



Hemant Umre

GSoC 2023 PROPOSAL

Improving Zulip's Permissions with User Groups

List of **contents**

1. [Personal details](#)
2. [About me](#)
3. [Why Zulip?](#)
4. [Why me?](#)
5. [Contributions to Zulip](#)
6. [Proposal Abstract](#)
7. [Proposed Contributions](#)
8. [Project timeline](#)
9. [My availability](#)
10. [Post GSoC](#)

Personal details

NAME Hemant Umre

COLLEGE University institute of technology, RGPV

COURSE Bachelors in Technology(2020 - 2024)

NATIONALITY Indian

TIME_ZONE (GMT+5:30) Asia/Kolkata - IST

EMAIL hemantumre12@gmail.com

LINKEDIN [Hemant Umre](#)

GITHUB [HemantUmre12](#)

About me

Hello, my name is Hemant Umre and I am an undergraduate student currently pursuing my studies at the University Institute of Technology, RGPV. My experience with programming languages such as C++, JavaScript, and Python has enabled me to work on [various projects](#), and I had the opportunity to learn Django while [contributing to Zulip](#). I have also gained valuable experience working in teams and holding leadership positions in several extracurricular activities.

Problem-solving is one of my passions, and I began learning data structures and algorithms during my freshman year. I have since solved over 1000 problems across various platforms like [Leetcode](#) and [Codeforces](#), actively participating in coding contests. However, my interest shifted towards working on real-world projects with a team, which is where I was introduced to open-source software and Zulip.

As a member of [Asper](#) (a design and development club of my college), I have collaborated with fellow members to [design](#) and [implement](#) websites using Node.js, React and Tailwind CSS, as well as other technologies. During my time as a co-head of the core-coding department, I mentored juniors on data

structures and algorithms.

When I'm not coding, I enjoy playing games like Valorant and GTA V, watching anime, weight training and playing volleyball.

Why Zulip?

Zulip was the first open-source organization I had the opportunity to contribute to. Although I had attempted to contribute to other large codebase organizations previously, I faced difficulties setting up a development environment for some of them. However, I was pleasantly surprised by the ease with which I could set up a development environment within the Zulip community.

In my view, Zulip is a user-friendly open-source project to contribute to. The community is highly active and supportive, with seamless discussions, extensive documentation, and thorough pull request reviews. My time at Zulip has been invaluable in enhancing my teamwork and collaboration skills, as well as my knowledge of code quality and best practices, community building, and technical skills. As a GSoC contributor, I am eager to continue making quality contributions and enhancing the product.

Why Me?

With a background in [problem-solving](#) and experience with [JavaScript](#), CSS, and related frameworks like [React](#) and [TailwindCSS](#), I am well-equipped to design efficient algorithms, understand code written by other people, and effectively communicate my ideas.

While my contributions to Zulip may not be extensive, I am deeply invested in the project I aim to undertake. For the past month, I have focused my efforts

on addressing the "[Add settings for restricting direct messages](#)" issue by familiarizing myself with the UserGroup model, exploring relevant pull requests and discussions, and starting additional [conversations](#) about it. Despite ongoing [technical blockers](#), I am committed to seeing this project through and plan to begin actively working on it once my proposal is complete. It will be the first realm-level permission that uses user groups, and it is crucial to the overall vision of [Zulip's migration to user groups for permissions](#).

I have also gained experience working on stream settings areas through projects such as [#24216](#) and [#24254](#), and familiarized myself with settings for admins and organizations by working on project [#24762](#).

Contributions to Zulip

Links to all the relevant contribution made by me, As of **4th April** -

- ☐ [Bug reports, discussions and helping others.](#)
- ☐ [All pull requests.](#)
- ☐ [All issues raised.](#)

ISSUE	DESCRIPTION	STATUS(PR)
PULL REQUESTS		
#23241	left_sidebar: Fix section divider not shown when all streams are muted.	Merged
#22844	left_sidebar: Add login link to left sidebar for logged-out-users.	Merged
#23737	setting(user): Sort Uploaded files with newest on top.	Merged
#24048	stream_settings: Add an option to make a stream default for new users in stream creation and editing UIs.	Open
#24238	stream_settings: Fix Announce stream disables unexpectedly.	Merged
#24467	settings(admin/org): Add settings for restricting direct messages.	Draft
ISSUES RAISED		
#23241	left_sidebar: Fix section divider not shown when all streams are muted.	Merged
CZO	Make "Who can unsubscribe others from the stream?" wide in stream creation UI.	Merged
PR REVIEWED		
CZO	Add "Nobody" system user group.	Merged

Proposal **abstract**

My goal during the GSoC period would be to-

- Develop a framework and discuss a strategy for migrating existing realm and stream settings to user groups in reviewable chunks.
- Add high-priority new features and permissions at the organizational, stream, and group levels.
- Contribute to the development of new features and bug fixes in the [user group management UI](#).

The [proposed contributions](#) elaborate on these points.

Proposed **contributions**

I am thrilled to apply for the **350-hour** large project and contribute to Zulip's goal of implementing user groups for meaningful permissions. Through some extensive research of relevant pull requests and discussions, as well as keeping up to date with project's progress. In this section, I will outline my proposed contributions.

1. Migrate permissions to user groups:

ISSUE: [#19525](#)[Permissions UI elements]

NOTE: Before proceeding, we need to determine whether to use only system groups or also incorporate custom groups as configurations. As the user group management UI may not yet be available, we can initially utilize system groups. Once the user group management UI is deployed, we can then update the permissions and incorporate custom groups as well. Additionally, we should add a feature to navigate to the user group management UI.

Migrate “Who can post to this stream” -

This is currently the only stream level permission we aim to migrate to user groups. This can serve as a sample project before we start migrating realm permissions.

Migrate realm permissions -

List of realm permissions needs to migrate to user groups -

S.No	Permissions	Variable name(API)
1.	Who can invite user to this organization	invite_to_realm_policy
2.	Who can create public streams	create_public_stream_policy
3.	Who can create web-public streams	create_web_public_stream_policy
4.	Who can create private streams	create_private_stream_policy
5.	Who can add users to streams	invite_to_stream_policy
6.	Who can use @all/@everyone mentions in large streams	wildcard_mention_policy
7.	Who can move messages to another topic	edit_topic_policy
8.	Who can move messages to another stream	move_messages_between_streams_policy
9.	Who can delete their own messages	delete_own_message_policy
10.	Who can add bots	bot_creation_policy
11.	Who can create and manage user groups	user_group_edit_policy

12.	Who can add custom emoji	add_custom_emoji_policy
13.	Who can use direct messages(beta)	private_message_policy

APPROACH:

This is a rough approach to migrate “Who can create public streams” most of the realm permissions can use the same general approach:

- Add create_public_stream_group field in Realm table. This will replace create_public_stream_policy.
- Set create_public_stream_group while creating realm according to organization type.
- Add migration to set value of create_public_stream_group for already existing realms based on create_public_stream_policy.
- To handle database interactions we likely want to add something similar to the property_types framework, say usergroup_property dictionary, which will be used by do_set_usergroup_property added to zerver/actions/realm_settings.py. This function will update the database and trigger an event to notify clients about the changes for realm fields using UserGroup as ForeignKey. Since there are more than a dozen of permissions and we don’t want to write an action function for them individually.
- Currently “Who can create public streams” uses common_policy_values to populate dropdown instead we plan to use user groups to populate dropdown_list_widget.js- > DropDownListWidget and generalize this for realm permissions.
- We have to update settings_data.ts- > user_has_permission to check whether the acting user is part of the config user group.

2. New settings:

Add settings for restricting direct messages -

ISSUE: [#24467](#)

DRAFT PR: [#24762](#)

DETAILS: One of my ongoing projects is to add settings for restricting direct messages. Since this is the first realm feature using the user groups, I anticipate that some iterations to the PR may take place during the early GSoC period. Updating the logic for changing an empty PM view when does not have permissions([#21889](#)) will be the follow-up task.

APPROACH:

- Add a database model.
- Set default values when creating the realm.
- Add migration to set defaults for already-existing realms based on `private_message_policy`.
- Add action function and allow changes through API.
- Add UI elements (dropdown) for new settings.
- Add live update code for new settings.
- Update `check_message` and `can_send_private_message` using new settings.
- Remove "Who can use direct messages (beta)" from UI and `private_message_policy` from API.

New stream settings-

Now that we have added new settings at both stream and realm levels, we can start adding new settings by using them as reference. I will start with ones that are high-priority or that can be added more easily. Below is a list of some stream settings that have been long-awaited:

Who can change stream settings -

ISSUE: [#4075](#)

DETAILS: As of now, only organization administrators can change stream settings, which can be problematic in cases where a private stream doesn't have an admin user as a subscriber, making it impossible for anyone to change the stream settings.

Who can create new topics -

ISSUE: [#19642](#)

DETAILS: For some organizations, it would be useful to restrict creating topics in certain streams. We should add a setting for which groups can create new topics in a stream.

Apart from these, I will seek out opportunities to contribute to any important features that require my help.

Note: For stream settings, one of the most intuitive configurations will be the user who created the stream (e.g., settings like "Who can change stream settings" or "Who can subscribe other users to the stream"), but as of now, we don't have a framework to support these types of user groups (i.e., groups that are just user X). This issue is discussed in detail [here](#).

3. User group management UI:

UI improvement-

ISSUE: [#19526](#)

DETAILS: [@Purushottam Tiwari](#) is actively working on this project and has already implemented most of the features outlined in the issue. This is an important and mandatory component for using the full potential of user groups and achieving the overall vision described in issue [#19525](#). If any bugs or additional features arise, I am willing to provide assistance. My primary focus will be on designing and implementing the permissions tab, so that we can begin adding user group permissions as described in issue [#19525](#).

User group permissions-

ISSUE: [#19525](#) [User group management UI]

DETAILS: We need to add several user group permissions, including groups that can mention this group, groups that can add members to this group, groups that can remove members from this group, and groups that can edit settings for this group. I will work on implementing these permissions once the permissions tab is in place.

Project timeline

As previously mentioned, I will be taking on a large **350-hour** project. Although the timeline may appear aggressive, considering the necessary follow-ups for each new setting and migration. I believe it's necessary to tackle all three sections if we want to fully leverage the potential of user groups. I have kept the timeline simple and straightforward.

Community Bonding Period

(4th May - 28th May)

During this period, I may have semester exams(see "[My availaibility](#)"), so most of the work will focus on discussions and ongoing PRs.

0. Bonding with the community.

1. Migrate permissions to user groups-

- Start migrating "Who can post to this stream" and finish it.
- Discuss the framework and strategy for migrating realm permissions, then migrate the first realm permission.

2. New settings-

- Finish working on "Add settings for restricting direct messages" and discuss follow-ups.

3. User group management UI-

- Discuss any bugs or features related to the user group management UI before working on the "permissions" tab.
- Discuss the UI for the "permissions" tab and implement it.

Phase - 1

(29th May - 13th July)

1. Migrate permissions to user groups-

- Finish migrating first realm permission.
- Start migrating other realm permissions based on the discussed strategy, using the first migrated realm permission as a sample.

2. New settings-

- Discuss if we want "@user" system groups for stream settings.
- Implement "Who can change stream settings" and stream setting.

3. User group management UI-

- Start implementing "Groups that can add members to this group" and "Groups that can remove members from this group" group settings.

Midterm Evaluation

(10th July - 14th July)

Review progress so far and make any necessary adjustments to the plan.

<p style="text-align: center;">Phase - 2 (14th July - 28th August)</p>
<ol style="list-style-type: none"> 1. Migrate permissions to user groups- <ul style="list-style-type: none"> → Finish migrating all the realm permissions. 2. New settings- <ul style="list-style-type: none"> → Implement "Who can create new topics" new stream setting. 3. User group management UI- <ul style="list-style-type: none"> → Add "Groups that can mention this group" and "Group that can edit settings for this group" group settings.
<p style="text-align: center;">Final Evaluation (28th August - 4th September) Post GSoC.</p>

My availability

During the latter half of May(I don't know the exact dates), I will be occupied with my semester exams and won't be able to make significant progress on the project for a period of two weeks. However, I will still be available through CZO and email, and will continue to update the pull requests and actively participate in discussions that concern me. Apart from this, I do not have any other commitments for the summer and can easily dedicate around 35-40 hours per week for 12 weeks during the GSoC program period.

In the event of any unforeseen circumstances that affect my availability, I will communicate these to my mentors and in check-ins.

Post **GSoC**

After the conclusion of the GSoC program, I will continue to work on the pull requests that I have opened or that require my attention. This project is a long-term initiative that will keep growing and evolving to become an essential component of Zulip. To ensure that I remain involved in the community, I will be more than happy to guide new contributors to get started in the area around this project.

Overall, I am enthusiastic about the opportunity to continue contributing to this project, as well as to other initiatives beyond the GSoC program. I believe that this experience will help me grow as a developer, and I am eager to continue learning and improving my skills.