

CSE 325
Information and System Design



Assignment-1
Use-case and Test-cases of Notion

Section - A1
Group - 04

Group Members:

1. 2005001 - Anik Saha
2. 2005006 - Kowshik Saha
3. 2005012 - Abrar Jahin Sarker
4. 2005013 - Al Muhit Muhtadi
5. 2005017 - Abdullah Muhammed Amimul Ehsan
6. 2005023 - Jaber Ahmed Deeder

1 Workspace Management

1.1 Use Case: Creation of New Workspace

Scenario: User creates a new workspace from their account.

Actors: User

Steps:

1. User navigates to 'Settings Members' in the sidebar.
2. User selects 'Create or Join Workspace'.
3. User enters the workspace name and email domain, if applicable.
4. User clicks 'Create' to complete the process.

Test Cases

Name	Description
Successful Workspace Creation	Verify that the user can successfully create a new workspace with a unique name.
Duplicate Workspace Name	Verify that creating a workspace with an existing name does not matter
Exceeding Workspace Limit	Verify that the user cannot create more workspaces than their plan allows.

1.2 Use Case: Switching Workspaces

Scenario: User switches between multiple workspaces they are a part of.

Actors: User

Steps:

1. User clicks on their profile picture in the top-left corner.
2. User selects the workspace they want to switch to from the dropdown menu.
3. User is redirected to the selected workspace.

Test Cases

Name	Description
Successful Workspace Switch	Verify that the user can successfully switch to another workspace.
Switch to Unavailable Workspace	Verify that the user cannot switch to a workspace they have left or been removed from.
Switch to Same Workspace	Verify that selecting the current workspace does not reload or cause any issues.

1.3 Use Case: Joining an Existing Workspace

Scenario: User joins an existing workspace by invitation or allowed domain.

Actors: User

Steps:

1. User clicks on the invitation link or logs in with an email matching the allowed domain.
2. User accepts the invitation to join the workspace.

Test Cases

Name	Description
Successful Join via Invitation	Verify that the user can join the workspace successfully using an invitation link.
Join with Invalid Invitation	Verify that the user cannot join the workspace using an expired or invalid invitation link.

Join via Allowed Domain	Verify that the user can join the workspace using an email with an allowed domain.
Join via Restricted Domain	Verify that the user cannot join the workspace using an email with a restricted domain.

1.4 Use Case: Update Workspace Name

Scenario: User updates the name of current workspace

Actors: User

Steps:

1. User enters workspace name

Test Cases

Name	Description
Valid Workspace Name Selection	Verify that the user can successfully update the workspace name with a valid input.
Empty Workspace Name	Verify that the user cannot update the workspace name with an empty input.

1.5 Use Case: Upload Workspace Icon

Scenario: User updates the icon of current workspace

Actors: User

Steps:

1. User selects workspace icon

Test Cases

Name	Description
Valid Workspace Icon Selection	Verify that the user can successfully update the workspace icon with a valid file.
Empty Workspace Icon	Verify that the user cannot update the workspace icon without selecting a file.
Invalid Workspace Icon File Type	Verify that the user cannot update the workspace icon with an invalid file type.
Corrupted Workspace Icon	Verify that the user cannot update the workspace icon with a valid file type but corrupted data.
Large Workspace Icon	Verify that the user cannot update the workspace icon with a file that exceeds the maximum file size.

1.6 Use Case: Delete Workspace

Scenario: User deletes the current workspace

Actors: User

Steps:

1. User types 'Workspace Name' confirms deletion

Test Cases

Name	Description
Correct Workspace Name	Verify that the user can successfully delete the workspace with the correct workspace name.
Incorrect Workspace Name	Verify that the user cannot delete the workspace with an incorrect workspace name.

1.7 Use Case: Leaving a Workspace

Scenario: User leaves a workspace they are a member of.

Actors: User

Steps:

1. User navigates to 'Settings Members' in the sidebar.
2. User clicks on the 'Leave Workspace' button.
3. User confirms the action in the dialog box.

Test Cases

Name	Description
Successful Workspace Leave	Verify that the user can successfully leave a workspace they are a member of.
Leave Non-Member Workspace	Verify that the user cannot leave a workspace they are not a member of.
Cancel Leave Workspace	Verify that the user can cancel the leave action in the confirmation dialog.

1.8 Use Case: Adding Another Account

Scenario: User adds another email account to their Notion account.

Actors: User

Steps:

1. User navigates to 'Settings Members' in the sidebar.
2. User clicks on 'My Account' and selects 'Add Email'.
3. User enters the new email address and verifies it through the email link sent.
4. User confirms the new email as added to their account.

Test Cases

Name	Description
Successful Account Addition	Verify that the user can successfully add another email account to their Notion account.
Invalid Email Format	Verify that the system prevents adding an email with an invalid format.
Email Already Linked	Verify that the system prevents adding an email that is already linked to the account.
Verify Email for Account Addition	Verify that the user must verify the email through the link sent to successfully add the account.

1.9 Use Case: Import Workspace

Scenario: User imports entire workspace data from a file or cloud storage

Actors: User

Steps:

1. User uploads the file to be imported
2. Alternatively, user selects a cloud storage

Test Cases

Name	Description
Valid Workspace File	Verify that the user can successfully import the workspace with a valid file.
Invalid Workspace File	Verify that the user cannot import the workspace with an invalid file.

Valid Workspace Cloud Storage	Verify that the user can successfully import the workspace from a valid cloud storage.
Invalid Workspace Cloud Storage	Verify that the user cannot import the workspace from an invalid cloud storage.

1.10 Use Case: Add Member to Workspace

Scenario: User adds a member to the current workspace

Actors: User

Steps:

1. User searches the email of the member to be added
2. User selects the person to be added or alternatively sends an invitation email

Test Cases

Name	Description
Invalid Email	Verify that the user cannot send an invitation to a member with an invalid email.
Person Already in Workspace	Verify that the user cannot add a person to the workspace who is already a member.
Person Found on Search	Verify that the user can successfully add a person to the workspace.
Person not Found on Search	Verify that the user can successfully send an invitation to a person to join the workspace.
Add Valid Member with Plus Subscription	Verify that the user can successfully add a member with the 'Member' role when the workspace has a Plus subscription.
Add Member Without Plus Subscription	Verify that when the user does not have a Plus subscription, all new members are assigned the 'Owner' role.
Add Member with Invalid Email	Verify that the system handles invalid email addresses during member addition.
Assign Workspace Owner Role	Verify that the user can assign the 'Workspace Owner' role to a new member regardless of subscription.
Invite Using Restricted Domain	Verify that the system prevents adding members with email addresses from restricted domains.

1.11 Use Case: Managing Allowed Email Domains

Scenario: User sets or updates allowed email domains for the workspace.

Actors: User

Steps:

1. User navigates to 'Settings' under the workspace menu.
2. User scrolls to 'Allowed email domains'.
3. User enters the domain name to be allowed (e.g., 'company.com', 'ugrad.cse.buet.ac.bd').
4. User clicks 'Update' to save changes.

Test Cases

Name	Description
Adding Valid Allowed Domain	Verify that the user can add a valid email domain to the allowed list.
Adding Invalid Allowed Domain	Verify that the system handles invalid domain inputs correctly.
Adding Public Domain	Verify that the system handles public domain inputs correctly.
Adding Educational Domain	Verify that the system handles public domain inputs correctly.
Removal Allowed Domain	Verify that the user can remove a previously added domain from the allowed list.

1.12 Use Case: Export Workspace Content

Scenario: User exports all workspace content for backup, sharing, or legal purposes.

Actors: User

Steps:

1. User navigates to 'Settings' under the workspace menu.
2. User clicks on 'Export all workspace content'.
3. User selects export options such as format (Markdown, HTML, CSV, PDF) and content inclusion.
4. User clicks 'Export' to download the content.

Test Cases

Name	Description
Valid Export as Markdown Verify that the user can successfully export all workspace content in Markdown	CSV CSV format.
Valid Export as HTML	Verify that the user can successfully export all workspace content in HTML format.
Valid Export as PDF with Enterprise Plan	Verify that the user can export workspace content as PDF when subscribed to an Enterprise plan.
PDF Export without Enterprise Plan	Verify that PDF export is not available without an Enterprise plan.
Export with Insufficient Permissions	Verify that members without export permissions cannot export workspace content.
Exclude Private Pages from Export	Verify that private pages of other users are not included in the export.
Incorrect Export Format	Verify that the user cannot export the workspace with incorrect workspace export format.

2 Content Management

2.1 Use Case: Rich Text Formatting

Scenario: Users format their text with rich options adding links.

Actors: User

Steps:

1. User selects the text block.
2. User applies the formatting options
3. User checks the formatted result.

Test Cases

Name	Description
Valid Text Hyperlinking	Verify that a user can add a hyperlink to selected text.
Valid Markdown Syntax	Verify that correct markdown content is rendered successfully.
Invalid Markdown Syntax	Verify that incorrect markdown results in failure to apply formatting.
Invalid Text Hyperlinking	Verify that an invalid link is not added to a content text.
Valid font change	Verify that a user can select a valid font of selected text.
Valid Text Coloring	Verify that a user can apply a color to the text.

2.2 Use Case: Organize Content into Columns

Scenario: A user arranges content side-by-side by creating multiple columns on a page.

Actors: User

Steps:

1. User drags and drops blocks into columns.
2. Content is organized into multiple columns.

Test Cases

Name	Description
Create Two Columns	Verify that a user can create two columns by dragging and dropping blocks.
Remove Columns	Verify that a user can remove a column by dragging the content back.

2.3 Use Case: Add Headings for Structure

Scenario: The user adds headings to structure the content on a Notion page.

Actors: User

Steps:

1. User adds H1, H2, and H3 headings for different sections.
2. Content is structured with headings.

Test Cases

Name	Description
Valid Heading Creation	Verify that a user can add an heading to the page.
Header Creation in Incorrect Block	Verify that trying to add a header inside unsupported blocks (e.g., code blocks) fails.

2.4 Use Case: Add Icons and Cover Art

Scenario: A user personalizes a page by adding icons and cover images.

Actors: User

Steps:

1. User adds or changes the page icon.
2. User adds or changes the cover image.

Test Cases

Name	Description
Valid Icon Selection	Verify that a user can successfully add an icon to a page.
Valid Cover Image	Verify that a user can successfully change the cover image on a page.
Invalid Cover Image type	Verify that an image change is rejected if the image type is wrong.

2.5 Use Case: Add and Manage Images in Content

Scenario: A user uploads images, arranges them on a page, and resizes them as needed.

Actors: User

Steps:

1. User uploads an image to the page.
2. User arranges and resizes the image.
3. User adds captions and alt text for the image.

Test Cases

Name	Description
Valid Image Upload	Verify that a user can successfully upload an image to a page.
Valid Image Resize	Verify that a user can resize an uploaded image to a reasonable percentage.
Too Large Resize Percentage	Verify that an image resize is rejected when the entered percentage amount is too large.

2.6 Use Case: Add and Embed Files

Scenario: A user uploads or embeds files like PDFs or documents into a Notion page.

Actors: User

Steps:

1. User uploads a PDF file to the page.
2. User embeds a file link from an external source.
3. User arranges the file block on the page.

Test Cases

Name	Description
Valid PDF Upload	Verify that a user can upload a PDF file to the page.
Too Large Filesize	Verify that a file that is too large is discarded.
Valid External File Embed	Verify that a user can embed a file link from an external URL.
Invalid External File Embed Link	Verify that a file embed is rejected when the link is invalid.

2.7 Use Case: Embed and Manage Media

Scenario: A user embeds videos and audio files into a Notion page.

Actors: User

Steps:

1. User embeds a video from a streaming service.
2. User uploads an audio file.
3. User resizes and aligns the media blocks.

Test Cases

Name	Description
Embed Video	Verify that a user can embed a video from a streaming platform.
Upload Audio File	Verify that a user can upload an audio file and play it in Notion.
Unsupported Media Format	Verify that unsupported audio or video formats fail to play.
Exceeds File Size Limit	Verify that uploading a file that exceeds the size limit fails.

2.8 Use Case: Link to Notion Page

Scenario: A user links one Notion page to another within a paragraph or as a block.

Actors: User

Steps:

1. User types '@', '[' or '+' followed by the page name.
2. User selects the page from the dropdown.
3. Link is added in the desired format.

Test Cases

Name	Description
------	-------------

Valid Page Linking in Paragraph	Verify that a user can link another Notion page inline within a paragraph.
Valid Page linking as Block	Verify that a user can add a Notion page link as a block.
Broken Link Creation	Verify that trying to create a link to a deleted or non-existent page results in an error.

2.9 Use Case: Link to a Web Page

Scenario: A user pastes a URL into Notion and formats it as a mention for easier readability.

Actors: User

Steps:

1. User pastes the URL into a Notion page.
2. User selects 'Paste as mention' to format the link.

Test Cases

Name	Description
Valid Web Link Embedding	Verify that a user can embed a web link and format it as a mention.
Broken Web Link	Verify that a broken link is detected by the system and rejected.

2.10 Use Case: Transfer Content to Another Account

Scenario: A user transfers ownership of content (pages, databases) from one Notion account to another.

Actors: User, New Account Owner

Steps:

1. User shares the page with the new account.
2. User grants ownership permissions to the new account.

Test Cases

Name	Description
Valid Ownership Transfer	Verify that a user can transfer ownership of a page to another account.
Transfer Between Incompatible Accounts	Verify that transferring content between a free plan and a paid account fails if file size limits are exceeded.

2.11 Use Case: Duplicate and Delete Content

Scenario: A user duplicates or deletes content on a Notion page.

Actors: User

Steps:

1. User selects a block of content.
2. User chooses either 'Duplicate' or 'Delete' from the menu.

Test Cases

Name	Description
Duplicate Block	Verify that a user can duplicate a block of content.
Delete Page	Verify that a user can delete an entire page.

2.12 Use Case: Restore Deleted Content

Scenario: A user restores deleted content from the trash within 30 days.

Actors: User

Steps:

1. User navigates to the trash.

2. User selects the deleted page to restore.

Test Cases

Name	Description
Restore Deleted Page	Verify that a user can restore a deleted page from the trash.
Restore Deleted Page After Time-out	Verify that restoring a deleted page fails after the allowed restoration time has passed.

2.13 Use Case: Sync Content Across Pages

Scenario: A user syncs content across multiple Notion pages.

Actors: User

Steps:

1. User copies the content from a page.
2. User pastes it on a different page with the 'Paste and sync' option.
3. Content is now synced across pages.

Test Cases

Name	Description
Sync Content Successfully	Verify that the user can sync content across two pages.
Sync Content Failure - No Permissions	Verify that a user cannot sync content to a page where they don't have permission.

2.14 Use Case: Unsync Content

Scenario: A user unsyncs specific content across pages.

Actors: User

Steps:

1. User selects the synced block.
2. User chooses the 'Unsync' option to remove the sync.

Test Cases

Name	Description
Unsync Specific Block	Verify that a user can unsync a specific copy of the block.
Unsync Failure - Not Original Block	Verify that unsyncing fails when attempting to unsync a block that is not the original.

2.15 Use Case: Create and Edit Code Blocks

Scenario: User creates and edits code blocks within Notion for documentation or sharing code snippets.

Actors: User

Steps:

1. User navigates to a Notion page and selects the option to add a new block.
2. User selects the code block option from the available block types.
3. User types or pastes code into the code block.
4. User modifies the code block by selecting the language, formatting, or editing the code.

Test Cases

Name	Description
Valid Code Block Creation	Verify that a user can successfully create a code block and input code.

Valid Language Selection	Verify that a user can select a programming language for a code block.
Invalid Language Selection	Verify that a user can select a programming language for a code block.
Empty Code Block	Verify that the system handles an empty code block without any code input.
Valid Code Block Editing	Verify that a user can successfully edit an existing code block.

2.16 Use Case: Create and Display Math Equations in Notion

Scenario: A user creates and formats math equations in Notion using LaTeX syntax.

Actors: User

Steps:

1. User types an equation using the inline math command ‘

’or the block equation command’ /math’. The system processes the LaTeX syntax.

2. The system displays the equation in the correct format.

Test Cases

Name	Description
Valid Inline Equation	Verify that a valid inline math equation is rendered correctly.
Invalid Inline Equation	Verify that the system handles improper LaTeX syntax in inline equations.
Valid Block Equation	Verify that a block equation is correctly rendered in Notion.
Invalid Block Equation	Verify that the system handles incorrect LaTeX syntax in block equations.

3 Database Creation and Management

3.1 Use Case: Creating a Database

Scenario: User creates a new database in Notion.

Actors: User

Steps:

1. User navigates to a Notion page and types ‘/’ to access the command menu.
2. User selects a database type (e.g., table, board, list).
3. User enters the name of the new database.
4. User configures properties for the database.

Test Cases

Name	Description
Creating a Table Database	Verify that the user can create a new table database.
Creating a Database Without Name	Verify that the user cannot create a database without providing a name.
Creating a Database with Existing Name	Verify that the system handles duplicate database names within the same workspace.

3.2 Use Case: Customizing Database Views

Scenario: User customizes the views of a database to display data in different formats.

Actors: User

Steps:

1. User navigates to the database and selects ‘Add a view’.

2. User chooses a view type such as 'Table', 'Board', 'Calendar', etc.
3. User customizes the view by adding filters, sorts, and grouping.
4. User saves the view for future use.

Test Cases

Name	Description
Creating a Kanban Board View	Verify that the user can create a board view to display tasks by status.
Applying Filters to Calendar View	Verify that the user can apply filters to show only events for a specific team in the calendar view.
Sorting Table View by Priority	Verify that the user can sort tasks by priority in the table view.

3.3 Use Case: Managing Database Pages

Scenario: User manages individual pages within a database, adding, opening, and editing them.

Actors: User

Steps:

1. User navigates to the database and clicks on 'New' to add a new page.
2. User enters the details for the new page and saves it.
3. User opens an existing page to edit its content.
4. User modifies properties or adds new content to the page.

Test Cases

Name	Description
Creating a New Database Page	Verify that the user can create a new page within the database.
Editing Existing Page Content	Verify that the user can edit an existing page's content in the database.
Deleting a Database Page	Verify that the user can delete a page from the database.

3.4 Use Case: Collaborating in a Database

Scenario: User collaborates with team members in a shared database with specific permissions.

Actors: User

Steps:

1. User shares the database with team members with 'Can edit content' permissions.
2. Team members add and edit pages within the database.
3. User reviews changes made by team members and provides feedback in comments.

Test Cases

Name	Description
Granting Edit Permissions	Verify that the user can grant 'Can edit content' permissions to team members.
Editing Content with Edit Permissions	Verify that team members with 'Can edit content' permissions can add and edit pages in the database.
Restricting Property Editing	Verify that team members cannot edit properties or views with 'Can edit content' permissions.

3.5 Use Case: Managing Database Lock

Scenario: User locks and unlocks a database to control structure changes.

Actors: User

Steps:

1. User navigates to the database settings.
2. User selects 'Lock/Unlock database' and confirms the action.

Test Cases

Name	Description
Lock and Unlock Database	Verify that the user can lock and unlock a database, preventing or allowing structural changes.
Lock Database Without Permission	Verify that users without permissions cannot lock/unlock the database.

3.6 Use Case: Handling Changes in a Locked Database

Scenario: User attempts to change structure of a locked database.

Actors: User

Steps:

1. User tries to modify properties or views of the locked database.

Test Cases

Name	Description
Edit Locked Database Properties or Views	Verify that users cannot modify properties, add views, or delete properties in a locked database.

3.7 Use Case: Managing Database Properties

Scenario: User adds, edits, and deletes properties in a database.

Actors: User

Steps:

1. User navigates to database settings.
2. User selects 'New property' to add a property.
3. User edits or deletes properties as needed.

Test Cases

Name	Description
Adding a New Property	Verify that the user can add a new property with a valid name and type.
Adding a Property with Empty Name	Verify that the user cannot add a property with an empty name.
Deleting a Property	Verify that the user can delete a property.

3.8 Use Case: Editing Property Values

Scenario: User edits values for different property types in the database.

Actors: User

Steps:

1. User selects a property cell in the database.
2. User enters or updates the value for the property.
3. User saves the changes.

Test Cases

Name	Description
Editing Text Property	Verify that the user can enter text into a text property.
Entering Invalid Number in Number Property	Verify that the user cannot enter non-numeric values in a number property.
Clearing a Date Property	Verify that the user can clear the value in a date property.

3.9 Use Case: Handling Invalid and Empty Values for Properties

Scenario: User enters invalid or empty values for different property types in a database.

Actors: User

Steps:

1. User selects a property cell in the database.
2. User attempts to enter invalid or empty values for each property type.
3. System validates and either accepts or rejects the input.

Test Cases

Name	Description
Entering Invalid Number	Verify that the system rejects non-numeric values for a number property.
Empty Number Property	Verify that the system accepts an empty value for a number property.
Invalid Email Format	Verify that the system rejects an invalid email format.
Empty Email Property	Verify that the system accepts an empty value for an email property.
Invalid URL Format	Verify that the system rejects an invalid URL format.
Empty URL Property	Verify that the system accepts an empty value for a URL property.
Invalid Date Format	Verify that the system rejects an invalid date format.
Empty Date Property	Verify that the system accepts an empty value for a date property.
Invalid Phone Format	Verify that the system rejects an invalid phone number format.
Empty Phone Property	Verify that the system accepts an empty value for a phone property.

3.10 Use Case: Creating and Managing Relations

Scenario: User creates and manages relations between databases to link related data.

Actors: User

Steps:

1. User navigates to the database settings.
2. User adds a new 'Relation' property.
3. User selects the target database to link.
4. User configures relation settings and saves.

Test Cases

Name	Description
Creating a One-Way Relation	Verify that the user can create a one-way relation between two databases.
Creating a Two-Way Relation	Verify that the user can create a two-way relation between two databases.
Relating a Database to Itself	Verify that the user can create a relation within the same database to link related items.

3.11 Use Case: Creating and Managing Rollups

Scenario: User creates rollup properties to aggregate data from related databases.

Actors: User

Steps:

1. User navigates to the database settings.
2. User adds a new 'Rollup' property.
3. User selects the related property to roll up and chooses a calculation method.
4. User saves the rollup property settings.

Test Cases

Name	Description
Creating a Sum Rollup	Verify that the user can create a rollup property to sum numeric values from related items.
Creating a Count Rollup	Verify that the user can create a rollup property to count non-empty values from related items.
Creating a Date Rollup	Verify that the user can create a rollup property to find the latest date from related items.

3.12 Use Case: Handling Invalid or Empty Rollup Values

Scenario: User tries to create rollup properties with invalid or empty values.

Actors: User

Steps:

1. User attempts to create a rollup with invalid data types or empty values.

Test Cases

Name	Description
Invalid Data Type for Rollup	Verify that the system rejects rollup creation if the selected property is not compatible with the calculation.
Empty Values in Rollup Property	Verify that the system can handle rollup calculations with empty values correctly.

3.13 Use Case: Managing Database Views

Scenario: User creates, customizes, and switches between different views in a database.

Actors: User

Steps:

1. User navigates to the database and selects 'Add a view'.
2. User chooses a view type such as 'Table', 'Board', 'Calendar', etc.
3. User customizes the view settings (properties, filters, sorts).
4. User switches between different views using the dropdown menu.

Test Cases

Name	Description
Creating a New View	Verify that the user can create a new view with customized settings.
Switching Between Views	Verify that the user can switch between different database views without errors.
Editing View Components	Verify that the user can edit existing view components such as layout and properties.

3.14 Use Case: Applying Filters to a Database

Scenario: User applies filters to a database view to display specific data.

Actors: User

Steps:

1. User navigates to the database and clicks 'Filter'.
2. User selects a property and sets criteria for filtering.
3. User adds additional filters or groups using 'AND'/'OR' logic.
4. User saves the filter configuration.

Test Cases

Name	Description
Applying a Simple Filter	Verify that the user can apply a simple filter based on a single property.
Creating an Advanced Filter	Verify that the user can create an advanced filter using 'AND'/'OR' logic.
Removing a Filter	Verify that the user can remove an applied filter from the database view.

3.15 Use Case: Sorting Items in a Database

Scenario: User sorts database items based on a specific property in ascending or descending order.

Actors: User

Steps:

1. User navigates to the database and clicks 'Sort'.
2. User selects a property and chooses ascending or descending order.
3. User adds multiple sorts if needed and adjusts their order.
4. User saves the sort configuration.

Test Cases

Name	Description
Sorting by Due Date	Verify that the user can sort items by 'Due Date' in ascending order.
Sorting by Priority and Status	Verify that the user can sort items by 'Priority' and 'Status' with different orders.
Removing a Sort	Verify that the user can remove an applied sort from the database view.

3.16 Use Case: Grouping Items in a Database

Scenario: User groups database items by a specific property to organize data visually.

Actors: User

Steps:

1. User navigates to the database and clicks 'Group'.
2. User selects a property to group by (e.g., 'Status').
3. User arranges the groups in a desired order.
4. User saves the group configuration.

Test Cases

Name	Description
Grouping by Status	Verify that the user can group items by 'Status' in a board view.

Hiding Empty Groups	Verify that the user can hide groups without items in the database view.
Removing Grouping	Verify that the user can remove grouping from the database view.

3.17 Use Case: Searching a Database

Scenario: User searches for specific items within a database using keywords.

Actors: User

Steps:

1. User navigates to the database and clicks the search icon.
2. User enters keywords related to page titles or properties.
3. User reviews the search results displayed in real-time.

Test Cases

Name	Description
Searching by Keyword	Verify that the user can search for items using a keyword.
No Matching Results	Verify that no items are displayed when there are no matching results.
Clearing Search	Verify that the user can clear the search input to return to the default view.

3.18 Use Case: Managing Data in Table View

Scenario: User creates, modifies, and deletes rows and columns in a table view.

Actors: User

Steps:

1. User navigates to the database table view.
2. User adds, edits, or deletes rows and columns as needed.

Test Cases

Name	Description
Adding Rows	Verify that the user can add a new row to the table with all columns populated.
Deleting Columns	Verify that the user can delete a column and all associated data is removed.
Empty Values	Verify that leaving a cell empty in a required column triggers a validation message.

3.19 Use Case: Organizing Simple Data in List View

Scenario: User organizes notes or articles using list view with minimal properties.

Actors: User

Steps:

1. User navigates to the database list view.
2. User adds, edits, or deletes list items as needed.

Test Cases

Name	Description
Adding Items	Verify that the user can add a new item with minimal properties like title and date.
Invalid Date	Verify that entering an invalid date format shows an error message.
Editing Item Title	Verify that the user can successfully edit the title of a list item.

3.20 Use Case: Visualizing Workflows in Board View

Scenario: User creates boards to manage project stages or tasks.

Actors: User

Steps:

1. User navigates to the database board view.
2. User moves items between columns to represent changes in workflow stages.

Test Cases

Name	Description
Moving Items Across Stages	Verify that the user can drag and drop items between different stages (columns).
Invalid Stage Name	Verify that the system rejects a stage name with special characters.
Archiving Columns	Verify that the user can hide or archive columns with completed tasks.

3.21 Use Case: Managing Events in Calendar View

Scenario: User schedules and views events on a calendar.

Actors: User

Steps:

1. User navigates to the database calendar view.
2. User adds, edits, or deletes events as needed.

Test Cases

Name	Description
Adding Multi-Day Events	Verify that the user can add events that span multiple days.
Invalid Date Range	Verify that entering a start date later than the end date triggers an error.
Changing Calendar Start Day	Verify that changing the week start day updates the view correctly.

3.22 Use Case: Assigning Unique Identifiers to Database Items

Scenario: User assigns unique identifiers to each item for easy reference.

Actors: User

Steps:

1. User navigates to the database settings.
2. User configures or updates the unique ID format for each new item.

Test Cases

Name	Description
Generating Unique ID	Verify that each new item in the database is assigned a unique identifier automatically.
Custom ID Format	Verify that the user can set a custom format for unique IDs (e.g., prefix or suffix).
Duplicate ID Error	Verify that trying to manually set a duplicate ID triggers an error.

4 Sharing and Collaboration

4.1 Use Case: Create New Teamspace

Scenario: A user creates a teamspace and invites other members by mentioning their emails. Those who accept the invitations become members of the teamspace.

Actors: User

Steps:

1. User provides name of the new Teamspace
2. User provides icon of the new Teamspace
3. User provides emails of other invited members

Test Cases

Name	Description
Valid Teamspace Creation	Verify that a user can successfully create a teamspace when he provides valid information.
Empty Teamspace Name	Verify that the system handles the case where a user provides empty teamspace name.
Invalid Icon Type	Verify that the system rejects unsupported file types for the teamspace icon.
Too Many Invited Emails	Verify that the system prevents a user under Free plan from inviting more than 10 guests in a single teamspace creation.
Invalid Email Format	Verify that the system detects and rejects invalid email formats.
Invitation to an Email without Notion	Verify that the system detects when the given email address has no associated Notion account.

4.2 Use Case: Manage Teamspace Permissions

Scenario: A teamspace owner may change the permission levels of any other owner or member. A member may only change the permission levels of other members.

Actors: Teamspace Owner, Teamspace Member

Steps:

1. Teamspace owner changes the permission level of other members to one of 'Full Access', 'Can View', 'Can Edit' or 'Can Comment'
2. Alternatively, Teamspace owner upgrades another member to owner.
3. Alternatively, Teamspace owner downgrades another owner to member.

Test Cases

Name	Description
Consistent Permissions	Verify that consistent teamspace permissions can be set properly.
Invalid Permission Level	Verify that the permission level falls under one of the 4 categories.
Permission Level Update by Member	Verify that a general member cannot alter the permission levels of all members.
Downgrade of Owner by Member	Verify that a general member cannot downgrade a owner to a general member.

4.3 Use Case: Share Page with Individuals

Scenario: A user shares his page with other members or groups while specifying the access levels of each.

Actors: User

Steps:

1. User selects a page from his workspace.
2. User selects names of members or groups to share with.
3. User sets permission levels of each of the members or groups.

Test Cases

Name	Description
Valid Sharing	Verify that the page is shared properly when invites members and groups are valid.
Sharing with only Members	Verify that the page sharing is successful when only members are added and no groups are added.

Sharing with only Groups	Verify that the page sharing is successful when only groups are added and no individual members are added.
Empty Sharing Specification	Verify that the page sharing is rejected when no members or groups are selected.
Invalid Permission Category	Verify that the page sharing is rejected when the chosen permission category does not fall under the allowed 4 types of permissions.

4.4 Use Case: Publish Page as Website

Scenario: A user publishes a page as a website that if publicly available. The user selects domain name, SEO and link settings based on his subscription plan.

Actors: User

Steps:

1. User opens a page from his workspace.
2. User proceeds to publish as Website option.
3. User approves the proposed randomly generated website domain.
4. Alternatively, the user may choose his own domain if he is in a Plus subscription
5. The user chooses SEO and Link Expiration related preferences and finalizes publishing.

Test Cases

Name	Description
Valid Publishing	Verify that the website is created properly when the user follows a valid process.
Choosing Custom Domain in Free Plan	Verify that the cannot choose a custom domain when he is in a Free plan.
Choosing Custom Domain in Plus Plan	Verify that the can choose a custom domain when he is in a Plus subscription plan.
Specifying Link Expiry Period in Free Plan	Verify that the cannot specify a link expiry period when he is in a Free plan.

4.5 Use Case: Collaborate in real-time

Scenario: One or more than one user collaborates on a page content in real-time. The system approves or rejects the edits based on consistency.

Actors: User

Steps:

1. User opens the shared page.
2. User selects a particular block of the page.
3. User edits the content of the selected block.

Test Cases

Name	Description
Valid Editing	Verify that the editing is successful when all editors have enough privilege and they are not simultaneously modifying the same block.
Insufficient Permission	Verify that the editing is discarded when some editor has insufficient access privilege.
Simultaneous Editing	Verify that the editing is discarded when more than one editor are trying to simultaneously modify the same block.

4.6 Use Case: Add comments to Page Content

Scenario: On the comment portion of a page block, user writes a text and/or mentions another persons/guests/pages.

Actors: User

Steps:

1. User selects a particular block of the page.
2. User writes a textual comment.
3. Alternatively, user may mention other persons or groups or pages.

Test Cases

Name	Description
Valid Comment	Verify that the comment is successfully posted when all parameters are correct.
Self Mentioning	Verify that the comment is successfully posted without sending notification when a user mentions himself/herself.
Empty Text and Mention	Verify that the comment is rejected if no text or mention is there.
Only Mentioning without Text	Verify that the comment is posted even if it does not have a text but mentions some users/groups.

4.7 Use Case: Suggest Edits to Content

Scenario: A user suggests edit to a block of a page content and the owner accepts or rejects the proposed change.

Actors: Page owner, Guest User

Steps:

1. User selects a particular block of the page.
2. User writes a textual comment.
3. Alternatively, user may mention other persons or groups or pages.

Test Cases

Name	Description
Acceptance by Owner	Verify that the edit is properly reflected on the content when the owner approves the change.
Rejection by Owner	Verify that the edit is discarded when the owner rejects the change.

5 Plans and Payment

5.1 Use Case: Upgrade plan

Scenario: A user wants to upgrades their subscription plan to access additional features and benefits.

Actors: User

Steps:

1. User navigates to the Upgrade plan or Explore plans section in the workspace settings
2. User selects the desired plan to upgrade to
3. User inputs the payment details
4. User confirms the upgrade
5. The system processes the payment and upgrades the plan
6. User receives a confirmation message along with an email

Test Cases

Name	Description
------	-------------

User is not an admin or workspace owner	Verify that only an admin or workspace owner can upgrade the plan
Valid payment details	Verify the payment information provided by the user can yield the payment
Invalid payment details	Verify that the payment information provided by the user is invalid and cannot be used for payment
Sufficient balance	Verify that the user has sufficient balance to upgrade the plan
Insufficient balance	Verify that the user does not have sufficient balance to upgrade the plan
Successful upgrade	Verify that the plan is successfully upgraded after payment processing

5.2 Use Case: Downgrade plan

Scenario: A user wants to downgrade their subscription plan to reduce costs or access fewer features

Actors: User

Steps:

1. User navigates to the Plans section in the workspace settings
2. User selects the desired plan to downgrade to
3. User confirms the downgrade
4. The system processes the request and adjusts the plan
5. User receives a confirmation message along with an email

Test Cases

Name	Description
User is not an admin or workspace owner	Verify that only an admin or workspace owner can upgrade the plan
Successful downgrade	Verify that the plan is successfully downgraded after the user confirms the downgrade

5.3 Use Case: Update payment method

Scenario: A user wants to update their payment method for the subscription plan

Actors: User

Steps:

1. User navigates to the Billing section in the workspace settings
2. User modifies the details in the Payment method section
3. User confirms the update
4. The system processes the request and updates the payment method
5. User receives a confirmation message along with an email

Test Cases

Name	Description
User is not an admin or workspace owner	Verify that only an admin or workspace owner can upgrade the plan
Valid payment method	Verify that the new payment method provided by the user is valid and can be used for future payments
Successful payment method update	Verify that the payment method is successfully updated after the user confirms the update

6 Third Party Integration

6.1 Use Case: Task Management Integration

Scenario: A third-party task management application integrates with Notion to synchronize tasks across platforms.

Actors: User, Notion API, Third-Party Task Management System

Steps:

1. User creates or updates tasks in the third-party app
2. Tasks are synchronized with Notion
3. Notion database reflects updated task information

Test Cases

Name	Description
Successful Task Synchronization	Verify that tasks created in the third-party app are successfully synced to the Notion database.
Task Deletion	Verify that tasks deleted in the third-party app are removed from the Notion database.
API Rate Limit Handling	Verify that the system correctly handles API rate limits when syncing a large number of tasks to Notion.
Unauthorized API Call	Verify that unauthorized API calls are blocked by Notion during task sync attempts.

6.2 Use Case: CRM Data Synchronization

Scenario: A CRM system integrates with Notion to sync customer data and notes to Notion databases.

Actors: Sales Representative, Notion API, CRM System

Steps:

1. Sales rep adds or updates customer data in the CRM
2. The system syncs customer data to the Notion database
3. Notion database reflects updated customer information

Test Cases

Name	Description
Customer Data Sync Success	Verify that the customer data is correctly synchronized from the CRM system to Notion.
Duplicate Data Handling	Verify that duplicate customer data entries are properly handled by the integration.
Sync Failure Due to Invalid Data	Verify that invalid data (e.g., missing required fields) causes the sync to fail.