**Use Case Description**

**JS22-1**

|  |  |  |
| --- | --- | --- |
| Use case name: | Invite other users to *Workspace*. | |
| Scenario: | A *Workspace* Admin invite user to a *Workspace*. | |
| Triggering event: | A *Workspace* Admin wants to invite a user into a *Workspace*. | |
| Brief description: | A *Workspace* Admin will either invite a user using link or email | |
| Actors: | *Workspace* Admin | |
| Related use cases: | Might invoked by Adding Workspace  Will invoke Join Workspace and Notify Members | |
| Stakeholders: | *Workspace* Admin, *Workspace* Member, Invited User | |
| Preconditions: | *Workspace* must exist.  Invited user must have a CHello account.  Members who already joined or already invited can’t be invited again.  If invited by link, the link must be validated. | |
| Postconditions: | New member must be added to Workspace and saved.  Notification must be created to propose the invitation to the user.  Notification must be associated with User. | |
| Flow of activities: | Actor | System |
| 1. Admin indicates desire to invite another user to a Workspace.  2. Opens the invite modal view.  3. Admin selects desired user email from fetched options.  4. Admin clicks the invite button.  5. Admin clicks the generate link button. | 2.1 System delivers invite view to Admin.  2.2 System fetches all users data from Chello database.  2.3 System fetches if there is already a generated link.  2.4 System prompts for admin to either invite a user using link or email  4.1 System validates the selected email whether it is already joined or invited.  5.1 System generates a new link if there is no link with a specified timestamp.  5.2 System display the generated link. |
| Exception conditions: | 2.1 If Admin closes the invite modal.  4.1 Invalid invited email. | |

|  |  |  |
| --- | --- | --- |
| Use case name: | Leave Workspace | |
| Scenario: | A *Workspace* Member or Admin leave a workspace | |
| Triggering event: | A *Workspace* Member or Admin wants to leave a *Workspace* | |
| Brief description: | A *Workspace* Member or Admin proceed to *Workspace* settings to leave the corresponding *Workspace* | |
| Actors: | *Workspace* Member or Admin | |
| Related use cases: | Might invoke Close Boards and Delete Workspace  Will invoke Notify Members | |
| Stakeholders: | *Workspace* Admin, *Workspace* Member | |
| Preconditions: | *Workspace* must exist.  User who invoked the leave use case must be Workspace admin or member.  One admin must exists in the Workspace. | |
| Postconditions: | New member must be removed from the Workspace and saved.  Might delete the Workspace if there is no other member | |
| Flow of activities: | Actor | System |
| 1. Workspace Member or Admin indicates desire leave a Workspace.  2. Opens the Workspace settings view.  3. Clicks the leave button | 2.1 System delivers Workspace settings view.  2.2 System fetches members and admins data on the specific Workspace.  3.1 If role is member, remove member from Workspace  3.2 If role is an admin and there are other admins, remove admin from the workspace.  3.3 If there is no other member, delete Workspace and close boards.  3.4 On Success leave navigate user to home |
| Exception conditions: | 3.2 If role is an admin and there is no other admin, display toast to grant admin role to other member. | |

|  |  |  |
| --- | --- | --- |
| Use case name: | Delete Workspace | |
| Scenario: | A *Workspace* Admin delete a workspace | |
| Triggering event: | A *Workspace* Admin wants to delete a *Workspace* | |
| Brief description: | A *Workspace* Admin proceed to *Workspace* settings to delete the corresponding *Workspace* | |
| Actors: | *Workspace* Admin | |
| Related use cases: | Might invoke *Notify Delete Permission to Other Admins*  Will invoke *Notify Members* | |
| Stakeholders: | *Workspace* Admin, *Workspace* Member, Board Admin, Board Member | |
| Preconditions: | *Workspace* must exist.  There must be only one Admin exists | |
| Postconditions: | Workspace will be deleted.  If other Admins accept the permission to delete a workspace then it will also be deleted. | |
| Flow of activities: | Actor | System |
| 1. Workspace Admin indicates desire delete a Workspace.  2. Opens the Workspace settings view.  3. Clicks the delete button | 2.1 System delivers Workspace settings view.  2.2 System fetches members and admins data on the specific Workspace.  3.1 Delete the corresponding Workspace from database.  3.2 Close all boards inside the Workspace  3.4 On Success leave navigate user to home |
| Exception conditions: | 3.1 If there are more than one Admins, ask for permission by sending notifications to other Admins. | |

|  |  |  |
| --- | --- | --- |
| Use case name: | View Workspace Content | |
| Scenario: | A *User* view a Workspace content | |
| Triggering event: | A *User* wants to view a Workspace content | |
| Brief description: | A *User* proceed to *Workspace* page to view corresponding Workspace content | |
| Actors: | *Any User* | |
| Related use cases: | - | |
| Stakeholders: | *Workspace* Admin, *Workspace* Member, Board Admin, Board Member | |
| Preconditions: | *Workspace* must exist.  *Workspace Visibility* must be between *Public* or *Private*  *Workspace Members* can edit and view *Workspace content.*  Non-*Workspace Members* can only view *Workspace content.* | |
| Postconditions: | *Workspace Contents View* will be delivered to User. | |
| Flow of activities: | Actor | System |
| 1. User indicates desire to view a Workspace Contents.  2. Opens the Workspace page view. | 2.1 System delivers Workspace page view.  2.2 System fetches User data and Workspace data.  2.3 Validates User role whether it is Workspace Member or not.  2.3 If user is member display editable *Workspace content.*  2.4 If *Workspace Visibility* is Public and *User* is not a member, display view only page.  2.5 If *Workspace Visibility* is Private and *User* is not a member, redirect to home. |
| Exception conditions: | 2.1 If User doesn’t have a CHello account redirect to login. | |

|  |  |  |
| --- | --- | --- |
| Use case name: | Set Card Due Date | |
| Scenario: | A *Board Member* set a Due Date on a Card | |
| Triggering event: | A *Board Member* wants toset a Due Date on a Card | |
| Brief description: | A *Board Member* proceed to set a Due Date on a Cardto show which tasks must be done by specific date and reminder can also be set to remind board members. | |
| Actors: | *Board Member* | |
| Related use cases: | Involves *Opening Card Detail, Notify Reminder Date* | |
| Stakeholders: | Board Admin, Board Member | |
| Preconditions: | *Card* must exist.  *Workspace Visibility* must be between *Public* or *Private*  *Workspace Members* can edit and view *Workspace content.*  Non-*Workspace Members* can only view *Workspace content.* | |
| Postconditions: | *Workspace Contents View* will be delivered to User. | |
| Flow of activities: | Actor | System |
| 1. *Board Member* indicates desire to set a due date and reminder to card.  2. Opens the Card Detail view.  3. Choose date and time  3.2 Choose reminder type | 2.1 System delivers Card Detail view.  2.2 System fetches Card data and display it.  2.3 Display date and time calendar view and validates *Board Member* can’t choose date and time that have already passed.  3.1 System insert date and time to card details.  3.2 System insert corresponding reminder and calculate the reminder date to remind other members*.*  3.3 Trigger *Notify Reminder Date* use case |
| Exception conditions: | 3.2 If the Board Member choose no reminder then it will not trigger *Notify Reminder Date* use case. | |