Adding Spaces Support in NeoChat

Google Summer of Code 2022 Proposal

CONTACT INFORMATION

Name	Snehit Sah
Website	https://snehit.dev
Matrix	@flyingcakes:kde.org
GitHub	@flyingcakes85
KDE Invent	@flyingcakes
Twitter	@flyingcakes85
Email	hi@snehit.dev
Location	New Delhi, India (GMT +5:30)

PROJECT INFORMATION

Title	Adding Spaces Support in NeoChat
Duration	350 hours
Expected time	12 weeks

ABSTRACT

Spaces are a feature in Matrix to group related rooms. Spaces can be nested and have their summary page. A room may belong to multiple spaces at the same time. **This makes discovery of rooms easier for end-users, and at the same time allows communities to drive more participation.**

NeoChat and LibQuotient (the underlying library currently) don't support spaces. Having this support will make it a more viable and feature-rich Matrix client, up to date with the latest Matrix specification.

Spaces were introduced in MSC1772 and there is a related open feature request on the NeoChat bug tracker.

PROJECT GOALS AND BENEFITS

This project aims to add Spaces support in NeoChat. More specifically, users should be able to

1. View Space details (rooms, topic, description etc)
2. Join existing Spaces and child rooms
3. Create new Spaces (public or private)
4. Set and view Space landing page
5. Allow space administrators to make changes to Space settings
6. Allow space administrators to set users to automatically join certain room(s) when they join a Space

In the process, I also aim to make contributions to LibQuotient, which is the library NeoChat relies on.

Currently, NeoChat can only list Spaces on the sidebar. (MR 309) Other features are missing. Spaces are treated as normal rooms, and room membership changes are shown when opening a Space in NeoChat.

The benefits include easy discovery of rooms by users and driving more participation as users will easily find the right rooms to talk in.

IMPLEMENTATION

NeoChat uses LibQuotient for interacting with Matrix servers. Adding Spaces support to NeoChat will require making upstream contributions to LibQuotient.

The first step will be to have LibQuotient support spaces. As per the [Matrix Client Server Specification](), Spaces can be identified via the [room type]() property `m.space`. LibQuotient already recognizes the Space room type ([commit link]()).

While a Space can have any room as its child, a room can declare a space as its parent only if either the parent too declares the room to be its child or the sender of `m.space.parent` event has sufficient power level in the Space to send `m.space.child` event. LibQuotient needs to ensure the space hierarchy it reports satisfies this requirement.

The class Space will also be populated with required member functions. The most basic function is to get a list of rooms belonging to the Space. This can be done with a `GET` request at `/_matrix/client/v1/rooms/{roomId}/hierarchy`. It needs to be ensured that LibQuotient API call returns a room only once, as a child room might appear multiple times in the Matrix GET request (for example, as a grandchild).

A new function will be added to LibQuotient that will allow creating Spaces

Once LibQuotient gets support for Spaces, I will move on NeoChat to add relevant user interface and API interaction with LibQuotient.

`NeoChatRoom` class in NeoChat source code will be extended to add the new Spaces functionality. Spaces will be treated as Rooms with different user functionalities (the same way that Matrix specifications treat it as). The user interface will be adjusted based on whether the specific room is a normal chat room or a Space.

On the user front, I will design a new display page for spaces in QML. This page will allow users to view child rooms and join them. The Space administrators will also be able to edit Space settings (name, description, default rooms etc). A new option will be added in the application menu to allow creating Space.

TENTATIVE TIMELINE

- **May 20 - June 12** - *Community Bonding Period*

  Finalise meeting and report schedule with mentors; get a more technical grasp of how to go about adding required functionality to LibQuotient; play around with existing NeoChat source code to learn more about how the project is structured

- **Week 1 - 2** (*June 13 - June 26*)

  Work on implementing Spaces support in LibQuotient (create, edit Spaces).

- **Week 3** (*June 27 - July 3*)

  Update QuoTest to have tests for Spaces, and update existing tests if needed. Also add relevant usage instructions/ documentation.

- **Week 4 (** *July 4 - July 10*)

  Finalise UI code for NeoChat with Spaces. At this point, the UI will need to be populated with dummy data.

- **Week 5 - 6** (*July 11 - July 24*)

  Work on integrating the new Spaces API from LibQuotient into NeoChat and test features. By this time, NeoChat is expected to be able to preview Spaces and join rooms.

- **Week 7 - 8** (*July 25 - August 7*)

  Work to enable NeoChat to be able to create / edit Spaces

- **Week 9 - 10** (*August 8 - August 21*)

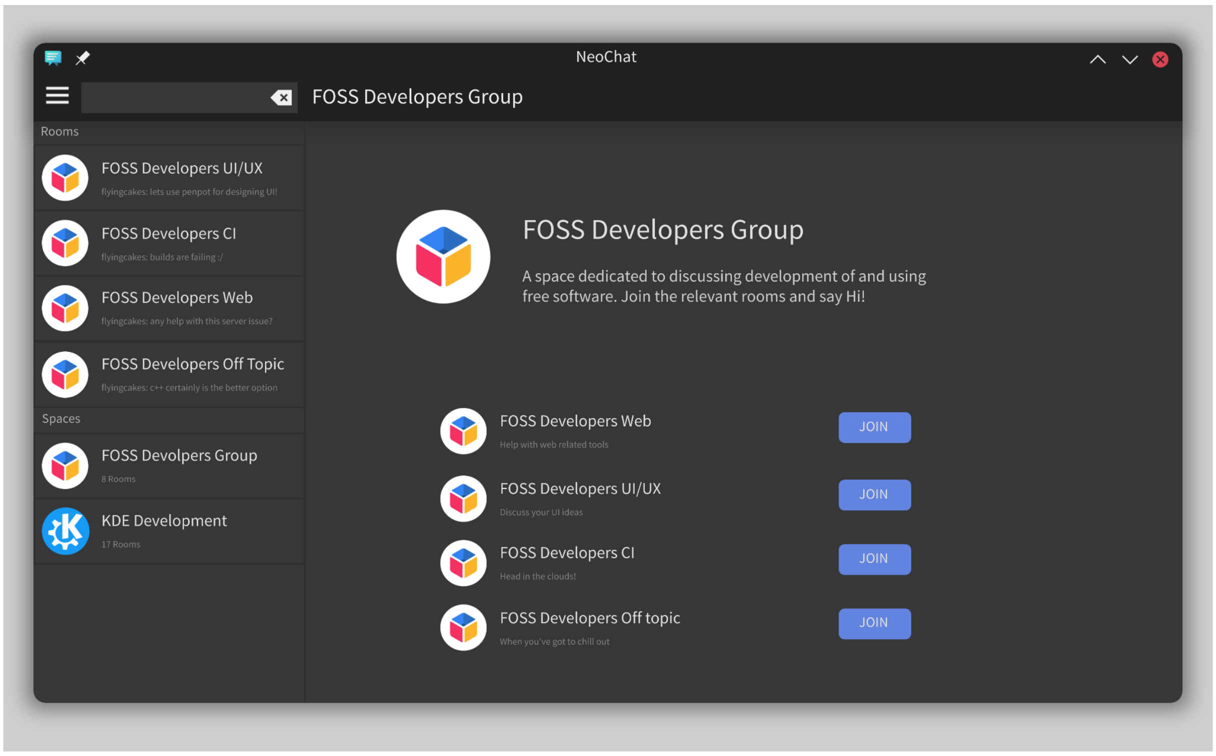
  Any required bug fixes, finalising features, refactoring (if needed), updating / adding more documentation wherever required

I have kept two weeks as a buffer to compensate for any unexpected dependency breakages or major updates that turn out as necessary changes to accommodate the Spaces API. I will also use this time to polish related documentation.

Through the contribution period, I will post regular updates on my blog, and be in constant communication with my mentors over the NeoChat Matrix channel. I shall also be available for weekly meets/discussions to formally discuss my progress and decide for the upcoming weeks.

UI DESIGNS

These are the major changes to the UI that I propose to make.



A space dedicated to discussing development of and using free software. Join the relevant rooms and say Hi!

Fig 1: Viewing a Space on Desktop

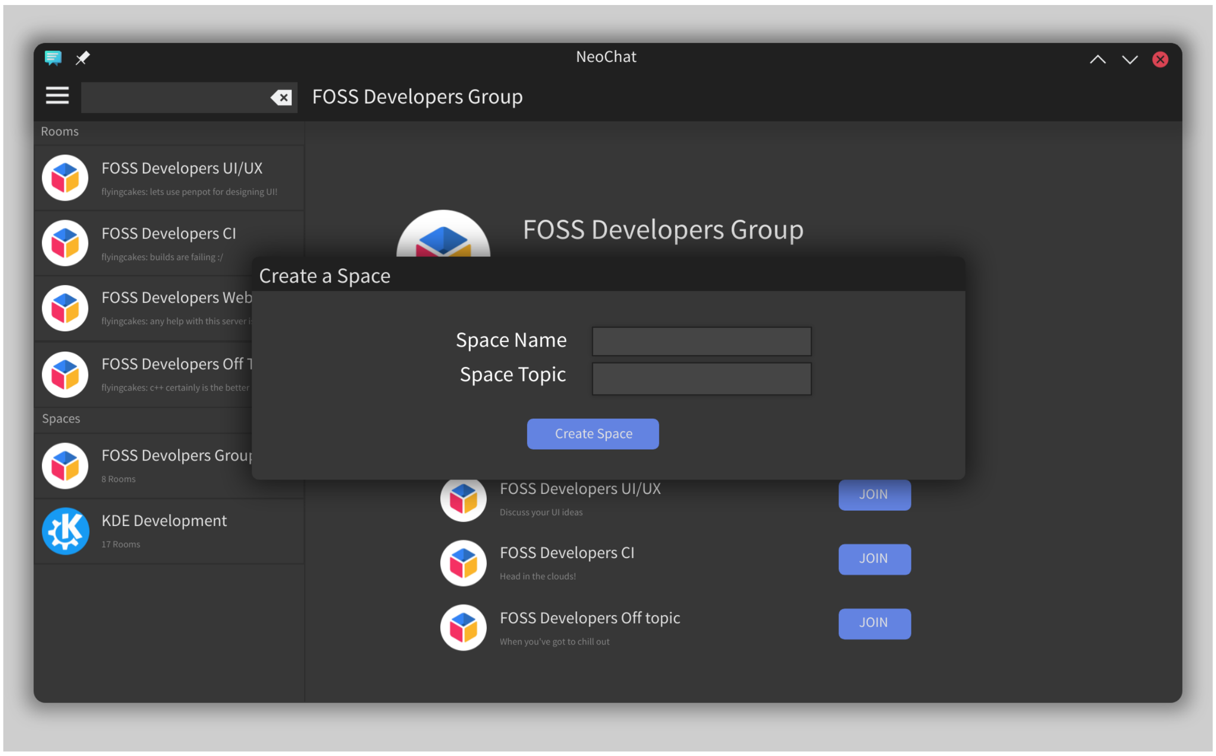
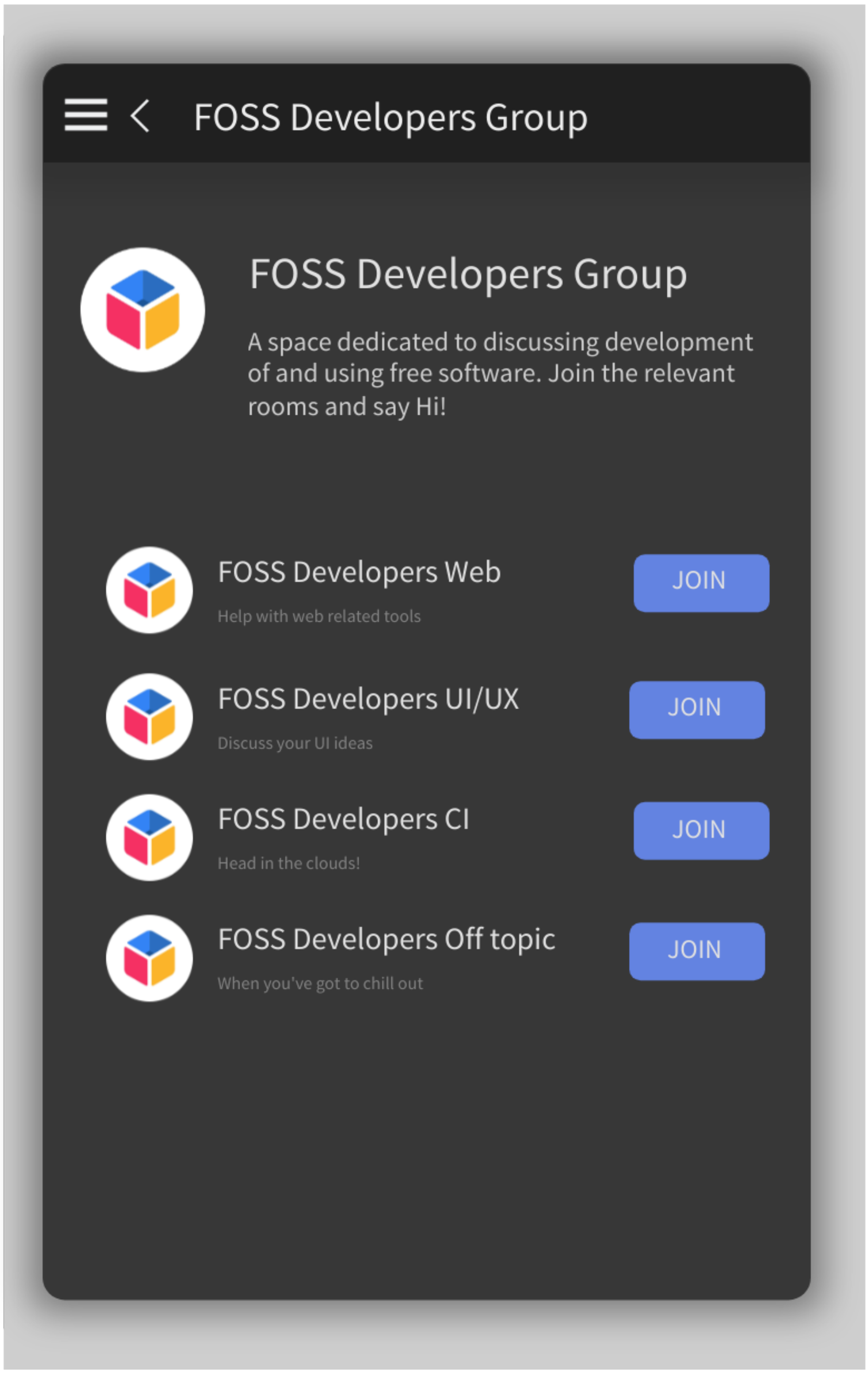


Fig2: Creating a Space on Desktop


FOSS Developers Group

FOSS Developers Group

A space dedicated to discussing development of and using free software. Join the relevant rooms and say Hi!

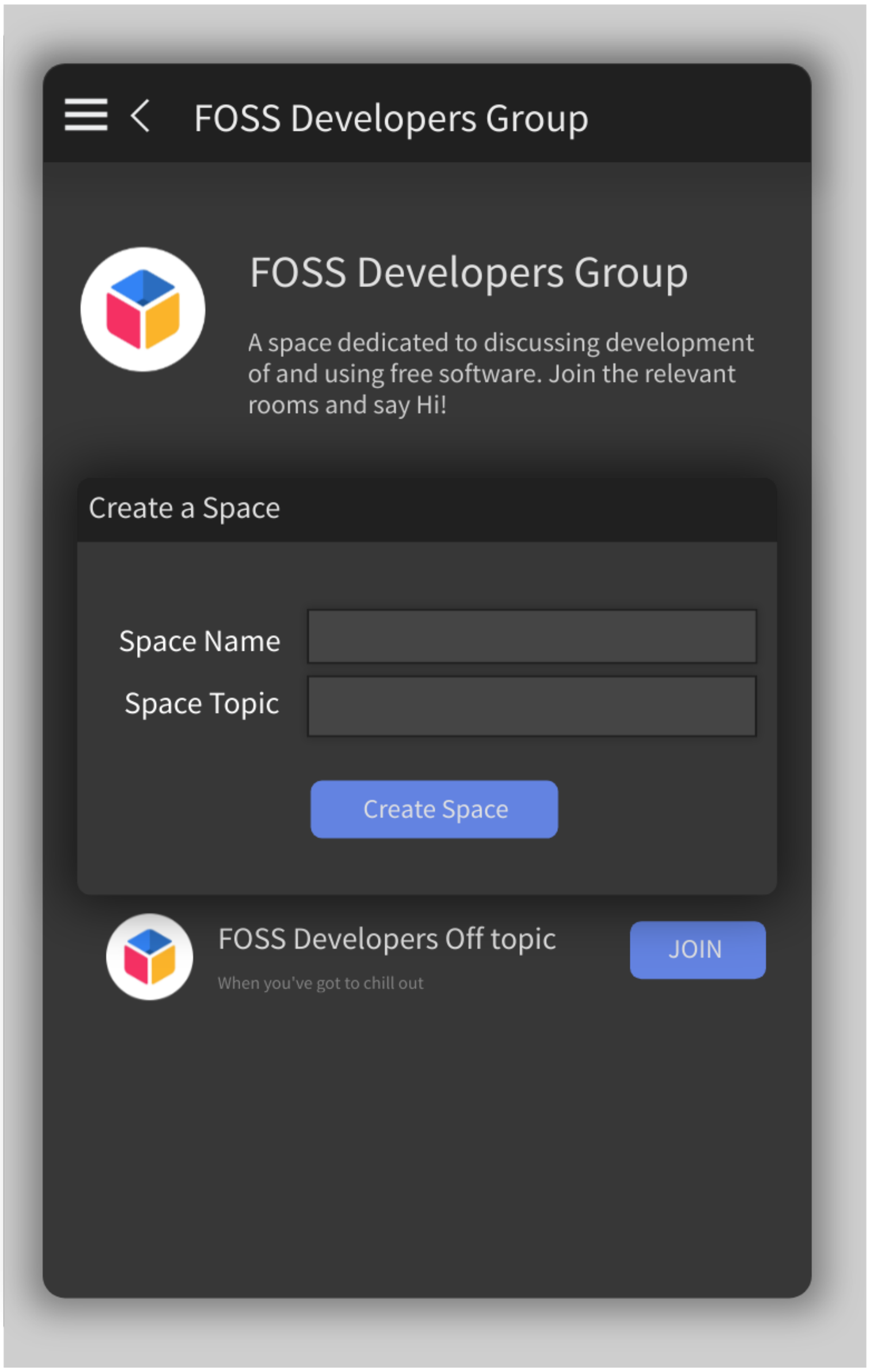
FOSS Developers Web
Help with web related tools

FOSS Developers UI/UX
Discuss your UI ideas

FOSS Developers CI
Head in the clouds!

FOSS Developers Off topic
When you've got to chill out

Fig 3: Viewing Space page on Mobile



FOSS Developers Group

A space dedicated to discussing development of and using free software. Join the relevant rooms and say Hi!

FOSS Developers Off topic

When you've got to chill out

Fig 4: Creating Space on Mobile

OTHER PROPOSALS AND TIME COMMITMENT

I have submitted no other proposal during Google Summer of Code 2022. If selected, I shall be available during the entire coding period to work on my project.

I shall be available to work for, on average, 40-50 hours per week. My schedule is flexible, and I am comfortable with working across time zones.

I have summer break, through mid May-July (~10 weeks). My classes will start in August. It will, however, not affect the time I devote to my project. I have already demonstrated the same while working on my Season of KDE project during regular class schedule.

BEYOND GSoC

I want to utilise Google Summer of Code as a platform to quickly get to grips with development on KDE applications. With my knowledge and experience gained during this time, I want to take up more tasks in future and maintain features. The experience will also help me do much needed bug fixes in NeoChat and other KDE applications too.

KDE provides the kind of software that I believe in, and I would be grateful to be a gear in its working.

ABOUT ME

I am Snehit Sah, a second year Computer Engineering undergrad. I have been using FOSS for around 7 years now. I take interest in working on user-facing applications. As a result, in 2020, I started off by contributing to testing and theming on **Endeavour OS**, the distribution I used.

In 2022, as a part of **Season of KDE**, I helped KDE with packaging and CI. I submitted applications to Flathub and updated all packages with metadata to automate updates. Over on KDE Invent, I worked on utilising the same metadata in GitLab CI pipelines, to have the latest dependencies in nightly builds. All through, I made over **150 merged pull requests**.

Contributions to KDE

1. Improving subtitle display in **NeoChat**
   https://x.snehit.dev/neochat-subtitle-pr

2. Updating **KDF** to work better with system file manager
   https://x.snehit.dev/kdf-pr-1
   https://x.snehit.dev/kdf-pr-2

3. **Season of KDE** 2022 blog:
   https://x.snehit.dev/sok-blog/

4. KDE packages submitted to **Flathub**:
   https://x.snehit.dev/flathub

5. All pull requests to **KDE** apps on **Flathub** (requires GitHub sign in to view):
   https://x.snehit.dev/all-kde-pr-gh

Contributions outside KDE

I am on the Endeavour OS team, maintaining Bspwm and Openbox editions. I help the project with user experience, theming, testing and community moderation.

In my free time, I like to work on small open source projects. Some of my projects include

1. **Git Notes**
   Web based notes to help new users to Git quickly learn commands. This usually accompanies the workshops I host but can be followed independently too.
   - Website: https://git-notes.snehit.dev
   - Repository: https://github.com/flyingcakes85/git-workshop-notes

2. **Pasta**
   A simple pastebin server written in Rust, with <100 LoC.
   - Repository: https://github.com/flyingcakes85/pasta

3. **Crabfetch**
   A fetch application with a crab icon for all Rust fans :)
   - Repository: https://github.com/flyingcakes85/crabfetch

REFERENCES

- **Timothée Ravier** - My mentor during Season of KDE 2022
  Contact: travier@redhat.com