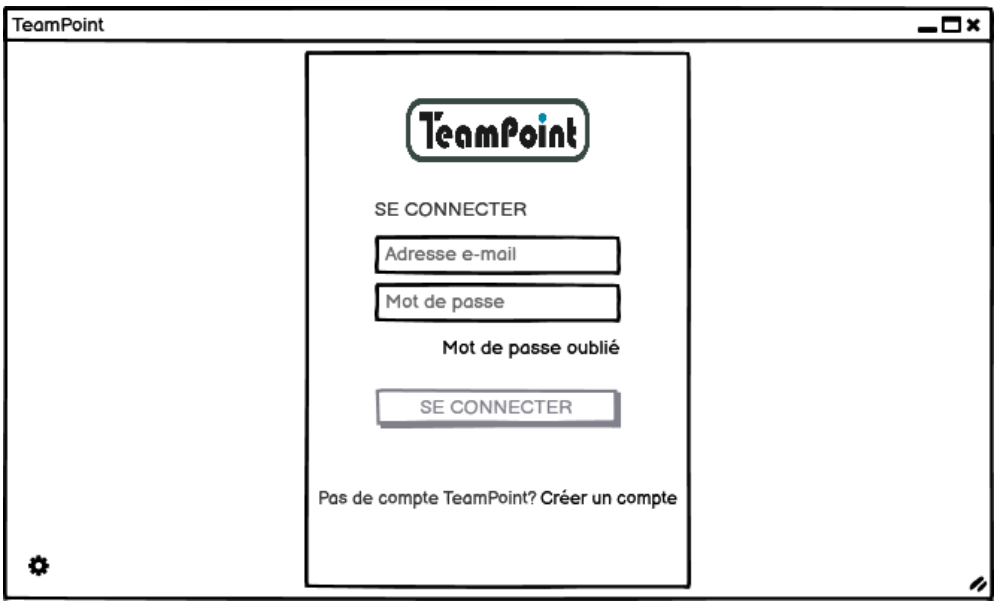
Use Case Diagram



Use Case Description

1. Login

1.1. Mockups



1.2. Brief Description

This use case describes how a user logs into TeamPoint.

1.3. Flow of Events

1.3.1. Basic Flow

This use case starts when the user wishes to sign up to the Team Point system.

1. The system requests that the user enter his email address and password
2. The user enters his email address and password.
3. The system validates the user input by checking his email and password are valid
4. The system logs the user into TeamPoint.

1.3.2. Alternative Flows

1.3.2.1. Invalid email/Password

If the user enters an invalid email address and/or password, the system displays an error message. The user can choose to either return to the beginning of the Basic Flow or cancel the login, at which point the use case ends.

1.4. Pre-Conditions

The user must have an account.

1.5. Post-Conditions

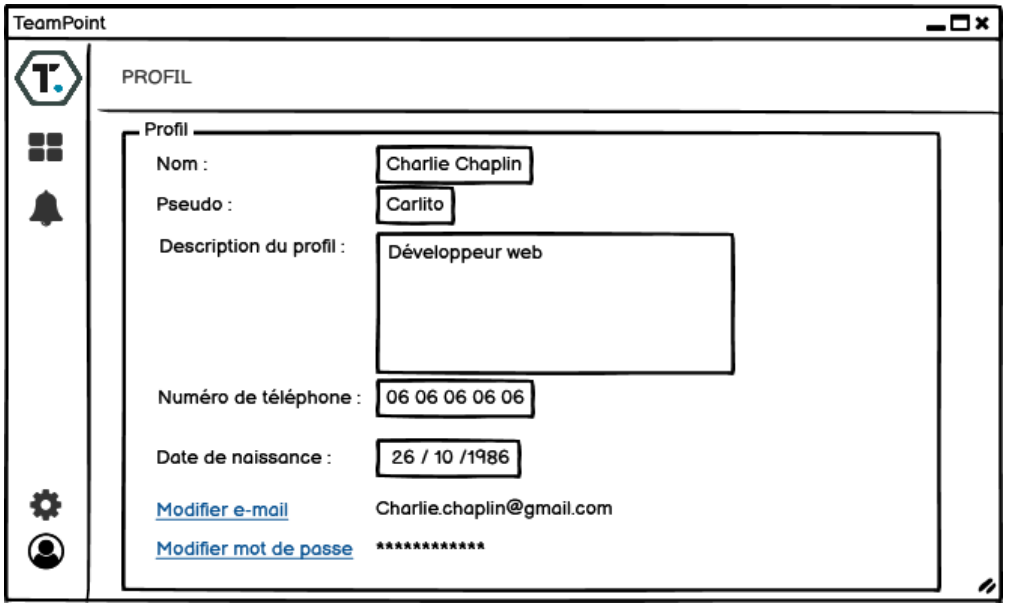
If the use case was successful, the user is now logged into TeamPoint. If not, the system state is unchanged.

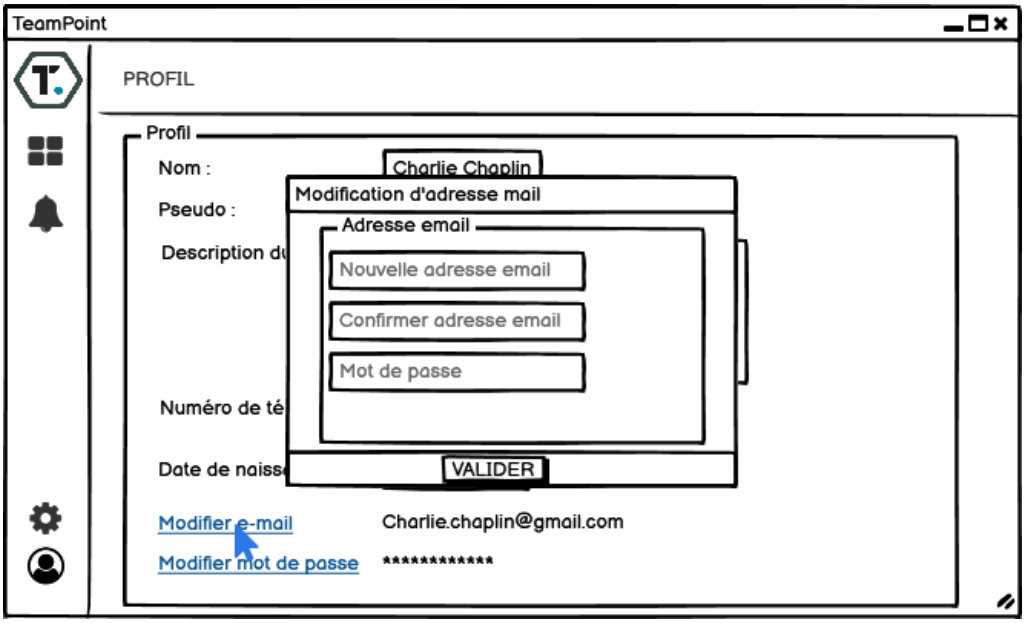
1.6. Extension Points

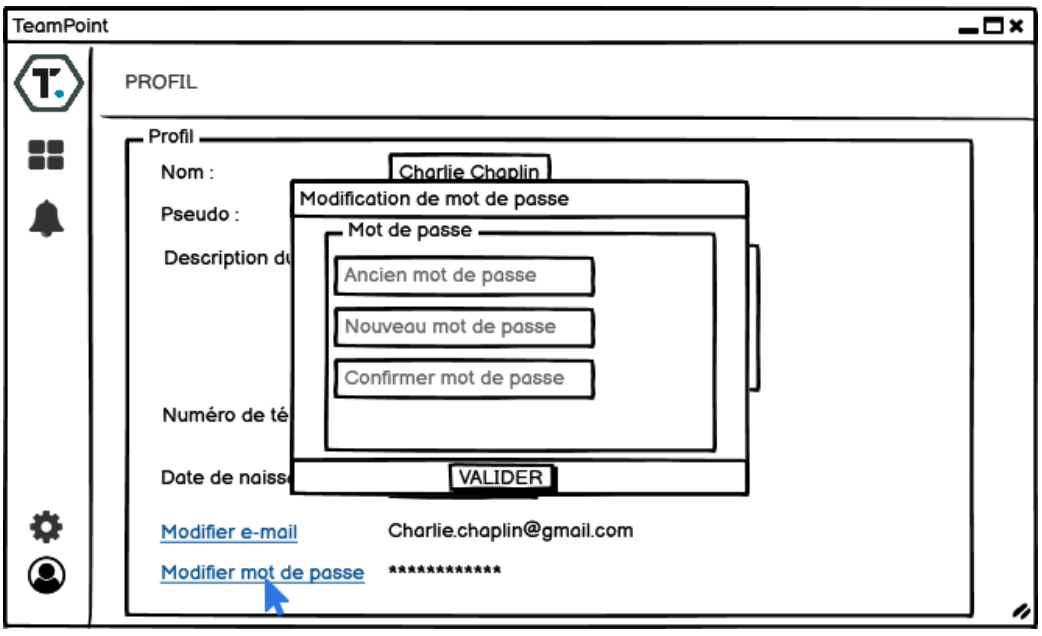
Forgotten password : The user can set a new password if forgotten.
Sign up : if the user does not have an account yet, he can create one.

2. Manage profile

2.1. Mockups







2.2. Brief Description

This use case describes how a user can manage his profile. The user will be able to edit the following fields: name, surname, profile description, birthday and phone number. Also the user will be able to change his email address and/or his password.

2.3. Flow of Events

2.3.1. Basic Flow

2.3.1.1. Basic Flow for common fields

This use case describes how a user edits a field of his profile. The relevant fields are: name, surname, profile description, birthday and phone number.

1. The system shows the user profile and all the current values of the profile.
2. The user edits one or several fields.
3. The system automatically updates the user profile.

2.3.1.2. Basic Flow for Password modification

1. The system asks for the user's old password, new password and new password confirmation.
2. The user gives the input for the old password and the new one plus the new password confirmation.
3. The system checks if the old password is correct.
4. The system checks if the new password and the password confirmation are equals.
5. The system updates the user's password.

2.3.1.3. Basic Flow for Email modification

1. The system asks for the user's new email, the new email confirmation and the user's password.
2. The user gives the input for the new email and the email confirmation and the password.
3. The system checks if the new email and the new email confirmation are equal
4. The system checks if the password is correct.
5. The system updates the user's email address.

2.3.2. Alternative Flow

2.3.2.1 Invalid input for password mofidication

If the old password is not correct:
The system displays an error message to inform the user. The user can choose to either return to the beginning of the basic flow for password modification or cancel it, at which point the use case ends.

If the new password and the new password confirmation are not equal:

The system displays an error message to inform the user. The user can choose to either return to the beginning of the basic flow for password modification or cancel it, at which point the use case ends.

2.3.2.2 Invalid input for email address modification

If the password is not correct:
The system displays an error message to inform the user. The user can choose to either return to the beginning of the basic flow for email modification or cancel it, at which point the use case ends.

If the new email address and the new email address confirmation confirmation are not equal:
The system displays an error message to inform the user. The user can choose to either return to the beginning of the basic flow for email modification or cancel it, at which point the use case ends.

2.4. Pre-Conditions

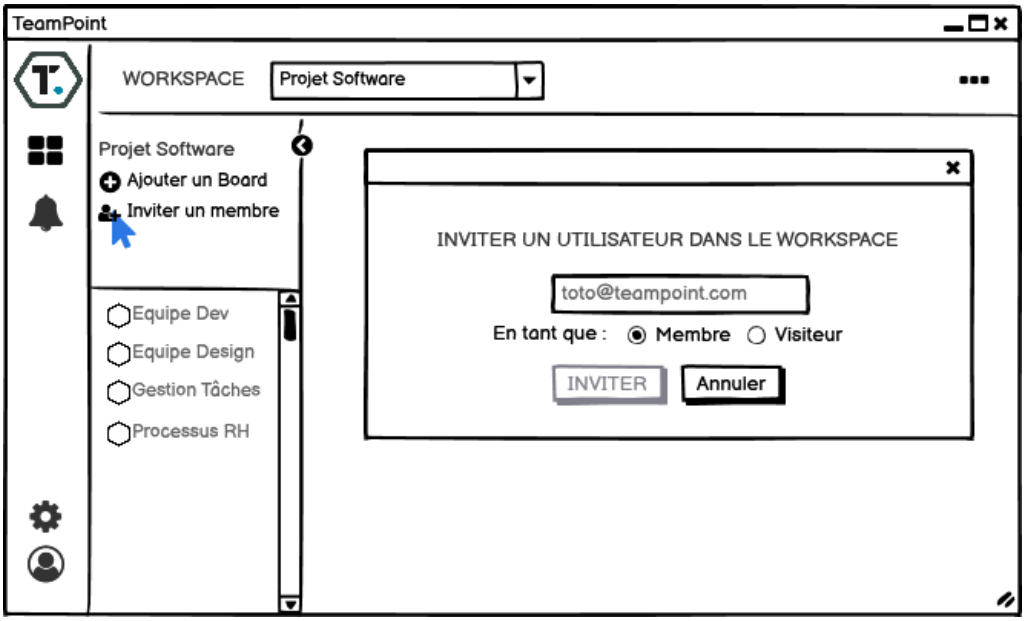
To edit his profile, the user must be logged in with his account.

2.5. Post-Conditions

If the use case was successful, the profile of the user is up to date. If not, the user profile unchanged.

3. Invite user to Workspace

3.1. Mockups



3.2. Brief Description

This use case describes how the workspace administrator can invite others to a workspace. A workspace can have two types of users: Member and Visitor. The Visitor is a particular workspace user. He can only read the workspace and is not allowed to modify anything. Thus the workspace administrator can send two type of invitation if he want the invited user to be a workspace member or a workspace visitor

3.3. Basic flow

This use case describes how the workspace administrator can invite others to a workspace.

1. The system asks for one or several emails separated by a semicolon and the type of invitation (Member or Visitor)
2. The workspace administrator provides one or several emails to the system and the invitation type.
3. The system checks if all emails belong to an account.
4. The system sends the invitation to the corresponding user

3.4. Pre-Conditions

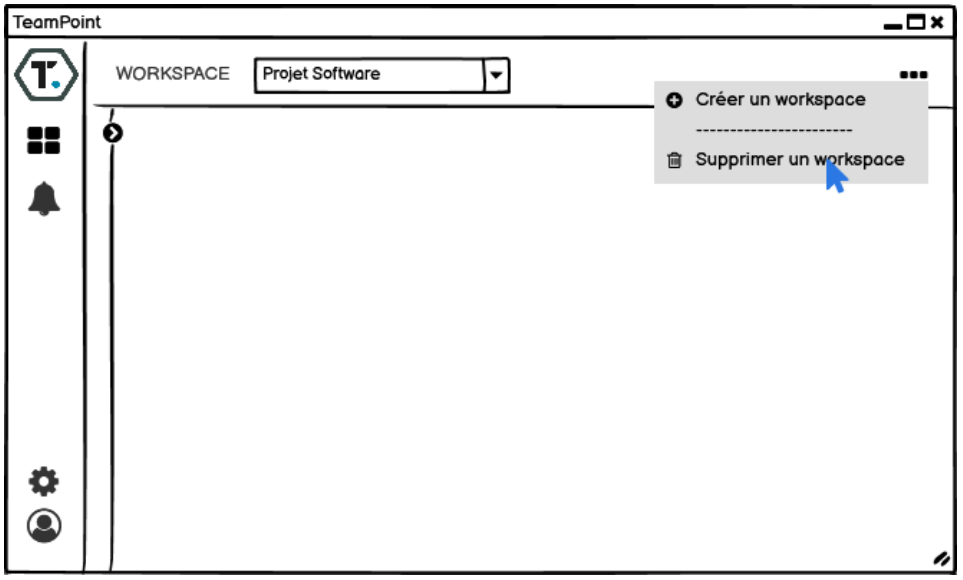
To invite a user in a workspace, the user has to be a workspace administrator.

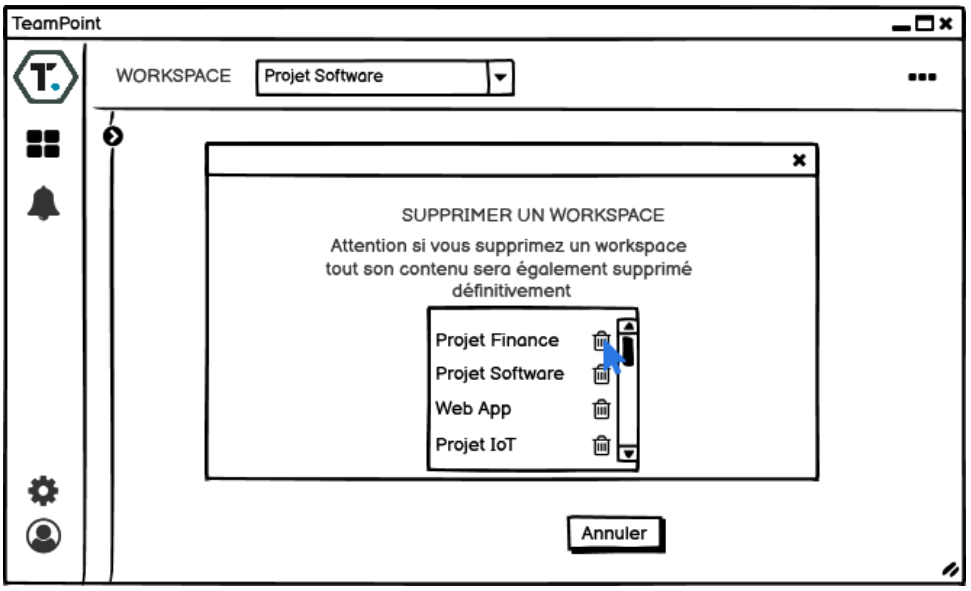
3.5. Post-Conditions

The user who is invited as a member receives an invitation notification and if he accepts, he can see the workspace and edit it.
The user who is invited as a visitor receives an invitation notification and if he accepts, he can only see the workspace.

4. Delete workspace

4.1. Mockups





4.2. Brief Description

This use case describes how the workspace administrator can delete his workspace. If a workspace is deleted then all the relevant boards and discussions will be deleted.

4.3. Basic Flow

This use case starts when the workspace administrator wants to delete it.

1. The system displays the list of the workspace where the user is a workspace administrator.
2. The user selects the workspace he wants to delete.
3. The system asks the user to give his password to validate the operation.
4. The system deletes the workspace permanently.

4.4. Pre-Conditions

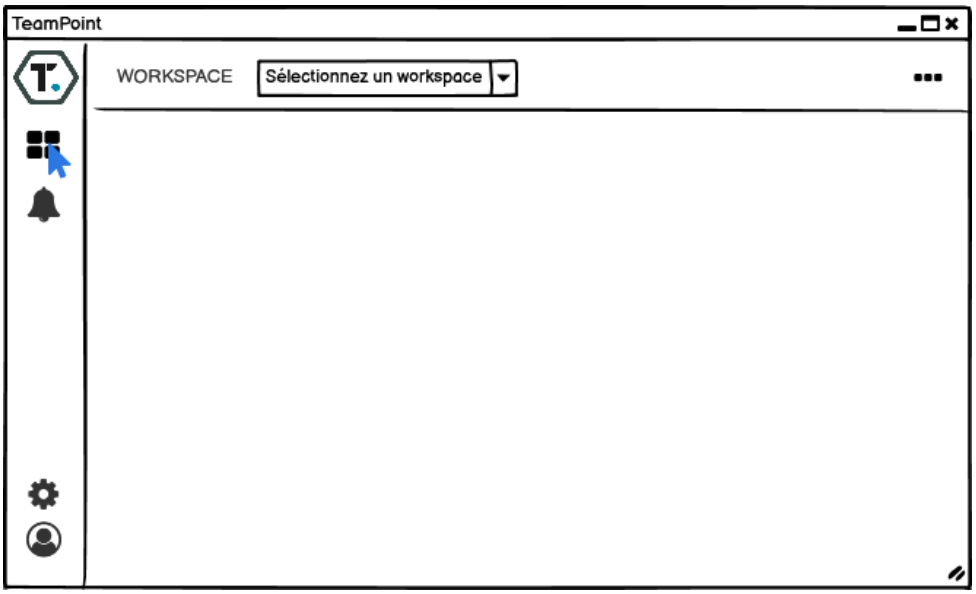
The user has to be a workspace administrator of the workspace.

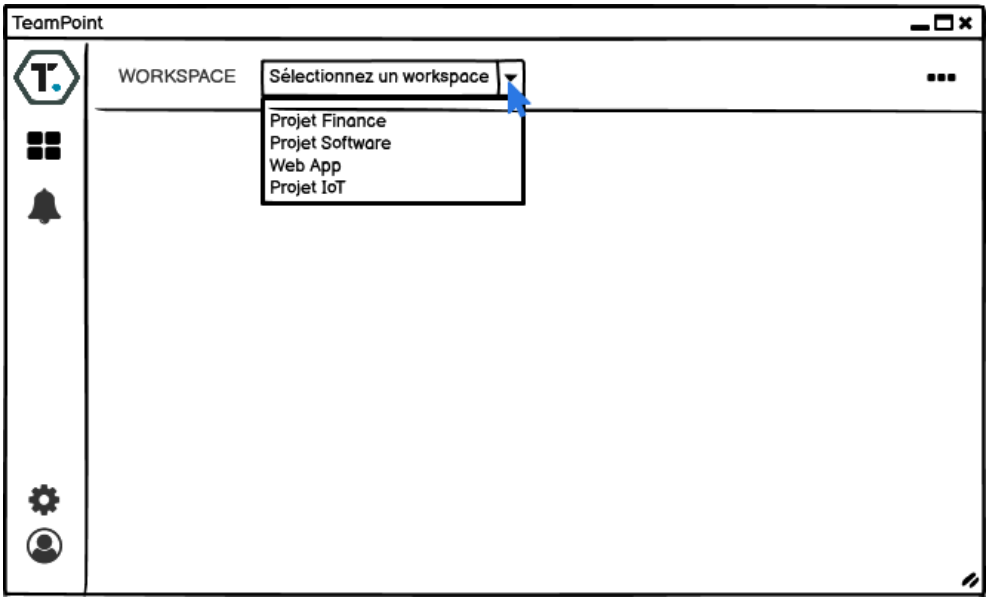
4.5. Post-Conditions

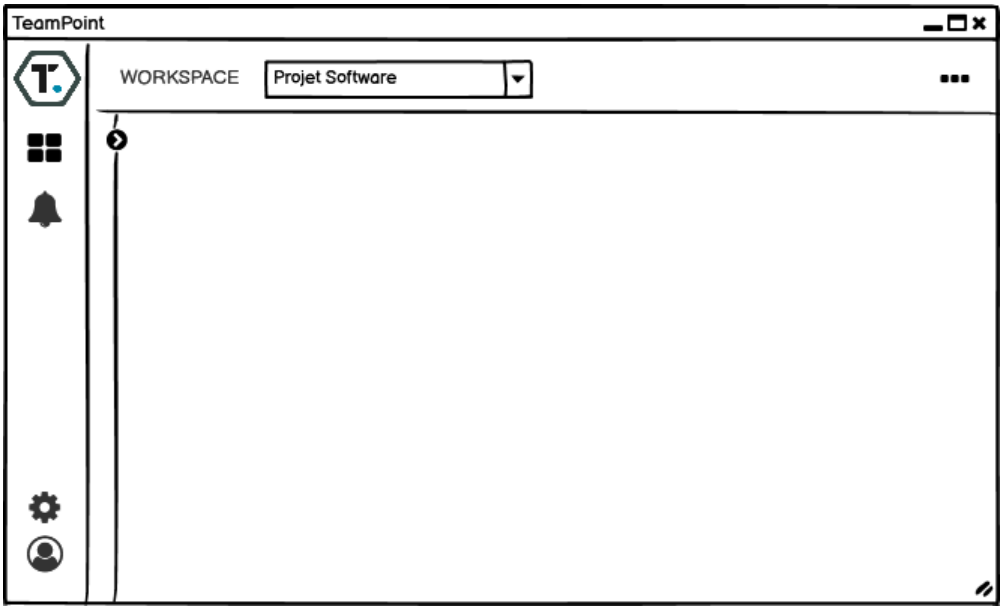
The workspace is totally deleted.

5. Manage Workspace

5.1. Mockups







5.2. Brief Description

This use case describes the management of the workspaces. A workspace is composed of a list of boards. A workspace is the root of a project.

5.3. Flow of Events

5.3.1. Basic Flow

5.3.1.1. Workspace Navigation

This use case describes how a user sees the workspaces where he is a member and navigates between them.

1. The system shows a list of workspaces the user has access to.
2. The user selects the one he wants to navigate to.
3. The system loads the workspace for the user.

5.3.1.2. Creation of Workspaces

This use case describes how a user creates a new workspace.

1. The user provides a name.
2. The system create the new workspace by the name given
3. The user is loaded into that workspace.

5.4. Pre-Conditions

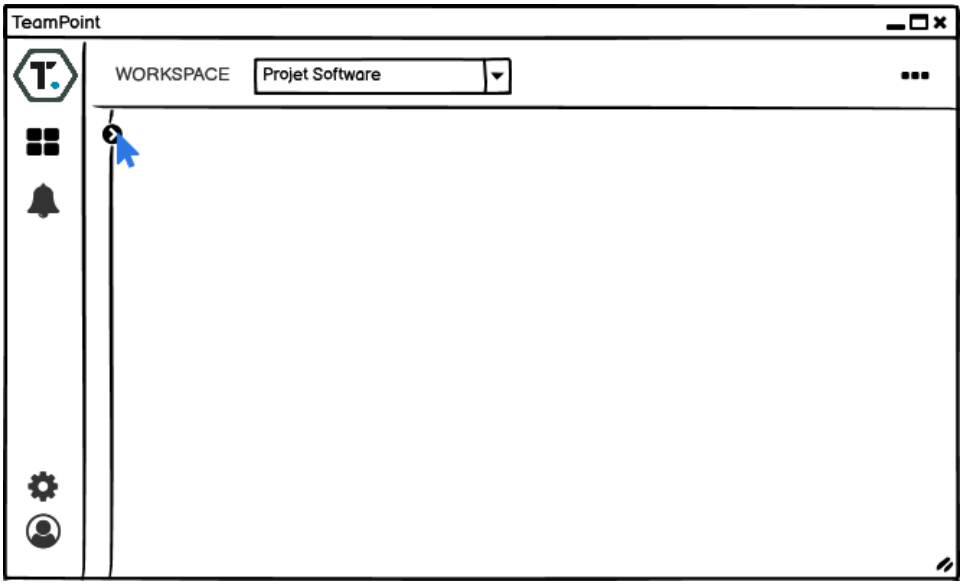
The user has to be logged to TeamPoint.

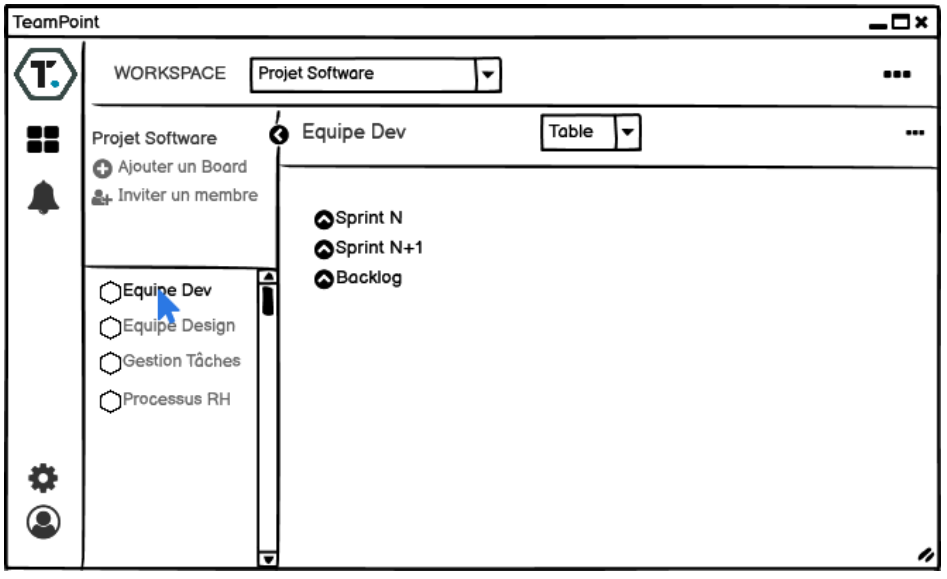
5.5. Post-Conditions

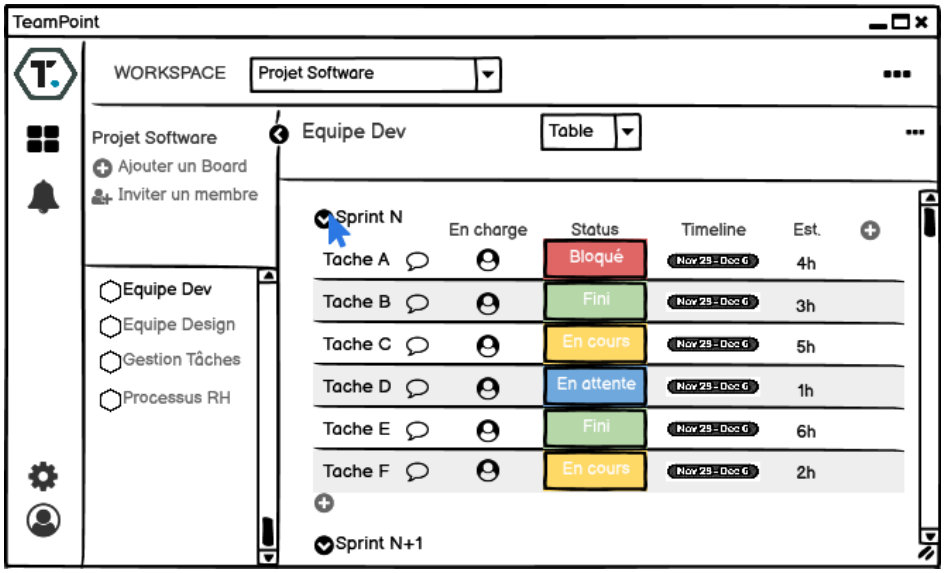
The workspace is loaded into the view.
The user who created the workspace becomes the workspace administrator.

6. Read board

6.1. Mockups







6.2. Brief Description

This use case describes the views of a board. A board will be in a specific workspace. All workspace's members/visitors can see all of that workspace's board. The board default view Table. A board is composed of items and columns. An item is represented by a row and has a label. The label is the item name.

6.3. Flow of Events

6.3.1. Basic Flow

This use case starts when the actor wants to read a board.

1. The system displays all the workspace's board as a list.
2. The user selects the board he wants to see
3. The system displays the corresponding board

6.4. Pre-Conditions

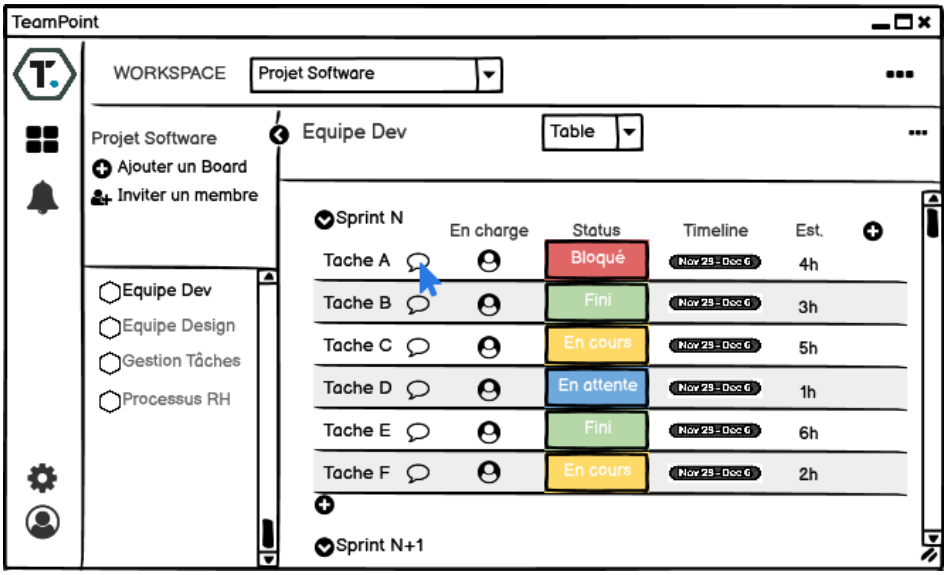
The board has to exist.

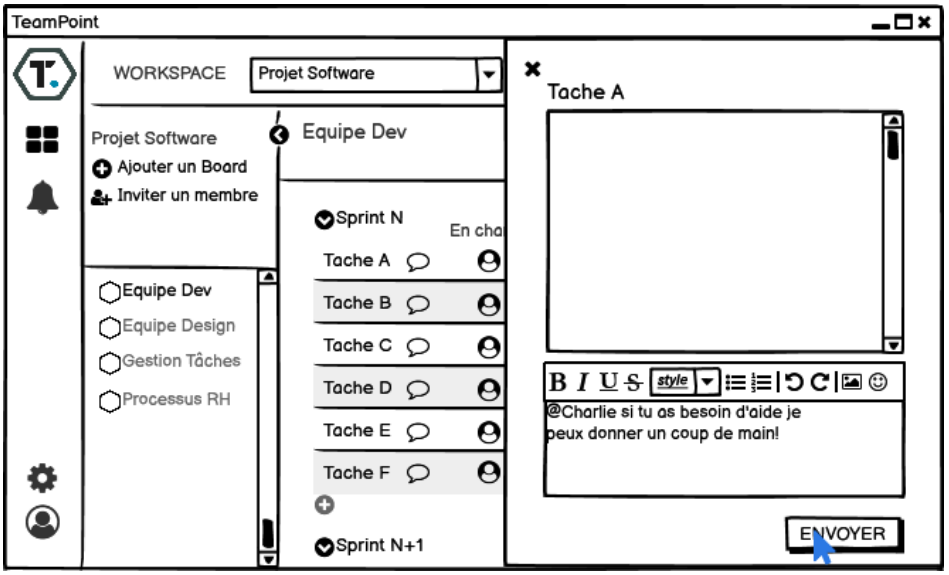
6.5. Post-Conditions

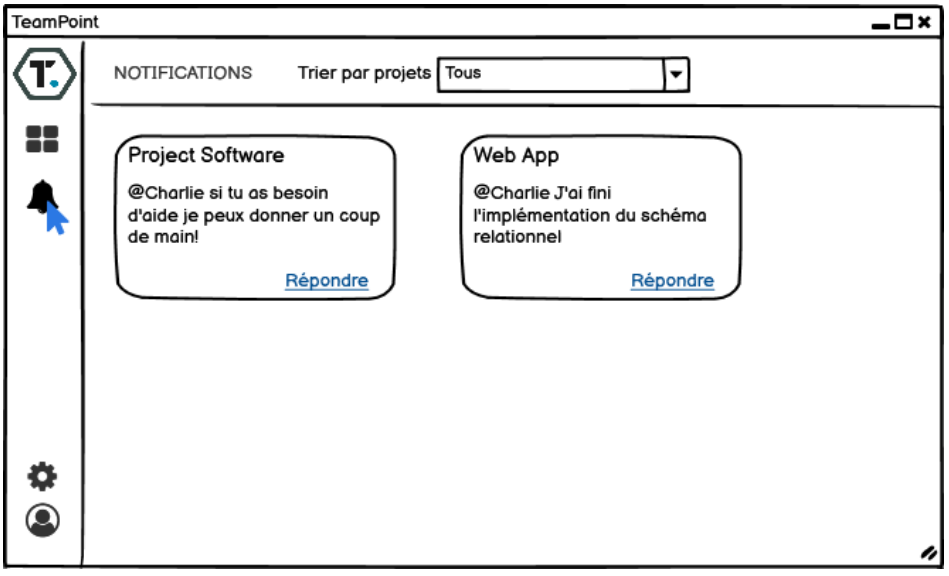
The board is displayed.

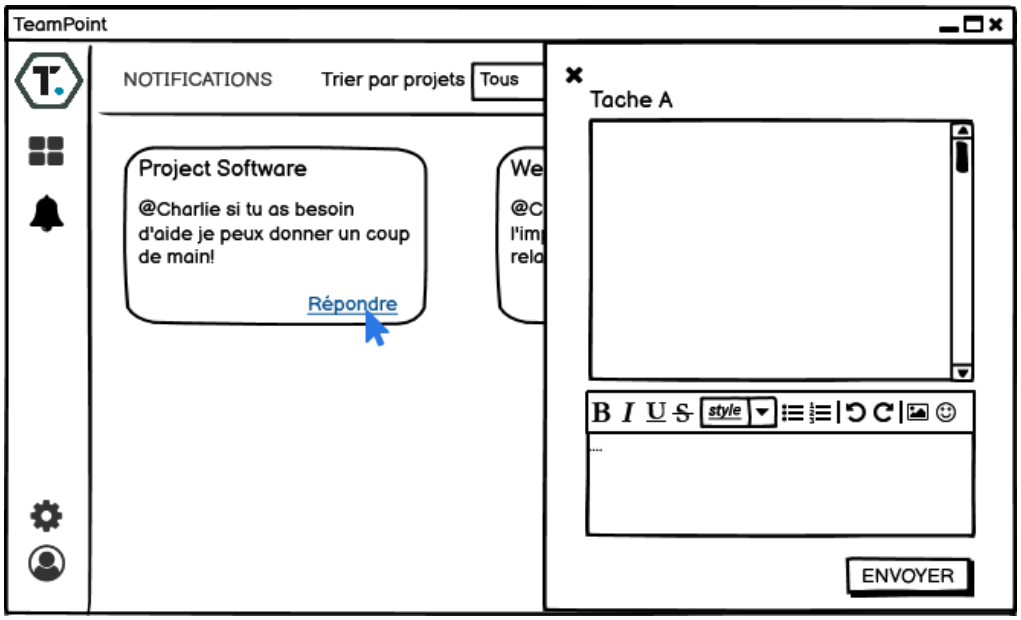
7. Communicate by comments

7.1. Mockups









7.2. Brief Description

This use case describes how a user communicates with his coworkers. Communication will be in the context of a specific item. An item is a row of a board. It means that for each item, a communication channel can be created. Any user can be tagged in a message with an at (@) and his member's id, therefore the tagged user will be notified. Any workspace member can see messages.

7.3. Flow of Events

7.3.1. Basic Flow

The basic flow describes how communication is made for a specific item.

1. The user selects an item.
2. The user writes the message.
3. The system sends the user's message in the channel.

7.3.2. Alternative flow

7.3.2.1. Channel creation

If the channel doesn't exist, the system creates a communication channel between all members for the corresponding item.

7.3.2.2. Message tagged

If there is a user tag in a message, the system sends a notification to the corresponding user.

7.4. Pre-Conditions

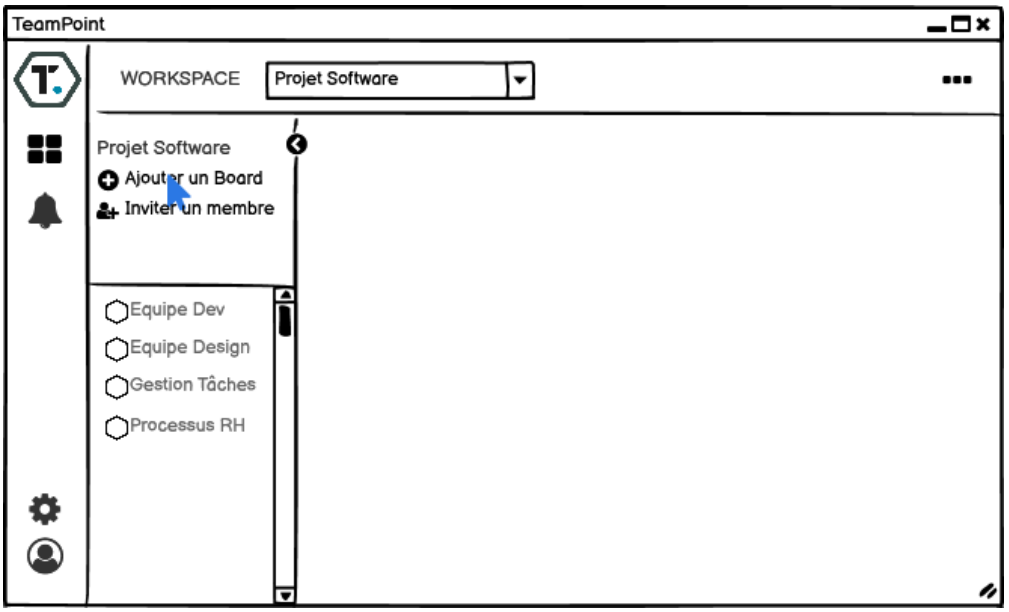
The user has to be a workspace member.

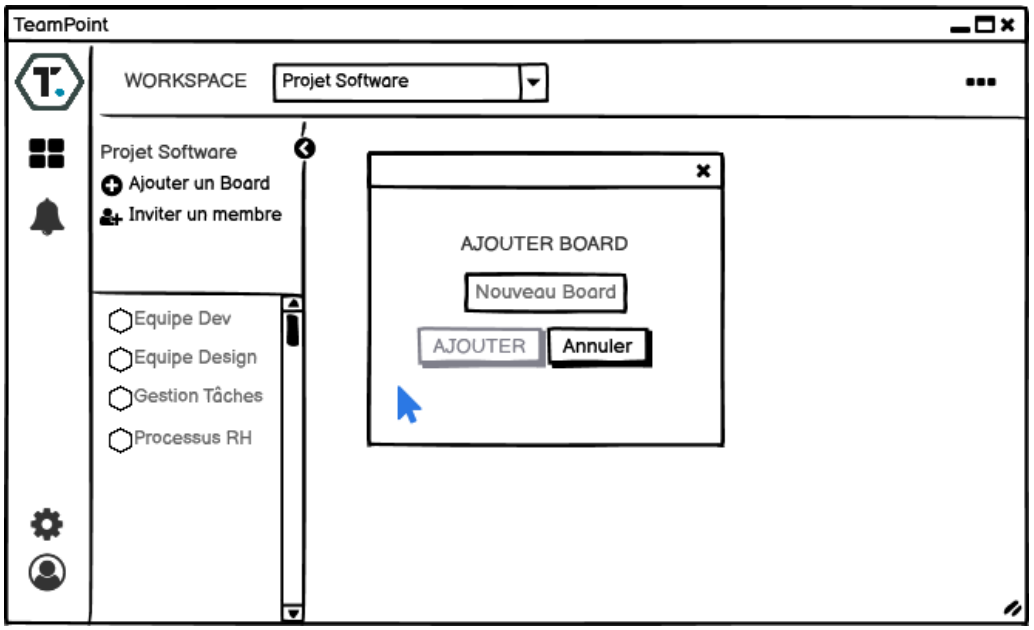
7.5. Post-Conditions

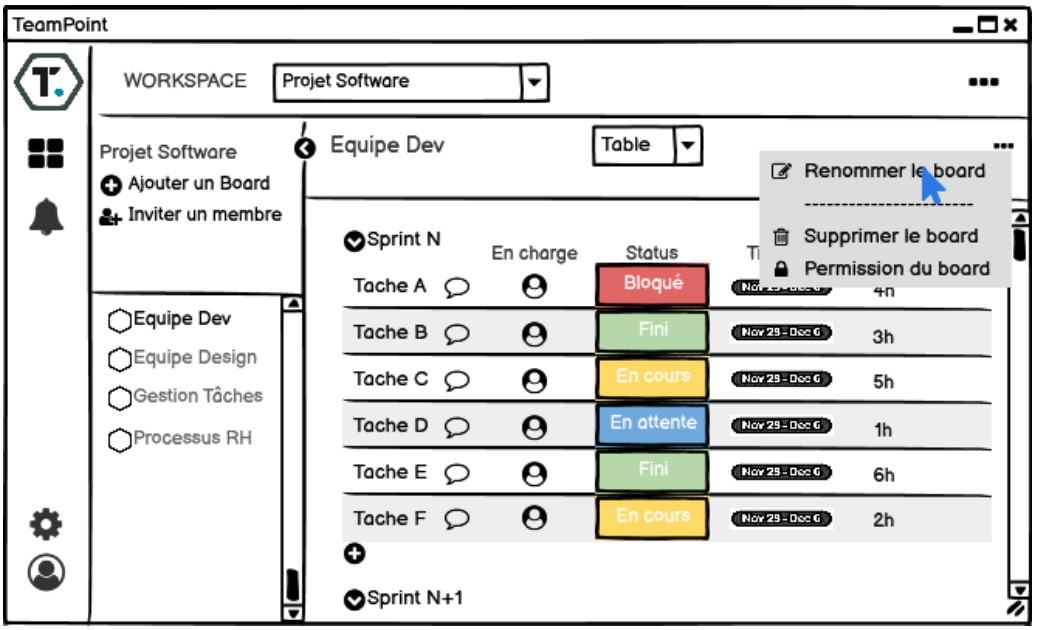
A thread of messages is created.

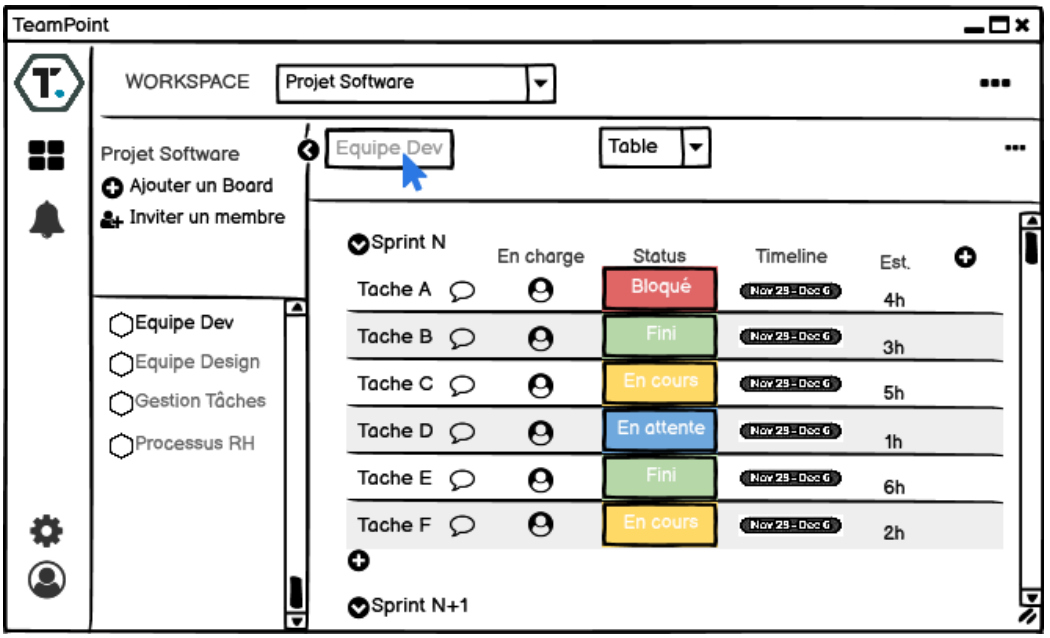
8. Create & Update board

8.1. Mockups









8.2. Brief Description

This use case describes how a member can create and update a board. Any workspace member can become the board owner. If a workspace member creates a board he will become the board owner for this board. Thus, he will be able to set the permission on this board.

8.3. Flow of Events

8.3.1. Basic Flow

The use case starts when a member wants to perform creation or update.

8.3.1.1. Create

1. The workspace member enters the board's name.
2. The system creates the corresponding board.

8.3.1.2. Rename

1. The workspace member enters the new board's name
2. The system automatically replaces the board's name.

8.4. Pre-Conditions

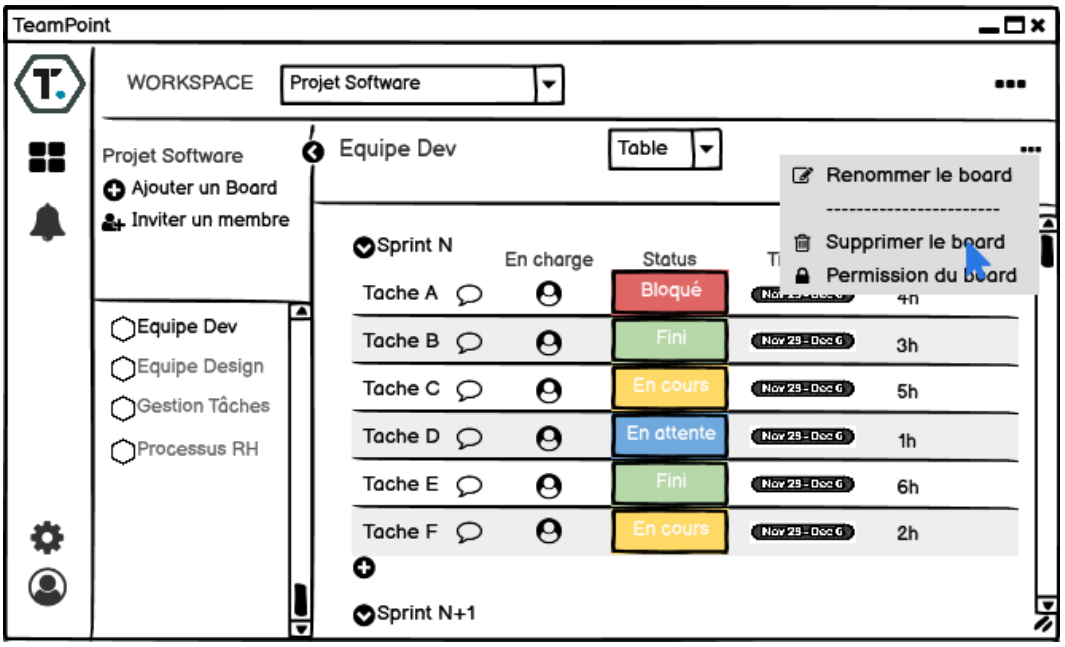
To create a board you must be a workspace member.

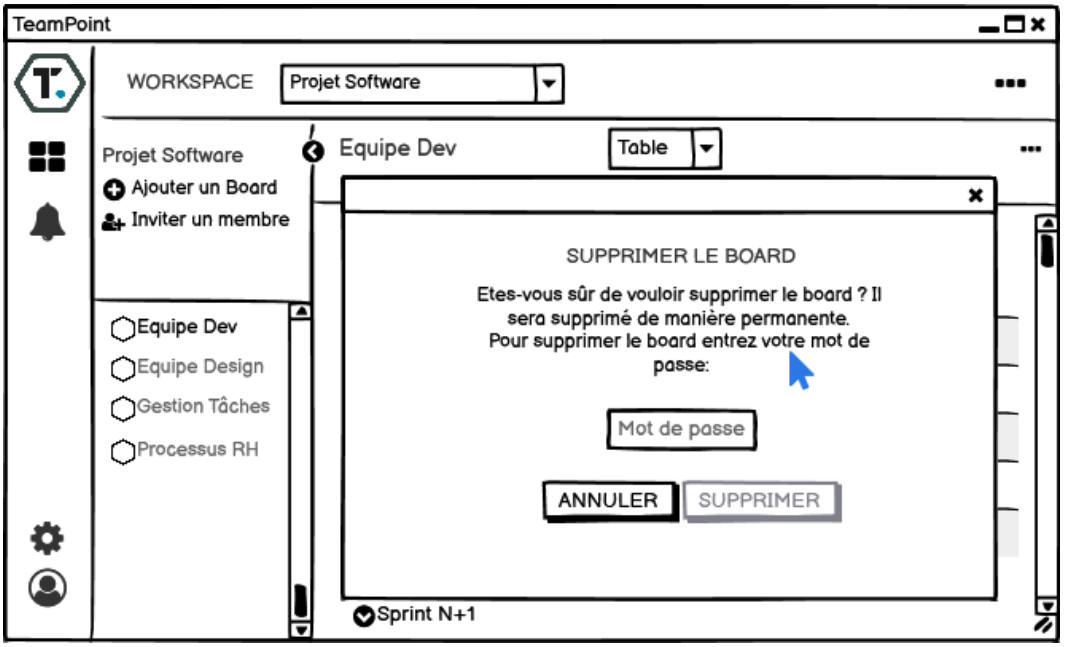
8.5. Post-Conditions

The user who created the board becomes the board owner.
The board created in a certain workspace becomes part of that said workspace.

9. Delete Board

9.1 Mockups





9.2 Brief description

This use case describes how the board owner or the workspace admin can delete. The board owner can only delete his own boards and the workspace admin can delete all the boards in his workspace.

9.3 Flow of Events

The use case starts when the board owner or the workspace admin want to delete a board.

1. The workspace administrator selects a board.
2. The system deletes the board.

9.4. Pre-Conditions

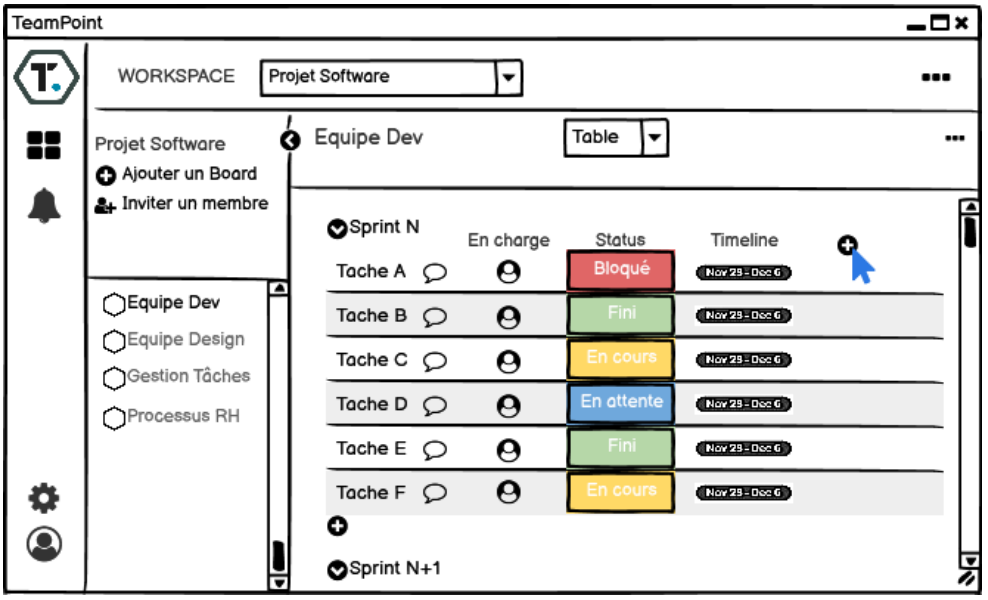
To delete the board you must be the board owner or the workspace administrator.

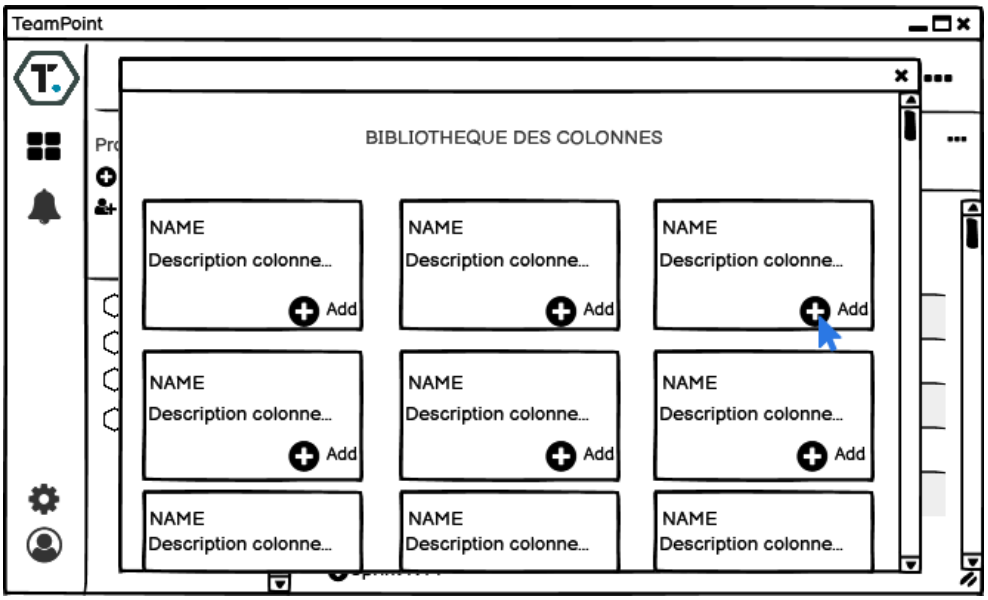
9.5. Post-Conditions

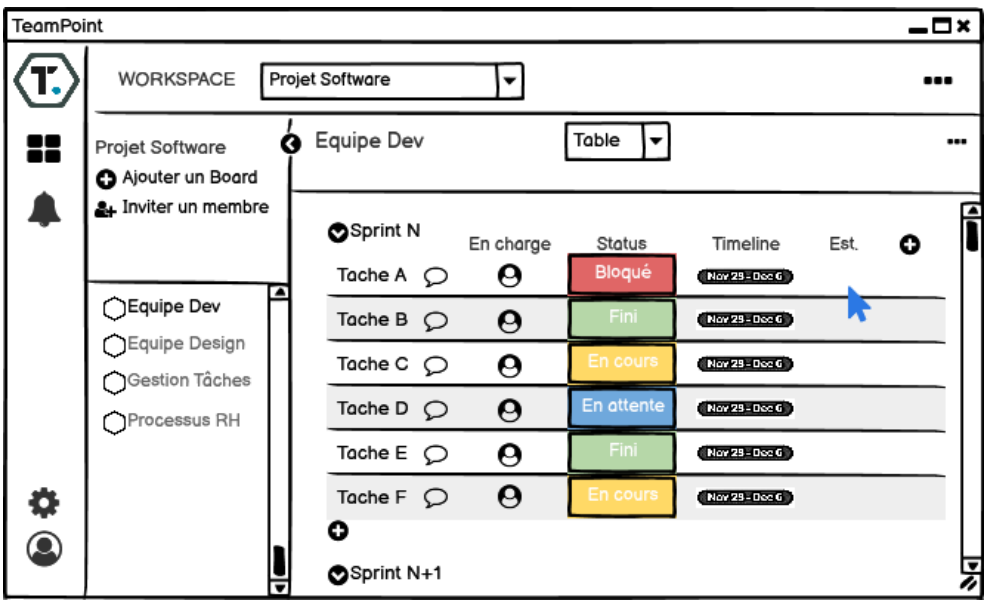
The delete board is no longer available for anyone.

10. Edit board column

10.1. Mockups







10.2. Brief Description

This use case describes how the workspace members can edit board columns. The column's type will be selected between the following types:

- Text : *Takes a string as value.*
- Long text : *large amount of text as value.*
- Status : *The user creates his label and chooses from one of them. The selected label is taken as value.*
- People : *To Assign people. Take one or several as a value.*
- Number : *Takes a number as value. The unit can be specified by the user.*
- Date : *Takes a date as value.*
- Hours : *Takes a number of hours as value.*
- Timeline : *Takes two dates as value*
- Dependency : *Takes one or several board's items as a value.*

10.3. Flow of Events

10.3.1. Basic Flow

10.3.1.1. Add column

This use case starts when the workspace member wants to add a column into a board.

1. The system requires a type of column.
2. The workspace member selects the new column type.
3. The system adds the column in the board.

10.3.1.2. Remove column

This use case starts when the workspace member wants to remove a column into a board.

1. The workspace member selects the column.
2. The system asks for confirmation with a warning.
3. The user confirms the action.
4. The system deletes the column

10.3.2. Invalid permission

If the workspace member does not have the permission to edit the board, the system will display a warning message with instructions to obtain the permission.

10.4. Pre-Conditions

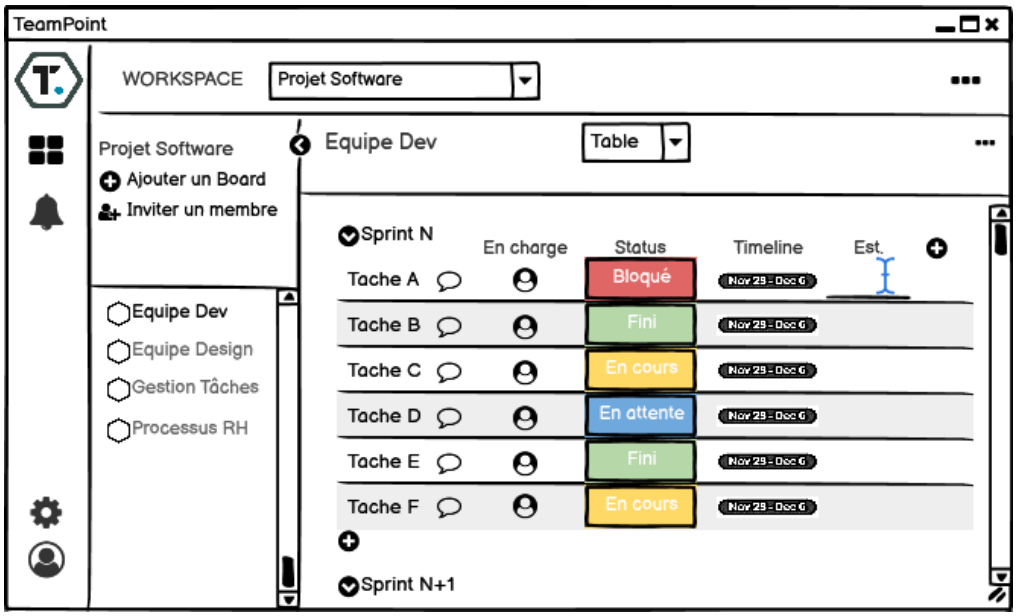
This action has to be allowed. (Permission constraint).

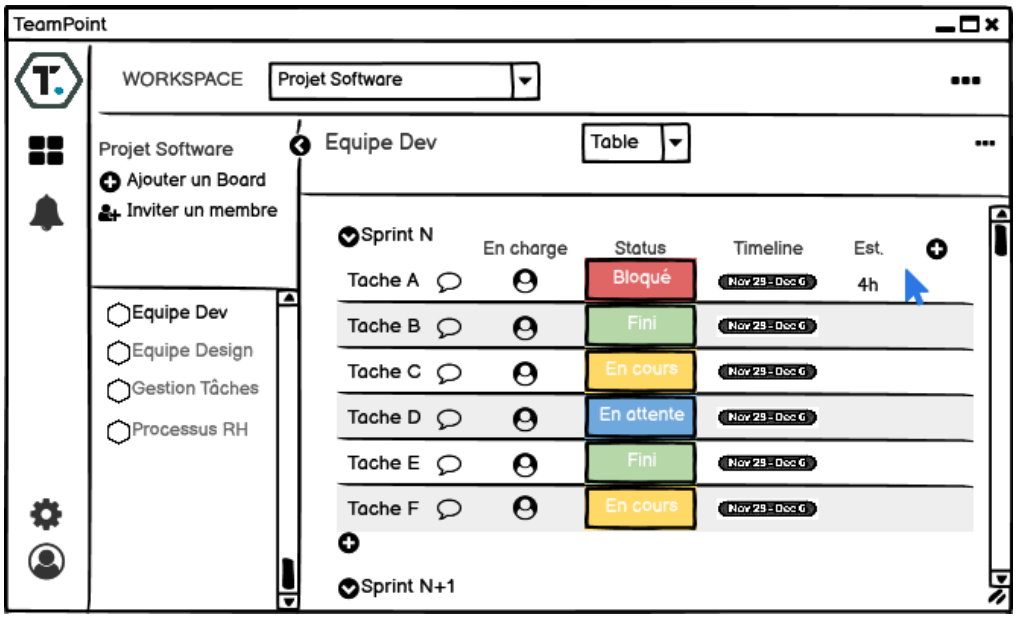
10.5. Post-Conditions

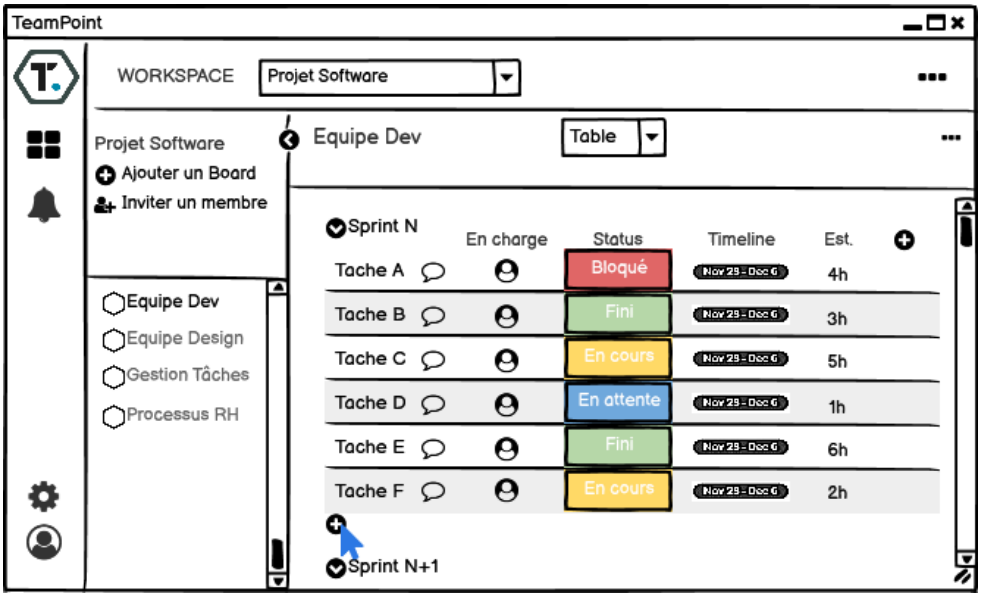
The column is added to the board.

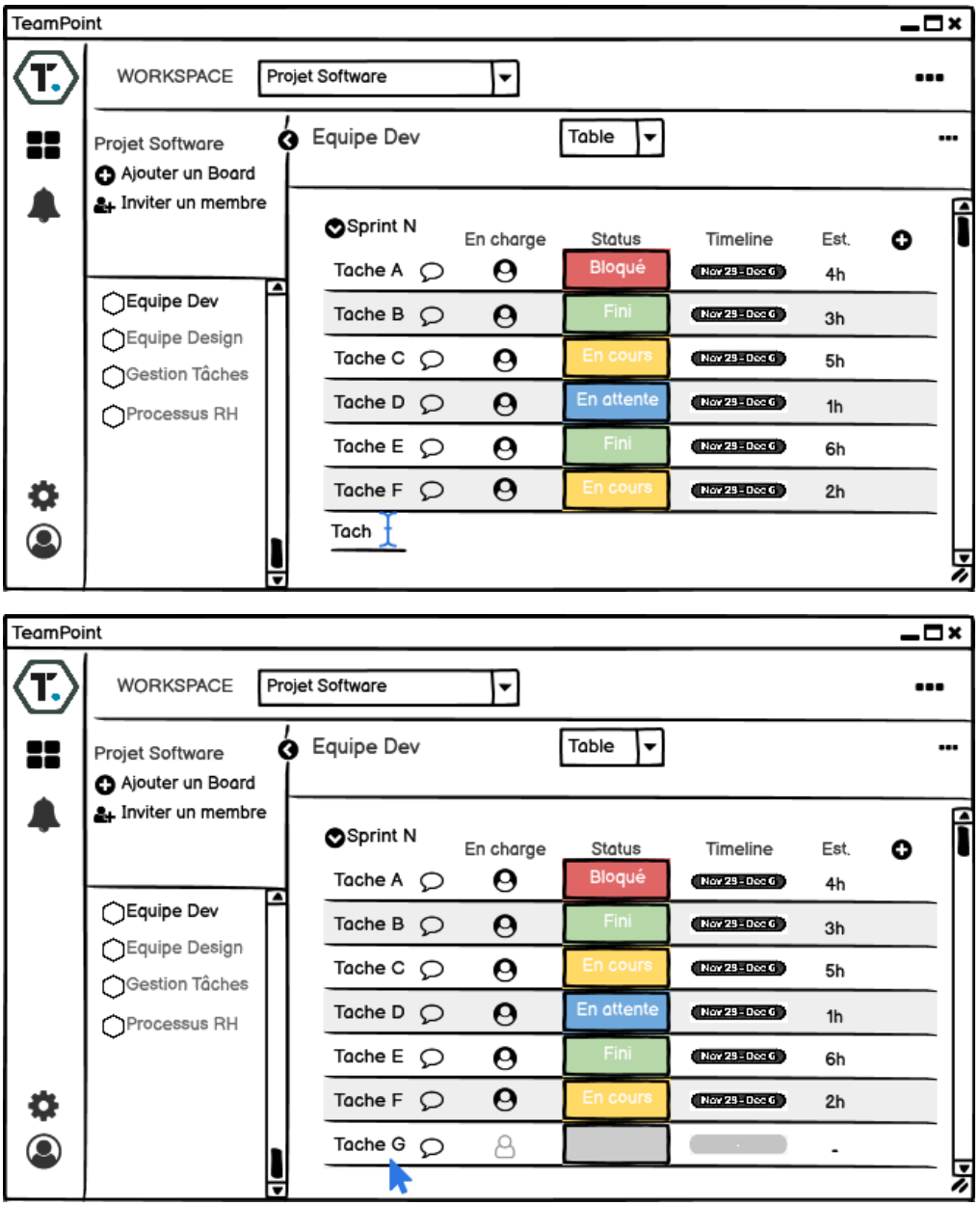
11. Populate the board

11.1. Mockups









11.2. Brief Description

This use case describes how the workspace member can edit the content of the board. The board is composed of items. An item is represented by a row and has a label. The label is the item name. When adding an item to a board the workspace member **must** specify this label in order to create the item and add it to the board. If the board already contains columns their values will be empty for the new added item. The workspace member can then fill the fields (column values) for this item.
The actions a workspace member is allowed to do on a board depend on the permission set by the board owner.

11.3. Flow of Events

11.3.1. Basic Flow

11.3.1.2. Add row

This use case starts when a workspace member wants to add an item into a board.

1. The system requires the new item’s label
2. The workspace member gives the value (a string) for the label.
3. The system creates the item. For the various columns of the board their field values will be empty.
4. The system adds the item to the relevant board.

11.3.1.2. Remove row

This use case starts when the workspace member wants to remove an item from a board.

1. The user selects the item to delete.
2. The system asks if the user is sure about performing this action.
3. The user confirms.
4. The system deletes the item from the board.

11.4. Pre-Conditions

This action has to be allowed by the board owner. (Permission constraint).

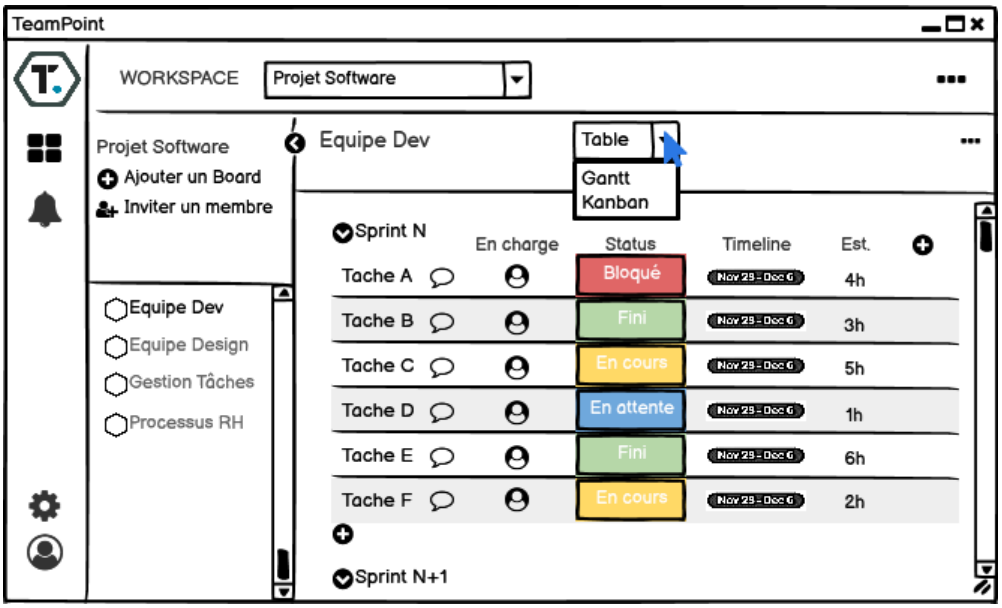
11.5. Post-Conditions

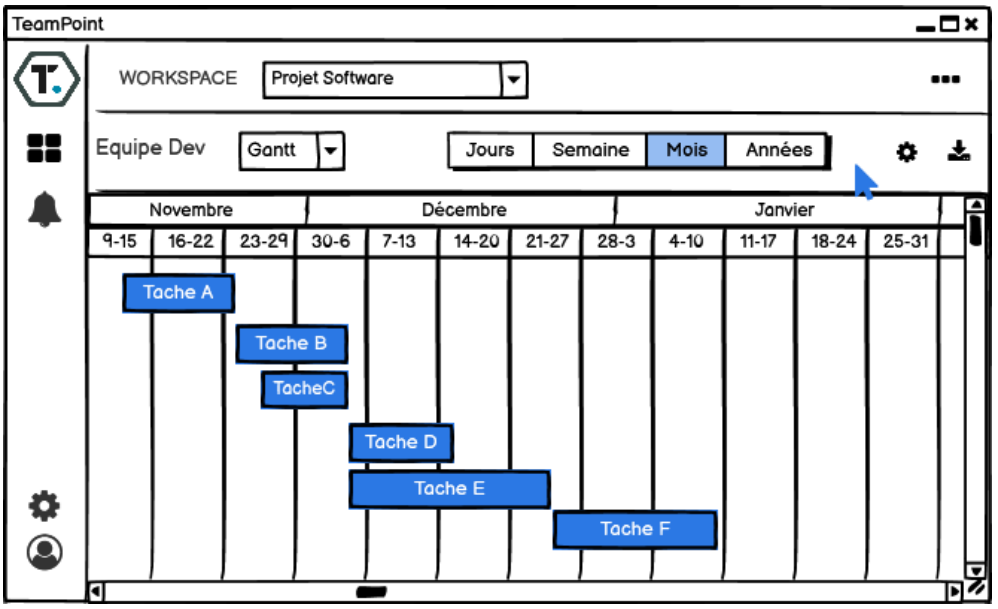
The table is now updated.

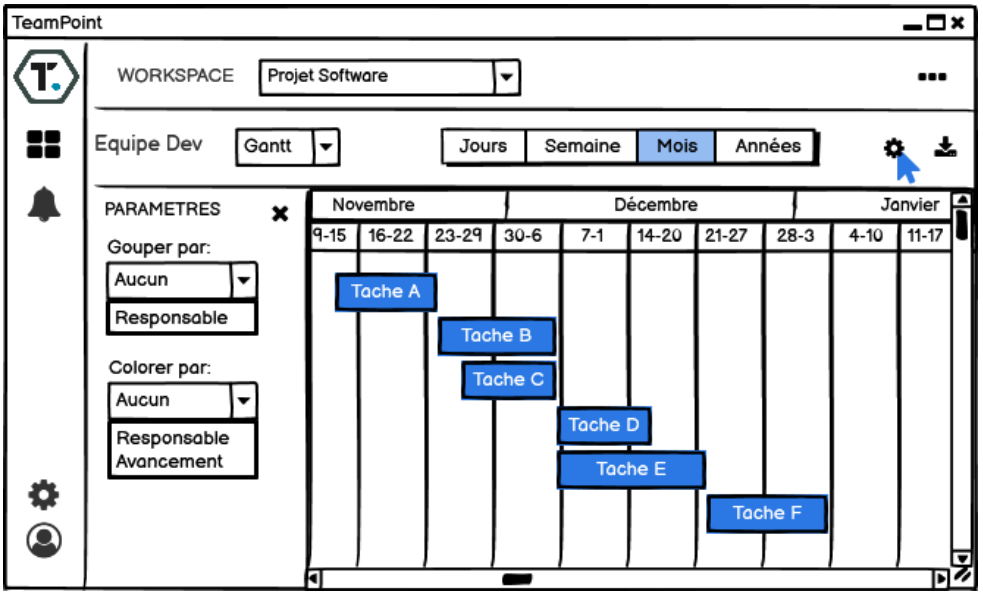
The actor can now fill all the item’s columns.

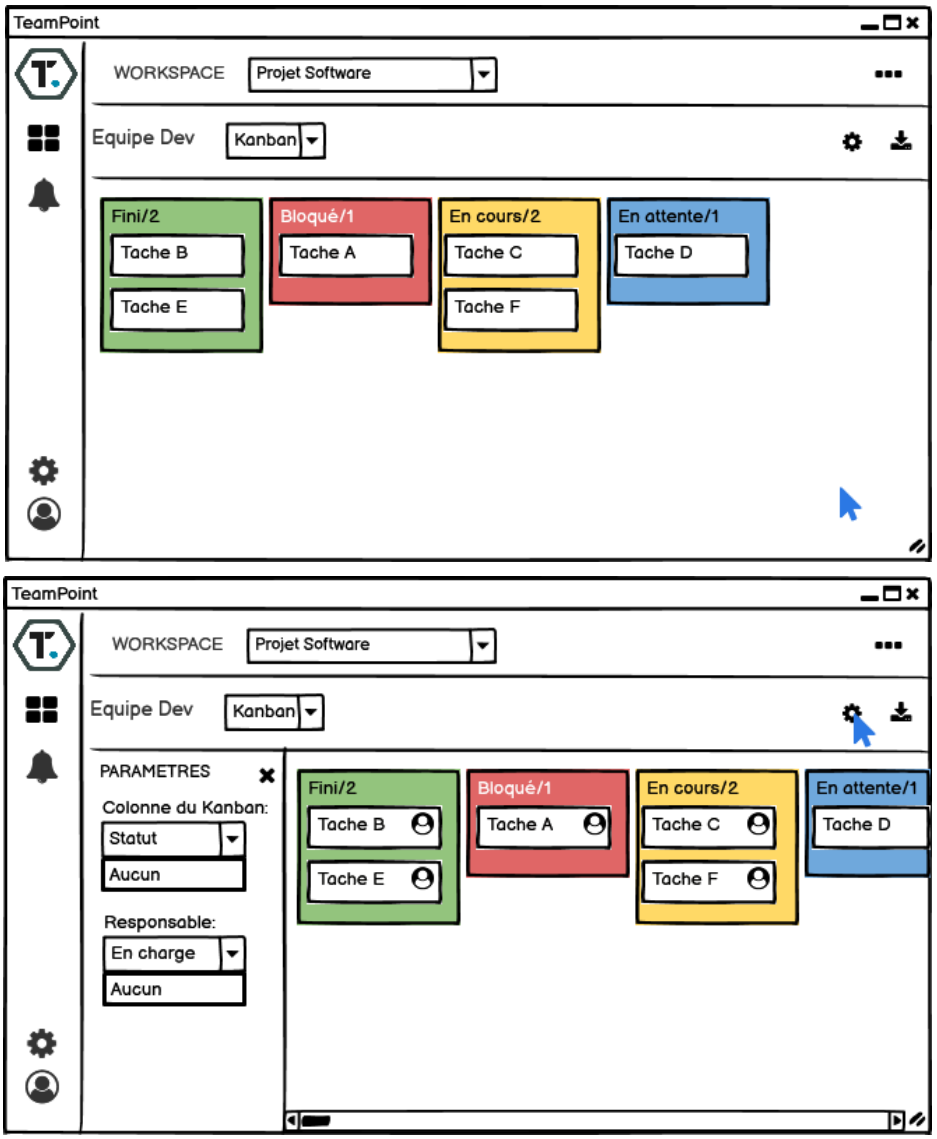
12. Change view of board

12.1. Mockups









12.2. Brief Description

This use case describes the views of a board. The possible views will be Table, Gantt and Kanban.
View generation will be possible with column types which are predefined in this system as explained in Edit board column use cases. The board must be populated with items and columns of a certain type.
The default view will be the table with all the rows.

12.3. Flow of Events

12.3.1. Basic Flow

This use case starts whenever the user wants to change the view from Table to a Kanban or to Gantt.

1. The system shows the list of views

2. The user selects one of them.
3. The system displays the selected view.

12.4. Pre-Conditions

The board must be populated with items and columns of a certain type.

12.4.1 Gantt pre-condition

To generate the Gantt view the board must have a timeline, dependency and status column.

12.4.2 Kanban pre-condition

To generate the Kanban view the board must have a status column.

12.5. Post-Conditions

The selected view is automatically generated.

12.6. Extension Points

12.6.1 Manage Gantt view

This flow describes what users can do in the Gantt view.

- Set the time scale: days, weeks, months, years.
- Group by a Person column to generate multiple sub-gantt related to the selected Person column.
- Color by a Status or a Person column.

12.6.2 Manage Kanban view

This flow describes what users can do in the Kanban view.

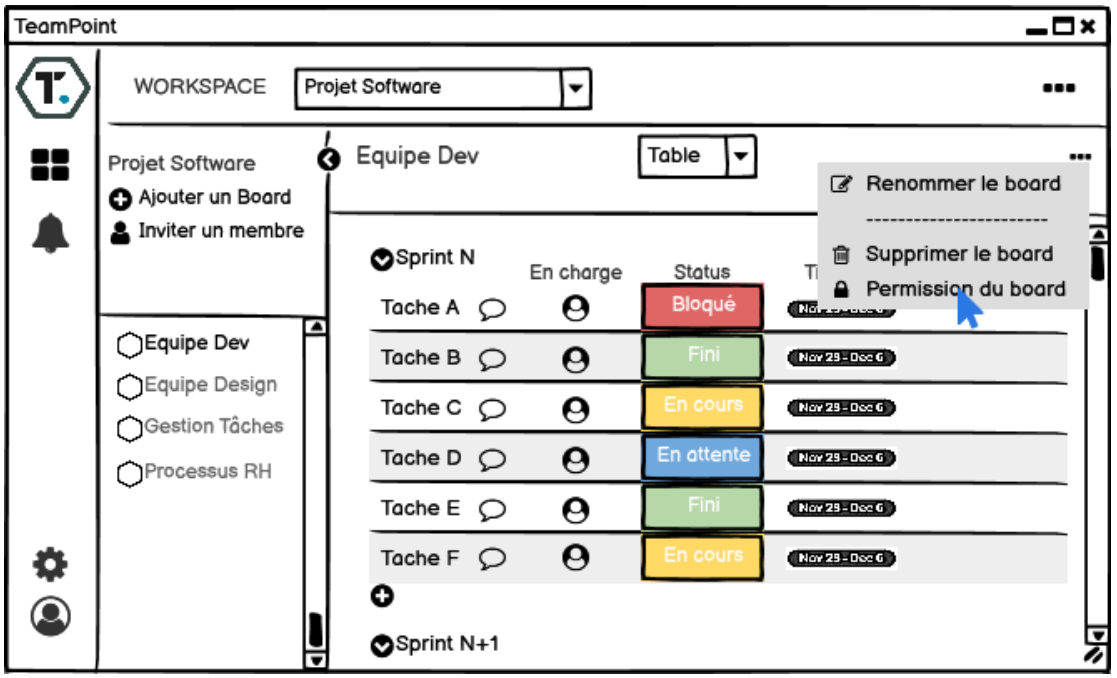
- Set a Status column from which the Kanban columns will be generated. Kanban columns correspond to the label of the status.
- See the responsibles for a task or not.

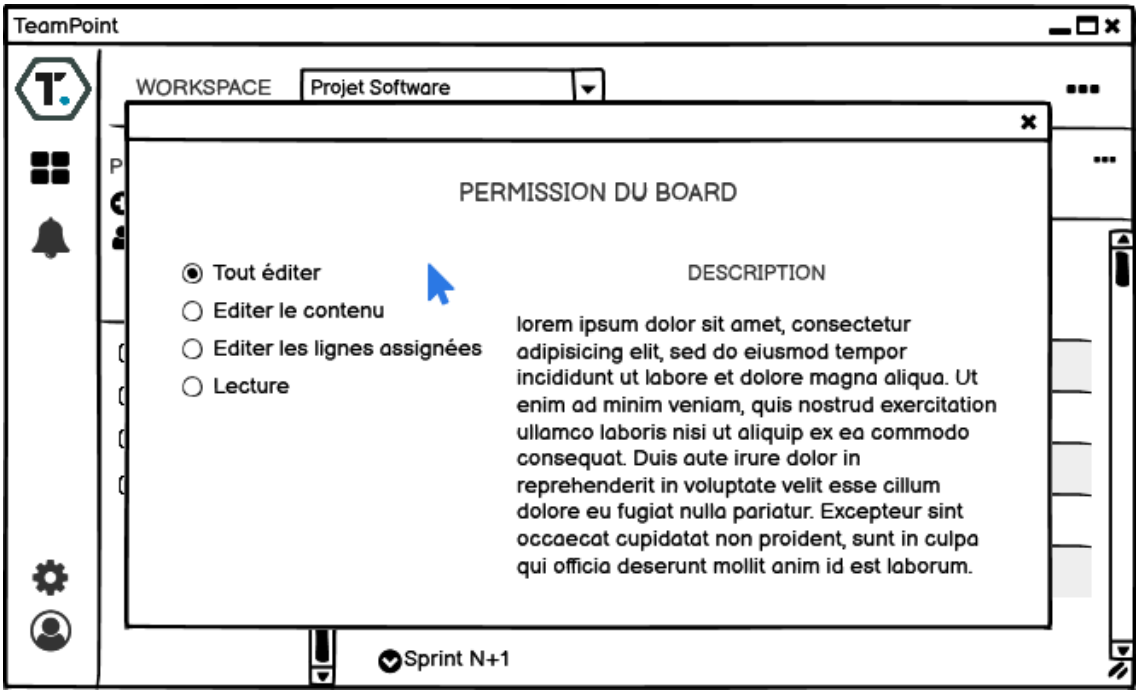
12.6.3. Export chart

The user will be able to export in a few formats like pdf.

13. Manage board permission

13.1. Mockups





13.2. Brief Description

This use case describes how a board owner can manage permissions and what kind of permission he can give. There are four types of permission :

- Edit everything: *you can add/remove board columns and items(rows).*
- Edit content: *you can only add/remove items (rows).*

- Edit assigned rows: *Team members can only edit rows assigned to them in a selected person column.*
- View only: *read the board and communicate.*

The permission concerns a board and is applied for all the workspace members in the same way. The default permission is Edit everything.

13.3. Flow of Events

12.3.1. Basic Flow

12.3.1.1. Set Board permission

This use case starts when the board owner wants to edit board permissions.

1. The system requires a permission constraint.
2. The board owner selects one permission state between all types given above.
3. The system saves the changes.

12.3.1.2. Set Board permission to Edit assigned rows

The system requires a permission constraint.

1. The board owner selects *Edit assigned rows* permission.
2. The system requires the Person column to refer to.
3. The board owner selects one of the board Person columns.
4. The system saves the changes.

13.4. Pre-Conditions

The workspace member must be the board owner to edit board permissions.

To Edit assigned rows permission the board must contain at least one Person column.

13.5. Post-Conditions

Permissions are changed.

Use case analysis organization

The use-case analysis manager's word:

This phase was a real pleasure to achieve. At the beginning of this phase we have started by working together on the use case diagram. It was really interesting to define all the software requirements and functionalities. We have spent a great time discussing and defining our project. Then we have distributed the work in equal charge (Task breakdown tables). So each of us described a few use-cases. After that we have put all the work in common. The next step was to design the mockups. We found this really helpful. Indeed it has helped us to improve our use case description but also our understanding.
All the tools we have used during this phase are detailed in the above section.

Used tools

Use-case analysis :

- We used Google Drive and GitHub to collaborate on the use cases descriptions
- We used Google Docs to collaborate on the final report.
- We used plantTextUml to design the use-case diagram.

Mockups :

- We used the software Balsamiq mockups to create and design our mockups.

Communication:

- Discord
- Messenger

Task breakdown tables

Legend :
UCX : Use case X
MUX : Mockup for Use case X
UC1 & MU1 : Login
UC2 & MU2 : Manage Profile
UC3 & MU3 : Invite user to Workspace
UC4 & MU4 : Delete workspace
UC5 & MU5 : Manage Workspace
UC6 & MU6 : Read board
UC7 & MU7 : Communicate by comments
UC8 & MU8 : Create & Update board
UC9 & MU9 : Delete board
UC10 & MU10 : Edit board column
UC11 & MU11 : Populate the board
UC12 & MU12 : Change view of board
UC13 & MU13 : Manage board permission

Task breakdown table for Use-cases description:

	Salim	Birane	Raphaël	Nicolas
UC1				
UC2				
UC3				
UC4				
UC5				
UC6				
UC7				
UC8				
UC9				
UC10				
UC11				
UC12				
UC13				

Task breakdown table for the mockups:

	Salim	Birane	Raphaël	Nicolas
MU1				
MU2				
MU3				
MU4				
MU5				
MU6				
MU7				
MU8				

	Salim	Birane	Raphaël	Nicolas
MU9				
MU10				
MU11				
MU12				
MU13				