COLLABHUB
USER MANUAL

This is the user manual document for the CollabHub, a Version Control system designed specifically for architecture domain. Detailed descriptions are provided alongside with visuals from the application.

1. Starting the Application

The CollabHub is implemented as a plugin to Revit 2026 application. The prerequisites for using it is as follows:

1.1. Prerequisites

- Revit 2026 application downloaded from official website.
- For ease of use, newest version of Docker Desktop
- Visual Studio

1.2. Steps to Run

1) Once the necessary applications are downloaded, open up Docker Desktop, and Visual Studio 2026.

2) Sometimes the .dll files are not where we expect them to be in different computers which is related to Revit Application. In the RevitVersionControl.csproj file, make sure that the Revit API dll's and target location of addIn, config file paths are according to where they exist in your computer.

   Example:

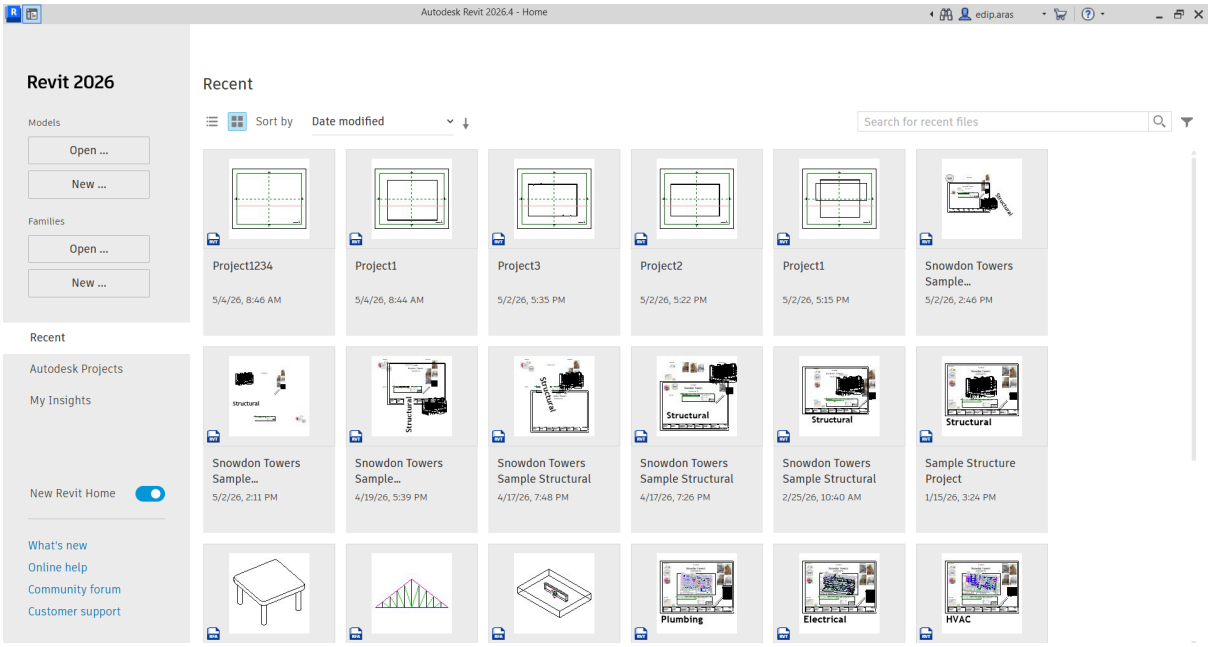
   RevitAPI.dll path: C:\Program Files\Autodesk\Revit 2026\RevitAPI.dll
   RevitAPIUI.dll path: C:\Program Files\Autodesk\Revit 2026\RevitAPIUI.dll
   target config / addin path: %AppData%\Autodesk\Revit\Addins\2026\

3) Run Backend & Database with:
   docker compose down -v
   docker compose up –build

4) Run Frontend with (terminal folder should be the folder of the RevitVersionControl.csproj):
   dotnet build

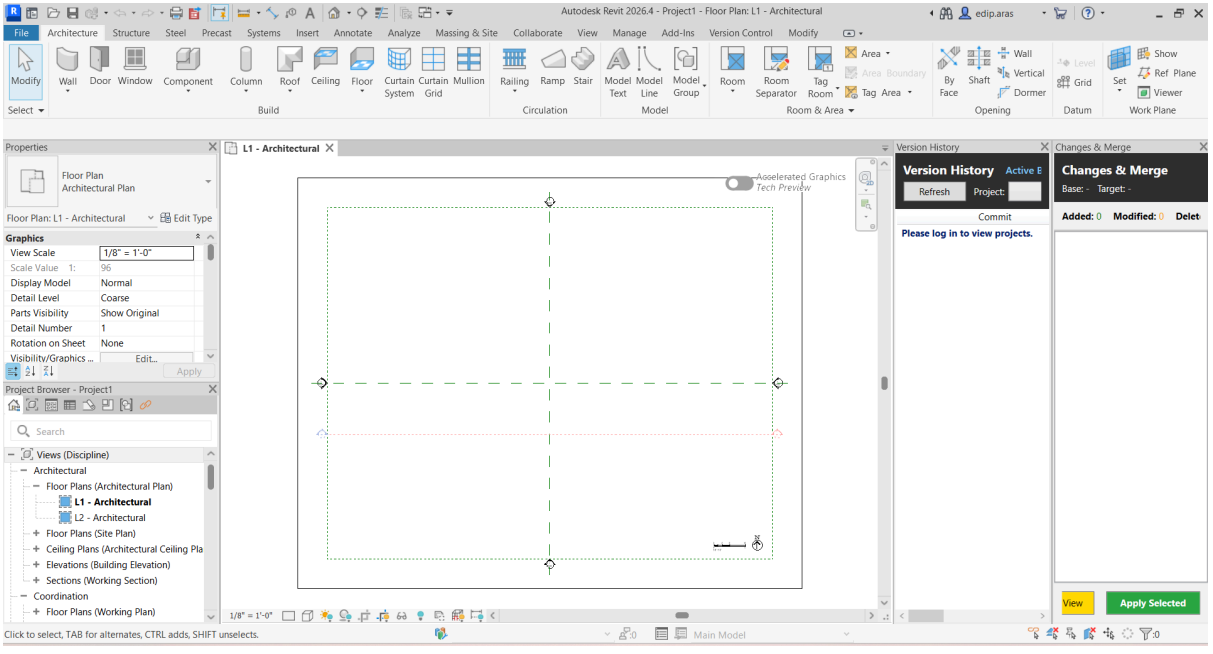
2. General Application Guideline

2.1. Home Page



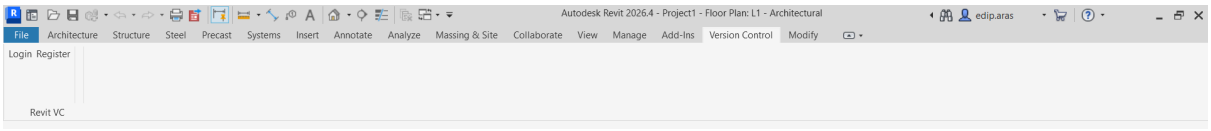
On startup of the Revit application, Home Page shows up. Here one may choose to create a file by picking new / existing models families, or a recent work can be accessed. Other options include filtering the files by features like date and alphabetical order, various Revit Help Tools and so on.

Once we open a file, the following screen will appear:



Here the user may add / modify the file with vast options available on the Top Ribbon. Once some changes are done by the user, the user may use Version Control feature on the right side of the Ribbon.

Once Version Control feature is clicked, the Top Ribbon will look like this:

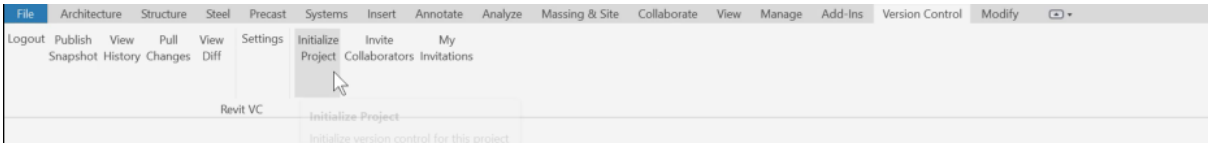


User is prompted to register, and if already registered, login.

Button	**Description**
Login	Opens the Login dialog. Once logged in, this button changes to Logout — clicking it signs you out immediately.
Register	Opens the Register dialog to create a new CollabHub account. This button is hidden once you are logged in.

2.2. Top ribbon: Version control tab:

Once we are logged in, Top Ribbon will show a variety of features as follows:

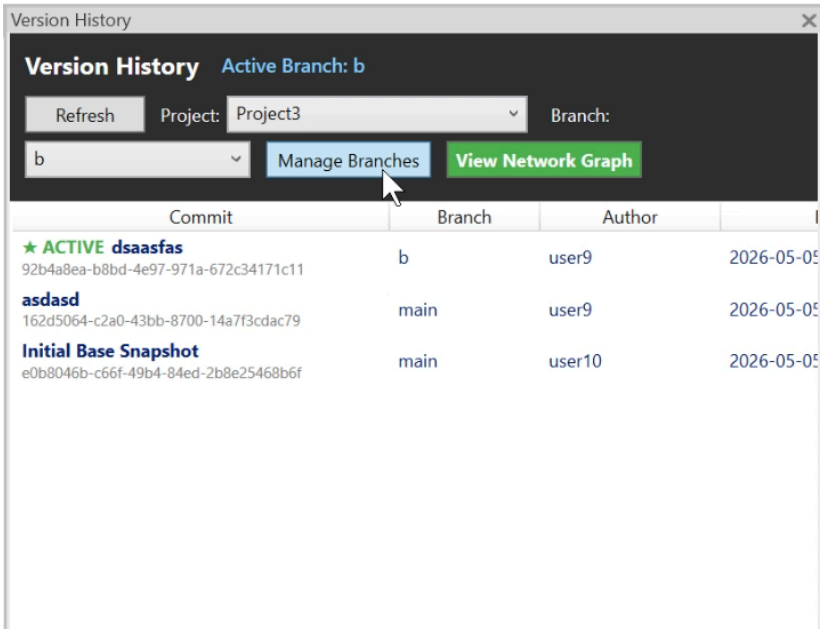


Button	**Description**
Publish Snapshot	Publishes your current model state as a new commit to the server. Only visible when logged in.
View History	Opens the Version History dockable pane. Only visible when logged in.
Pull Changes	Downloads and applies changes from a

	remote commit into your local model. Only visible when logged in.
View Diff	Opens the Commit Diff Viewer dockable pane to compare two commits side-by-side. Only visible when logged in.
Initialize Project	Creates a new CollabHub project for the currently open Revit file and uploads the first version. Only visible when logged in.
Invite Collaborators	Opens the Invite dialog to add other registered users to a project. Only visible when logged in.
My Invitations	Opens the Invitations dialog to view and respond to project invitations you have received. Only visible when logged in.

2.3. Version History Pane

On the right side of the application, there is a Version History section, from where the user can see the file change history in a specific branch.



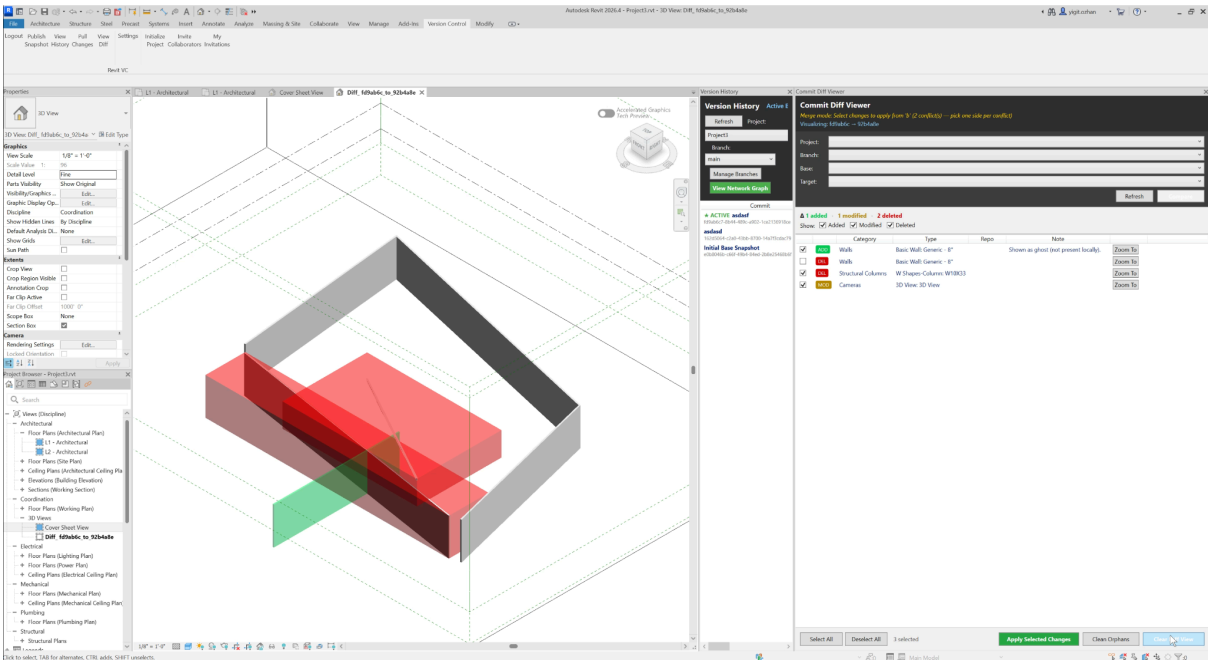
Features in this section are as follows:

Control	**Description**
Active Branch (label)	Shows the branch your local model is currently tracking in blue text.

Refresh (button)	Reloads the commit list from the server. Filter the history to a specific project.
Project (dropdown)	Filter the history to a specific project.
Branch (dropdown)	Filter the history to a specific branch within the selected project.
Manage Branches (button)	Opens the branch-manager-dialog.
View Network Graph (button)	Opens the network-graph-window showing a visual commit/branch diagram.

2.4. Commit Diff Viewer Pane

On the rightmost section of the file lies Commit Diff Viewer, a user can view differences between different versions of the file. What is added, modified or deleted is shown based on the base and target version selected by the user.

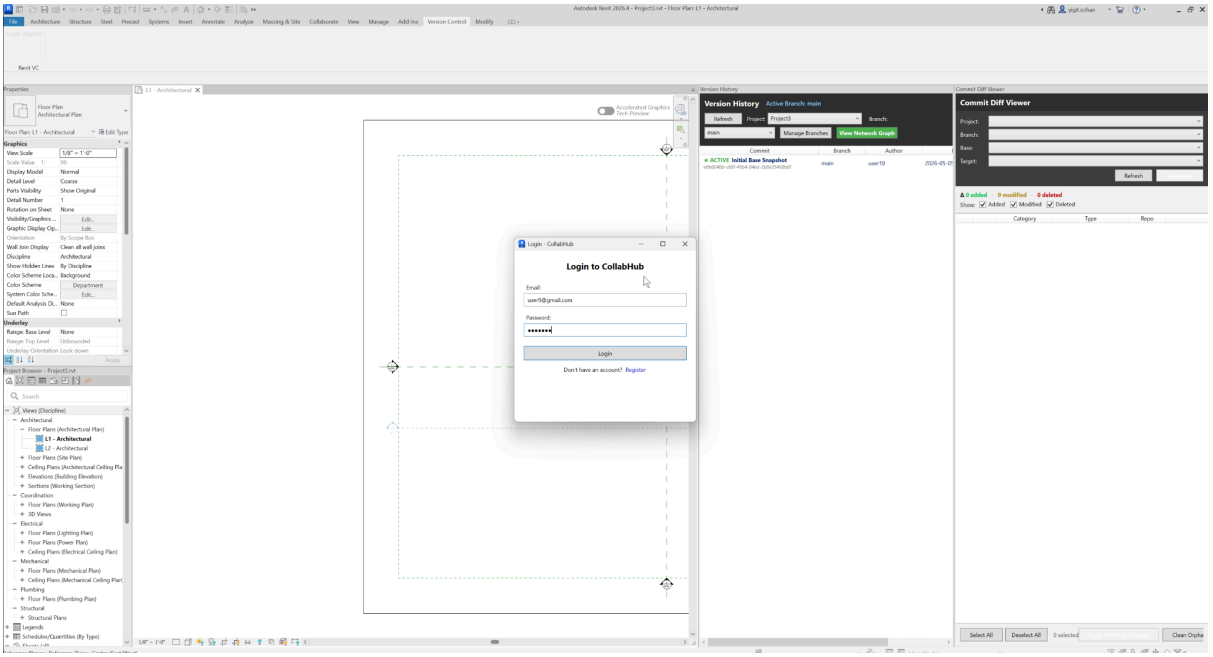


Control	Description
Project (dropdown)	Select the project you want to inspect.
Branch (dropdown)	Narrows commits to a specific branch.
Base (dropdown)	The older of the two commits : the "before" state.

Target (dropdown)	The newer of the two commits : the "after" state.
Refresh (button)	Reloads projects, branches, and commits from the server.
compare (button)	Runs the diff and populates the list below. Enabled only when both Base and Target are selected.
Checkbox	**Description**
Added	Show/hide elements that were created in the Target commit.
Modified	Show/hide elements whose parameters or geometry changed.
Deleted	Show/hide elements that were removed.
Change List Columns	**Description**
(checkbox)	Select this row to include it in an apply action.
(color badge)	Green = ADD, orange = MOD, red = DEL.
Category	The Revit category of the element (e.g., Walls, Floors).
Type	The specific type or family name.
Repo	A short identifier tying the element to its origin model.
Note	A human-readable summary of what changed (e.g., parameter names).
Footer Buttons	**Description**
Select All	Checks all rows in the list.
Deselect All	Unchecks all rows.
X selected (label)	Live count of checked rows.
Apply Selected Changes (button)	Applies only the checked element changes to your open Revit model. Enabled when at least one row is selected.
Clean Orphans	Removes all diff visualization geometry (ghost elements and temporary views) that may have been left in the document from a previous session.
Clear Diff View (red)	Clears the current comparison result and resets the pane to its empty state.

2.5. Login Dialog

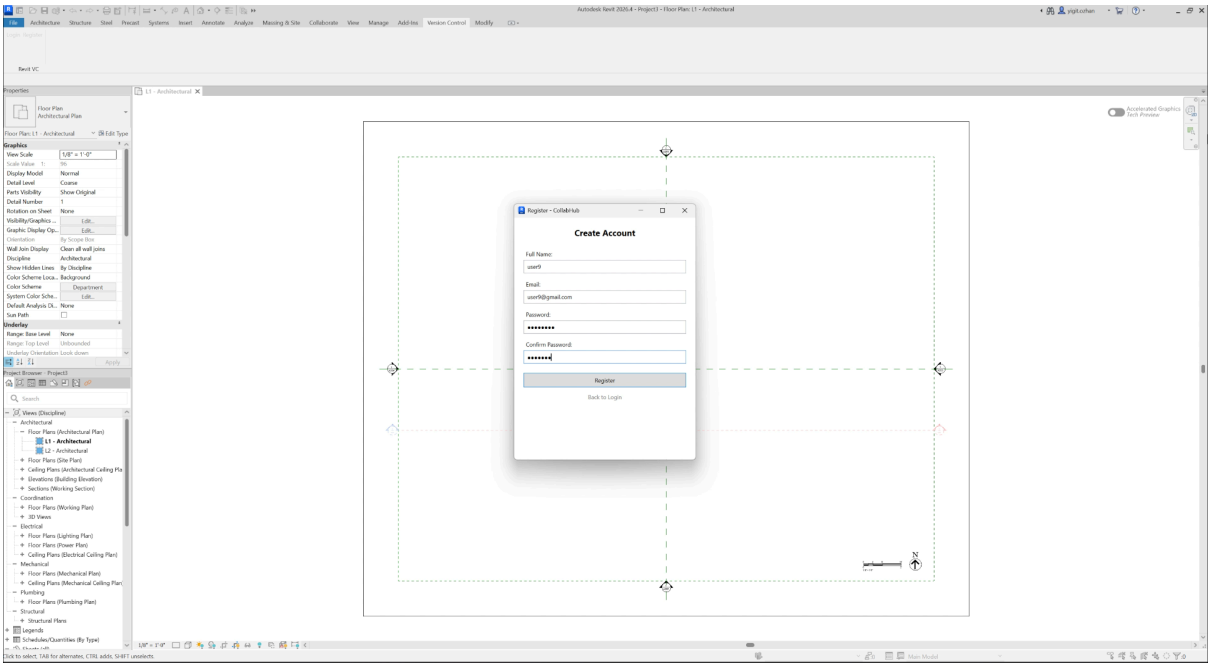
Helps user with logging in.



Field / Button	Description
Email	Enter the email address associated with your CollabHub account.
Password	Enter your account password.
Login (button)	Submits your credentials. On success, the dialog closes, the ribbon updates to show all features, and the History and Diff Viewer panes reload your projects.
Register (button)	If you don't have an account yet, click this link to open the Register dialog instead.

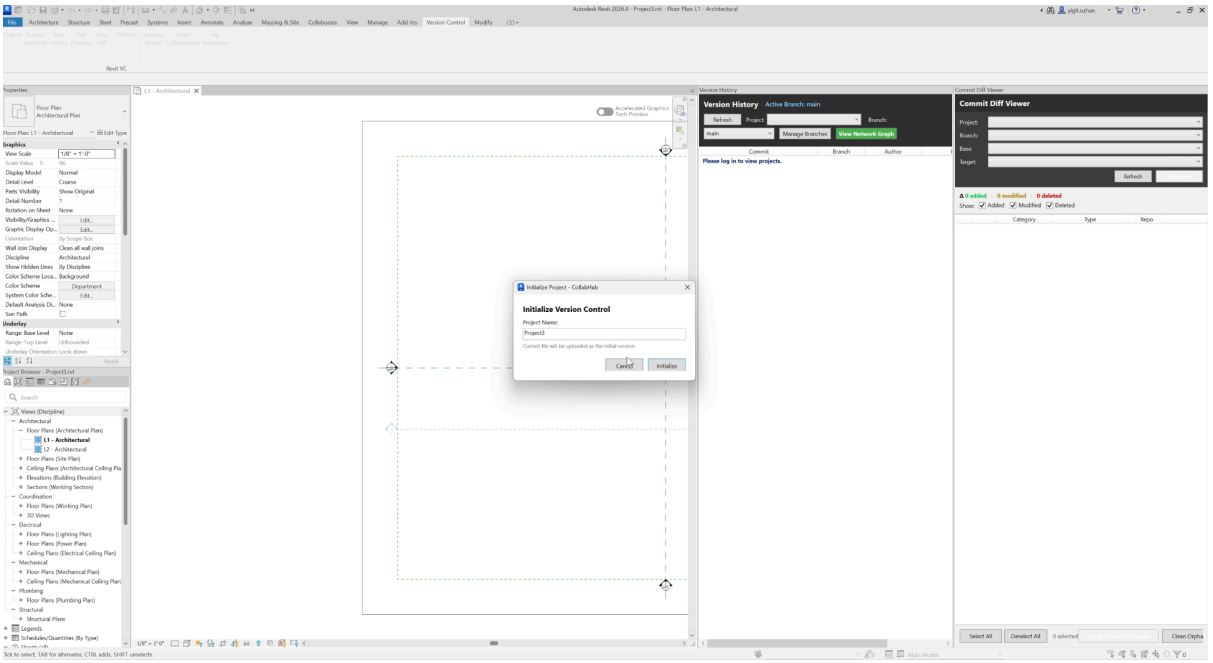
2.6. Register Dialog

Helps user with registration, if already exists rejects the user form and prompts user to retry.



2.7. Initialize Project Dialog

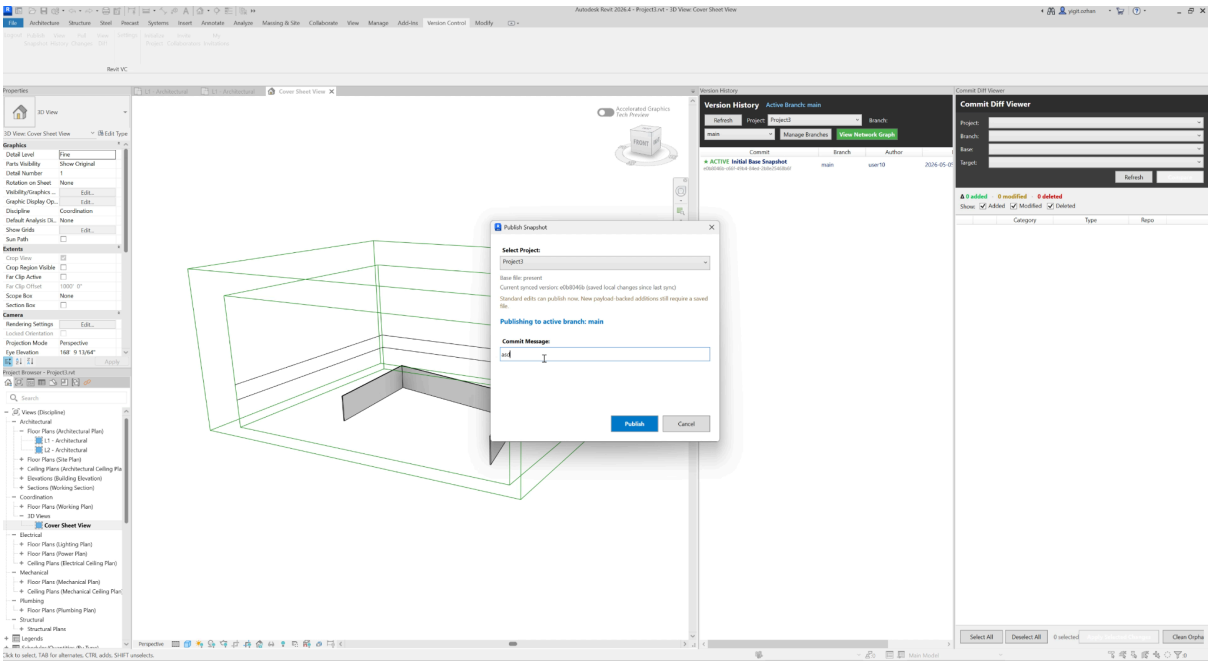
The user is prompted with a initialization step before using any of the version control system features. Initialize Project saves the current file to a specified location, and then publishes the first commit as ‘Base Commit’.



Field / Button	Description
Project Name	Enter a descriptive name for your project (e.g., "Tower A - Structural").
Initialize (button)	Creates the project on the server and uploads the current Revit file as the initial baseline version.
Cancel (button)	Closes the dialog without creating anything.

2.8. Publish Snapshot Dialog

Once the modifications are done, the user may push the changes made to the version control system so that the changes can be pulled by other users working on the same project. Version History panel is updated after publishing of the snapshot is completed.

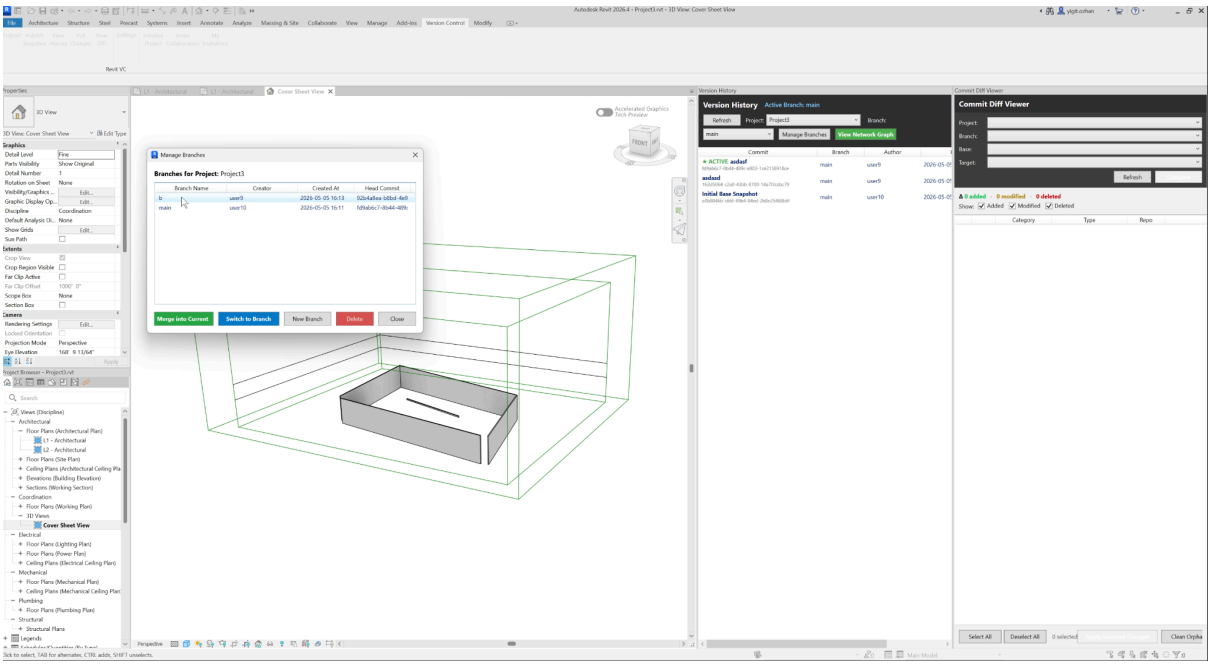


Field / Button	Description
Select Project (dropdown)	Choose which CollabHub project this publish belongs to.
Base file status	Informational line showing whether the server already has a base file for this model.
Current synced version	Shows the commit ID your local model is currently tracking.
Publishing to active branch	Displays the branch the commit will be added to (e.g., main).

Detached Branch Panel (appears when needed)	If you are on an older commit that is not the branch head, you must create a new branch. Enter a New Branch Name here before publishing.
Commit Message	Enter a short human-readable description of what changed (e.g., "Added floor slabs Level 3").
Publish (button)	Extracts all elements from the model, computes a delta diff against your last known commit, and sends the result to the server. A success dialog shows the short commit ID and element count.
Cancel (button)	Closes the dialog without publishing.

2.9. Branch Manager Dialog

From the Version History panel on the right side, user may click to ‘Manage Branches’ button to see different branches and their details. From here, the user can perform operations such as merging or switching between different branches.

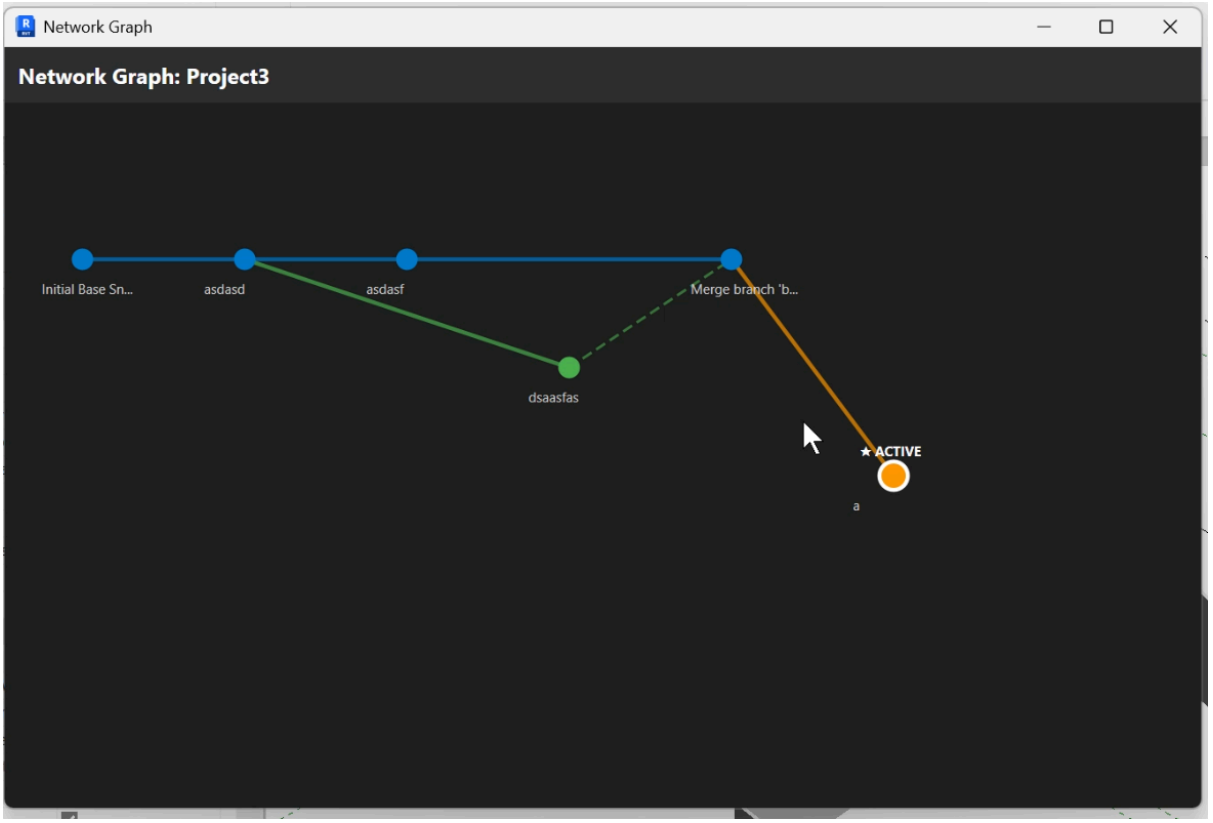


Field / Button	**Description**
Branch Name	The name of the branch.
Creator	The user who created the branch.
Created At	When the branch was first created.

Head Commit	The ID of the most recent commit on this branch.
Merge into Current (button)	Merges the selected branch into your currently active branch. This triggers a 3-way merge and opens the Changes & Merge pane if conflicts are found.
Switch to Branch (button)	Check out from the selected branch so future publishes go to it. Your local model is updated to track this branch.
New Branch (button)	Creates a new branch starting from the current HEAD commit. You will be prompted to enter a branch name.
Delete (button)	Permanently deletes the selected branch from the server. Use with caution.
Close (button)	Closes the dialog.

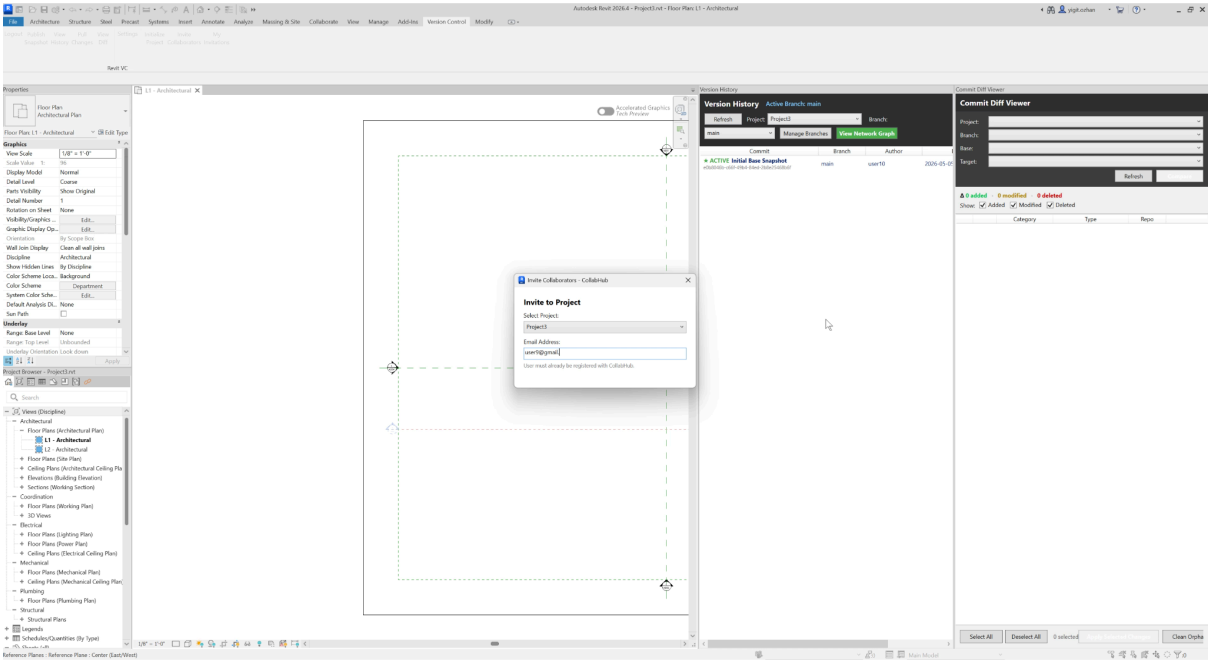
2.10. Branch Graph View

Shows all commits and branches in a single visual graph, including parent-child relationships, branch creation points, and merge points.



2.11. Invite Collaborators Dialog

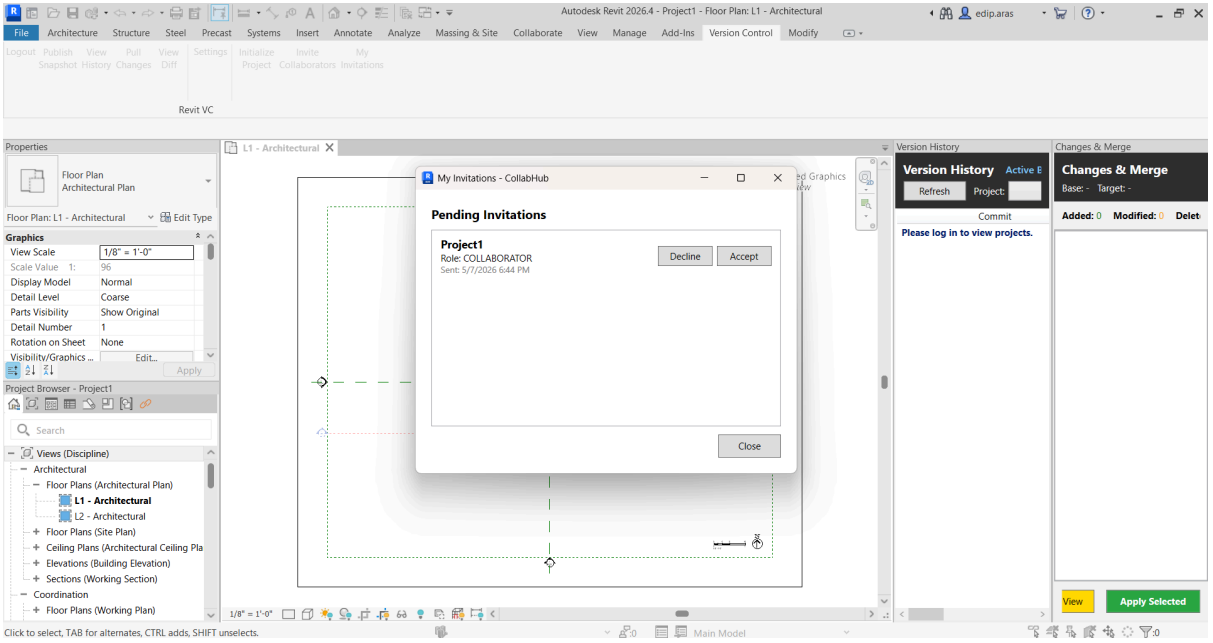
A user can invite other users to collaborate on a project, by providing their emails via ‘Invite Collaborators’ section on the Top Ribbon. Once the invitation is accepted, these users will be able to collaborate as well.



Field / Button	Description
Select Project (dropdown)	Choose which project to invite the user to.
Email Address	Enter the email of the person you want to invite. They must already have a CollabHub account.
Send Invite (button)	Sends the invitation. The recipient will see it in their My Invitations dialog.
Close (button)	Closes the dialog without sending.

2.12. My Invitations Dialog

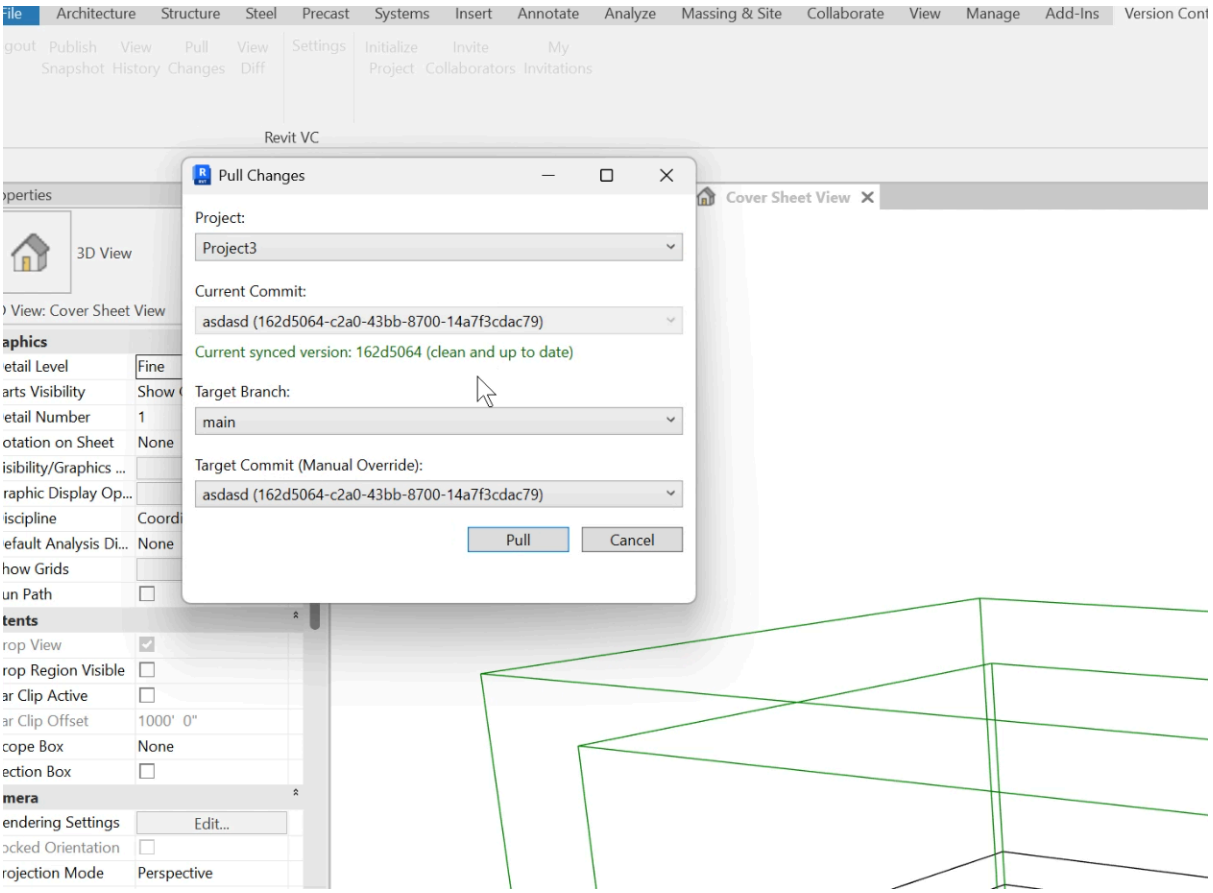
The user can view the incoming invitations via ‘My invitations’ part on the Top Ribbon. If invitation Accepted, the application prompts the user to store the new user’s version of the file in some folder in the computer.



Field / Button	Description
Accept	Joins the project. It will appear in your project dropdowns across all panes.
Decline	Rejects the invitation. It is removed from the list.
Close	Closes the dialog.

2.13. Pull Changes Dialog

The user can pull the changes of the target commit from the base commit with the ‘Pull Changes’ functionality. Pulling any commit is possible, not only the most recent one.



Field / Button	Description
Project (dropdown)	Select the project to pull from. The dropdown auto-selects the project that this document is already tracked against, if one exists.
Current Commit (dropdown)	Shows the commit your local model is currently at. If the document is tracked, this is locked and set automatically you cannot change it. If it is unknown, you must select it manually from the list.
Current version status (colored text line)	A live status line that describes the sync relationship between your local model and the server. The color communicates severity: green = clean and up to date, blue = remote has newer commits, orange = you have local unsaved changes, dark red = both local changes and remote changes exist simultaneously.
Target Branch (dropdown)	Choose the branch you want to pull from. Defaults to your currently tracked branch (usually main). Changing the branch automatically updates the Target Commit to that branch's head.
Target Commit (Manual Override) (dropdown)	Normally set automatically to the head of the selected branch. You can override this to pull to any specific

	historical commit on the list.
Pull (button)	Validates your selections, then contacts the server to compute changes between the Current and Target commits. If the model is already at the target, a message tells you it is up to date and nothing happens. If conflicts are detected, the Commit Diff Viewer pane opens automatically so you can review and choose which changes to apply. If there are no conflicts, a confirmation dialog shows a summary (added / modified / deleted counts) and asks whether you want to apply the changes.
Cancel (button)	Closes the dialog without pulling anything.