Google Summer of Code 2023 [Polari]

Project: Implement backlog search in Polari IRC client

GENERAL DETAILS

Name: Gurmannat Sohal

Time Zone: India (UTC+5.30)

University: Indian Institute of Technology, Roorkee (IIT Roorkee)

Expected Graduation Date: May 2025

Primary Email: iamgurmannatsohal@gmail.com

Secondary Email: g_sohal@ee.iitr.ac.in

Matrix ID: @itsgurmannatsohal:matrix.org

IRC Nick: patronus

GitHub: itsgurmannatsohal

GitLab: itsgurmannatsohal

Size of the Project: Medium (175 Hours)

Description

https://gitlab.gnome.org/Teams/Engagement/internship-project-ideas/-/issues/12

This project aims to implement backlog search capabilities for the Polari IRC client. This will require writing the SPARQL queries to retrieve the underlying data and developing the user interface via the GTK toolkit. Interaction with the design team will be necessary to iterate and refine the UI.

The Polari IRC client's logging infrastructure has ported from Telepathy Logger to Tracker, providing users with more powerful and flexible search capabilities. Using Tracker to store and access chat logs will also make it easier for developers to maintain and extend the Polari IRC client in the future.

Searching in old conversations is a valued tool in free software IM channels; this is expected to enhance productivity when using IRC.

Mentors

1. @fmuellner (fmuellner@gnome.org)
2. @carlosg (carlosg@gnome.org)

What city and country will you reside in during the summer?

For most of the summer, I will reside in **Roorkee, Uttarakhand, India** and for some while in my hometown **Chandigarh, India**. But regardless of where I reside, I will always have access to fast internet.

What applications/libraries of GNOME will the proposed work modify or create?

The proposal aims to implement Polari IRC's backlog search feature. First, I will work on fetching the dump of the older messages on the Polari client, followed by the front-end development of the search bar and related icons. Then I will work on implementing the search feature. Firstly, it will be simply a keyword search feature that works at the click of a button, followed by, if time allows, a dynamic search feature that fetches relevant messages as we type.

What benefits does your proposed work have for GNOME and its community?

Backlog search in an internet relay chat, or for that matter, any instant messaging service, is a very valuable feature.

1. This feature can save time and increase productivity, especially in busy IRC channels where a lot of information is exchanged.
2. Adding new features to the Polari IRC client can encourage more users to engage with the GNOME community, potentially leading to more contributions and support.
3. Backlog search capabilities can enhance the user experience by making it easier to find relevant information and improving overall satisfaction.

Thus, adding such improvements implies that GNOME and its community would benefit greatly from the improved Polari IRC application experience and an increased user base for the platform.

How do you plan to achieve completion of your project?

Creating the frontend

The plan is to create a search widget frontend with the GTK toolkit in `main-window.ui` and add a search button to perform the search function. The GNOME Design Team would work on the design from scratch, which would be more of an iterative process. We would need to make another child in the existing object with the id `searchButton` and fetch the id in `main-window.js`. Various tags and classes exist to add functionality and CSS to the widget, for example, the `GtkToggleButton`, which can add toggle functionality to a tab.

Database connection changes

Database connection has been described in the Polari database in the `src/lib/polari-util.c` file. The `tracker_sparql_connection_new()` would be modified to establish the connection.

Fetching the history

For a start, we shall fetch the dump of all the historical messages. We will use SPARQL queries in `polari-util.c` similar to this in case the database used is the one that Polari is creating from the process itself, without an endpoint.

```
tracker3 sparql -d ~/.var/app/org.gnome.Polari/data/chatlogs-v2
-q "SELECT ?message WHERE { ?message a polari:Message }"
```

Tracker is being used as a database based on open standards like SPARQL and RDF data models. It automatically manages database format and schema updates. Users only need to write and update the ontology to define data structure.

Fetching relevant history

A general approach:

The SPARQL query would iterate over messages for the keyword provided. The results would be displayed across the screen and as the query reaches the end, with the keyword highlighted. A general query can be shown with:

```
SELECT ?message WHERE {
  ?message a polari:Message ;
           nmo:plainText ?text .
  FILTER (REGEX(?text, "keyword", "i"))
}
```

In further detail:

A more specific approach can be shown here. This function first builds a SPARQL query string using the input keyword, then uses the TrackerSparqlConnection API to connect to the Tracker store and execute the query.

```
 // Build the SPARQL query string
  query_builder =
    g_string_new (
    "SELECT ?message WHERE { ?message polari:text
?text . FILTER(CONTAINS(?text, \"" );

  g_string_append(query_builder, keyword);
  g_string_append(query_builder, "\")) }");

  query_string = g_string_free(query_builder,
FALSE );
```

```
  // Connect to the Tracker store
  connection =
tracker_sparql_connection_get_for_bus_sync
(G_BUS_TYPE_SESSION, NULL, NULL, &error);
  if (error != NULL) {
    g_printerr("Error: %s\n", error->message);
    g_error_free(error);
    return;
  }
```

```
  // Execute the query
  results = tracker_sparql_connection_query (
    connection, query_string, NULL, &error);

  if (error != NULL) {
    g_printerr("Error: %s\n", error-> message);
    g_error_free(error);
    g_free(query_string);
    return;
  }
```

This would return the message URI as a result and we would need to make another query to fetch the actual message content. Assuming we put in the URI as input, we could use the following function to get the actual text.

```
GString *query_builder = g_string_new(NULL);
  g_string_append(query_builder, "SELECT
nie:plainTextContent(?content) WHERE {");
  g_string_append(query_builder, "<");
  g_string_append(query_builder, message_uri);
  g_string_append(query_builder, ">");
  g_string_append(query_builder, " nfo:hasContent
?content . }");
```

This function takes a ‘message_uri’ as input and fetches the plain text content of the message using a SPARQL query. To display the result in the console, as shown below, it loops through the results and prints the message content.

```
  for (int i = 0; i < g_variant_n_children(results); i++) {
    GVariant *result = g_variant_get_child_value(results, i);

    const gchar *message_content = g_variant_get_string
(g_variant_get_child_value(result, 0), NULL);

    g_print("Message Content: %s\n", message_content);
  }
```

Searching dynamically

For the second half of the project, if time permits, we should work on implementing a dynamic search that displays results as we type. We would need to include an event handler that would trigger the search query on the addition or deletion of a character in the search widget. The search button will be discarded and if everything goes well, we can also work on making the search feature more user-friendly by adding the feature to search the messages from a particular user.

All in all, we will have a great search feature in the Polari IRC!

Please provide a sequence of tasks and subtasks and how long (days) you estimate it will take you to complete each of them. Highlight important milestones/deliverables.

Pre GSOC Period	
April 4 - May 4	• I would try completing the remaining issues assigned to me (if any) on Polari before the community bonding period. • If not, I would simply continue working on other existing issues to improve my understanding of the project.
Community Bonding Period	
May 4 - May 28	• With the help of my mentor, I'll help to familiarize myself with the community and the codebase. • I'll discuss potential ideas to solve the issue. • Work on different issues and bugs related to syncing to better transition into this project and hence, identify more relevant parts of the codebase.

Coding Period	
May 29 - June 9	• Build the required frontend i.e the search widget and buttons, as decided by the design team. • Working on various front-end iterations over time.
June 10 - June 20	Work on • Fetching the message dump • Displaying it in the console of the local setup.
June 21 - July 2	Work on • Keyword search, i.e fetching the relevant messages containing a particular keyword.
June 3 - July 10	In case of no pending work, • start work on displaying the above-described messages in the text channel • Prepare for mid-evaluation
Mid evaluation	
July 14 - July 24	• Continue work on displaying the above-described messages in the text channel
July 25 - August 5	Once the above is successfully completed and implementation tested in all channels, • Start work on dynamic search. • Work on executing the search function in case of any change in the search box.
August 6 - August 16	• Debugging and testing. • If completed early, work on making the feature more user-friendly by adding the feature to search for the messages of a particular user or in a particular timestamp.
August 17 - August 28	Work on • Pending or last-minute issues and prepare for final evaluation.
Final evaluation	

What are your past experiences with the open-source world as a user and as a contributor?

As a passionate open-source advocate, I have been using Fedora Desktop with GNOME interface since college started. I was initially amazed by the free availability of all these services and soon realised the significance of contributing to the open-source community. Hence, I have been actively giving back to the community ever since.

I am an active member of SDSLabs, a student-run technical group in my college and have contributed to various projects under the group, the details of which are given in the upcoming sections.

I've interacted with several people from the open-source community and had a great experience so far. I started contributing a few months ago, and have learnt a lot since then.

Why are you the right person to work on this project?

With more than a year of experience in various Javascript projects, I'm eager to broaden my knowledge by diving into new technologies. I'm confident that my expertise can meet the high standards required in open-source organizations. I firmly believe that I will be able to work effectively within a team, provide and receive constructive feedback, and adapt to changing project requirements.

I also have a good understanding of the repository, and file structure, and know where to ask for help. The opportunity to work alongside seasoned professionals as my mentors is an exciting prospect, as it will provide me with a valuable learning experience. I have also made quite many contributions to open-source and thus have a good experience in coding and problem-solving.

Being a student of an institute of national importance, I have learned to solve challenging problems, including algorithmic problems, and give my best in everything. When offered a chance to work on this project under the GNOME organization, I will surely give my best to make this project more usable.

Please include links to your code contributions that have already been merged, or to Gitlab merge requests for the issues you fixed for the project of your proposal or any other GNOME projects. This demonstrates your willingness to learn and familiarity with development workflow.

On GNOME-Polari, I have 2 open merge requests:

1. Grey out past discussions older than a day
2. Marks new days with dates and timestamps [WIP]

If available, please include links to any code you wrote for other open-source projects.

Fix map floating over footer [publiclab/plots2]

Replace docker run with apptainer exec [FNNDSC/ChRIS_ui]

Feed pagination [FNNDSC/ChRIS_ui]

Add functionality to disable button and remove edit [FNNDSC/ChRIS_ui]

What other relevant projects have you worked on previously, and what knowledge have you gained from working on them?

I have worked on various projects, encompassing both frontend and backend fields. I have experience in Javascript (React, Next, Node, Express), Typescript, Golang, C++, Django, Ruby on Rails, Mongoose and MySQL. A few projects are listed below.

Eris:

[https://github.com/sdslabs/eris]

- Nymeria is a unified authorisation and identity access management system developed using Ory Kratos, an open-source framework.
- Developed frontend for user registration and login flows, along with an admin panel to access logs using NextJS for Nymeria.
- Integrated the front end with Nymeria while writing all the routes and resolving bugs and helped deploy the app.

Odyssey:

[https://odyssey.iitr.ac.in/]

[https://github.com/odysseyiitr/website-2022]

- Worked on entirely redesigning the profile page and making it responsive.
- Worked on building and compiling the leaderboard pages single-handedly, along with several other front-end components.

Quizio:

[https://github.com/sdslabs/Quizio-Utils]

- Contributed by improving, testing and creating schemas as well as the corresponding routes and controllers for user and quiz creation, fetching and rendering.
- Improved correlation and rendering of data by adding checks and filters in the backend.

De_FL:

[https://github.com/def-learner]

- The project trains AI models using federated learning accurately on Blockchain.
- Implemented Push Protocol, a web3 communication network, to enable cross-chain notifications between wallets.

MetaVid:

[https://github.com/ETHForAll-DopingAgain/notification-system]

[https://github.com/ETHForAll-DopingAgain/client]

- Developed frontend for user registration and login flows, as well as dashboards to access videos and payment verification using NextJS.
- Implemented Push Protocol, a web3 communication network, to enable cross-chain notifications between wallets.

What other time commitments include school work, exams, research, another job, planned vacation, etc.? What are the dates for these commitments, and how many hours a week do these commitments take?

I can dedicate more than 40 hours a week. The college's summer vacations span June and July. Therefore, there would be no classes for the first two months of the program's coding period and no exams throughout the coding period. There are no exams or planned vacations, though I might have to travel to my hometown and back, which may cost a day or two. Once classes commence, I would be able to easily devote about 25 hours a week. I generally work from 4 PM to 12 midnight IST, although I find it flexible to adjust my working hours if required. Other than this project, I have no commitments. I shall keep my status posted to all the community members in case of any changes and maintain transparency in the project.