

Organic-maps



GSoC’24 Proposal: The Track Recorder for Android.

**By Sandilya Bukkapatnam** Section 1: About Me

## **What project am I applying for?**

Project: The Track Recorder for Android

## **Why am I interested in working with Organic-maps?**

Organic-maps’ mission is to provide offline and easy Navigation, Routing and for cyclists, hikers & tourists to look up places on map, based on categories (like Hotels, Petrol Bunks, ATMs etc) without the need for network. The best part is that it is free and open-source, pure and organic, respects users’ privacy, saves phone battery and no unexpected mobile data charges.

## **Why did I choose this project out of many ideas?**

* I chose this project out of the 10 GSOC 2024 ideas because this is one of the most requested features by many cyclists, hikers who have been loving using this mobile application. They want to track their rides on mountains and trails where perfect navigation might not be possible and use the recorded track later when they want to revisit the place.
* The features that I have worked previously on Organic-maps align with this feature which got me beforehand chance to work with one of the active contributors (@[**biodranik**](https://github.com/biodranik)**)** , I’ve learned a lot from his guidance and suggestions.
* Although we are not going to build this feature from scratch (we are going to retrieve the old feature and develop it), technologies like C++, JNI and Java which I’m very good at are strongly required.

# **My Prior experience:**

* My name is Sandilya Bukkapatnam, and I started Android Development in March 2022.
* I have been contributing to Organic-maps since October 2023 and have learnt a lot in these 6 months.
* I’m very active on GitHub brushing up my git commands and practicing daily on programming languages like Node JS, Kotlin, C++ etc. [GitHub commits tracker](https://github.com/sandy99405?tab=overview&from=2023-12-01&to=2023-12-31).

## **My Contributions at Organic-maps:**

PRs & Features I worked on:

1. [[Android] Added Search Privacy by sandy99405 · Pull Request #6515 · organicmaps/organicmaps (github.com)](https://github.com/organicmaps/organicmaps/pull/6515) – Feature to Clear the Search History and to enable and disable the feature.
2. [[Android] Fix Spanish and Portugal TTS by sandy99405 · Pull Request #6934 · organicmaps/organicmaps (github.com)](https://github.com/organicmaps/organicmaps/pull/6934) – Fix Spanish and Portugal TTS bug.
3. [Bulk delete and bulk move in bookmarks · Issue #6675 · organicmaps/organicmaps (github.com)](https://github.com/organicmaps/organicmaps/issues/6675) – Working on a new feature for bulk actions on Bookmarks (like moving, deleting, changing the colour) of selected bookmarks in a list.

Also working on a new feature called Delete Bulk Bookmarks lists that I will pitch them soon.

# **Contact info and time zone(s):**

Name: Bukkapatnam Sandilya

University: Lovely Professional University, Punjab

Currently working company: Cognizant Technology Solutions

Country: India

Email: sandysandilya99@gmail.com

GitHub: [sandy99405 (Bukkapatnam Sandilya) (github.com)](https://github.com/sandy99405/)

LinkedIn: [Bukkapatnam Sandilya | LinkedIn](https://www.linkedin.com/in/bukkapatnam-sandilya-4374aa204/)

Time zone: Indian Standard Time (IST) (+5:30 GMT)

Preferred method of communication: e-mail, GitHub or LinkedIn.

# **Time commitment:**

I am committed to spending 5-6 hours a day on this project on weekdays (Monday to Saturday). The time I devote to the project on Sunday will depend upon the work to be completed in the project in that week. In total, I am committed to working at least 35 hours a week.

# **Project size:**

Intermediate complexity (~175 hours)

# **Project timeframe:**

I will be working during the default GSoC coding period, May 27th – September 3rd.

# **Essential Prerequisites:**

* I’m successfully able to run the unit test on my machine via Android Studio.

A screenshot of a computer

Description automatically generated

* I’m successfully able to BUILD the project on my machine via Android Studio.

A screen shot of a computer

Description automatically generated

# **Other summer obligations:**

Since I’m a part time employee at Cognizant company, during the month of May I will be working in my company project as it started in the month of January. And that project timeline will be finished by end of May. Apart from that I don’t have any other obligations.

# **Communication channels:**

I am comfortable with any mode of communication that the mentor chooses, be it email, GitHub or LinkedIn. Mentors can expect a response from me in about an hour.

## Section 2:

Problem Statement:

|  |  |
| --- | --- |
| **Problems and challenges faced before** | The. Previous versions of the app had a background track recorder activated in the Settings menu. This recorder has stopped working on Android 8+ due to new background restrictions introduced in the system. |
| **Target Audience** | Cyclists & Hikers who have been using the Organic-maps android app to not get lost in the middle of nowhere during a ride as well as the new users. |
| **Core User Need** | The current Organic-maps Android app offers navigation to the users which would be useful when the GPS connections is strong. But during trails, that is not helpful because they might get lost when the connection is weak.  The track recorder feature will be added in the Android app, which will be easily visible and accessible to the users. |
| **What goals do we want the solution to achieve?** | The goal is to create a smooth recording experience to the user with a simple UI which doesn’t do background data collection. Using the foreground services such as a notification when the app is in background solves the issue of permissions.  The notification created should be light-weight easy to understand without cluttering all the data in it just by showing the pause/resume and stop buttons and the distance covered/yet to be covered.  The recorder should start automatically when the user starts a ride without having to start it manually (we are going to discuss and add a bi state switch in the settings whether to start the recording automatically/manually) and keeping the automatic recorder on by default.  We should be able to export the recorded files to other applications which says that the files are going to be in the form of GPX/KML. |
| **What are the main steps and precautions that we need to take to achieve this solution?** | We start this feature by retrieving the previous feature implementation (which has been removed dues to background permissions) and improve the UI so that the user can easily access the recording feature.  The project includes adding, removing, hiding and redesigning some UI elements. |

## **Section 2.1: WHAT?**

*This section enumerates what the requirements are that the technical solution outlines in “Section 2.2: HOW?”.*

UI/UX for the recorder:

## **Start Button**:

We will add a start button in the main screen as well as the navigation screen because recording is independent from navigation.

* Start button on general screen: Once the user starts a ride with the application running on the foreground, the user can manually start the recording by clicking on the button and expect the app to record wherever he/she is going the track will get saved at the end of the ride when the user clicks on the stop button.

This clearly brings the necessity for pause and stop button in the application (so we are going to discuss among ourselves and decide if the pause button is required or not).

* Start button on Navigation Screen: Once the user chooses a destination, starts the ride, and clicks on the Start recording button (if the automatic recording is off),

the recording happens and be saved just like the previous scenario.

## **Pause button**:

* Once the user starts a ride and stops the vehicle in the middle of the ride for a break and doesn’t want the recording to end, he/she can click on the pause button to pause the recording for a while until the user is ready to record again.
* This is very helpful if the user stops for a while as it avoids potential errors.

Some advantages of having a pause button while recording:

**It helps to exclude stopping points**: When a user stops the tracking during their activity, they can use the pause feature to exclude that time from the recorded activity. This can give the user a more accurate record of their performance, especially for activities where breaks or stops are common, such as cycling.

**It saves battery percentage**: When the tracking is paused by the user to take a break, the app stops using the GPS sensor, which help save the battery percentage.

**It prevents errors in tracking (Although the solution for the missing of tracking points will be explained later in the “Section 2.2 How”, the pause button still comes for the rescue)**: When the user goes through areas where GPS signals are weak or unavailable, the tracking may become inaccurate or even stop altogether. The pause feature allows the user to stop tracking during these times to avoid errors in tracking.

**It avoids cluttering of the data**: If there is no pause button and the user leaves the tracking on while they are not actually moving, the recorded data can become cluttered with unnecessary information. By using the pause feature, the user can avoid this and keep their recorded data clean and accurate.

## **Stop Button**:

* This is button is used to stop the recording.
* Even if the user doesn’t reach the destination, the record of it will be saved.

## **Automatic/Manual Track Recording Switch (This needs to be finalized by the mentors):**

* Some users have been asking for automatic start or the recording as in the past, many time they have forgotten to start the recording and later realized that the recording wasn’t happening.
* The mentors wanted the recording to happen automatically. But I have noticed a potential disadvantage coming with it. If the user doesn’t want to start the recording, unnecessary files get saved and may lead to a clumsy interface while displaying the files. So, I came up with a solution here.
* The user can choose between start track recording automatic and manual. We create a new setting in the general settings, where the automatic recording will be enabled by default and the switch can decide if it’s automatic/manual.

This will solve the problem of both forgetting as well as the storage of unnecessary files.

## **Notification Bar:**

* When the user starts the recording and the application is running in the background, the Notification Bar acts as a foreground service which avoids the running of background services which requires a lot of permissions from Android 8+.
* The Notification bar consists of a pause/resume button, a stop button, the distance that is covered and the distance that needs to be covered.
* The pause button as mentioned before brings many advantages with it will be turned to a resume button to resume the recording. It will be simple lightweight and provides a good User Experience.

## **Naming a Track After Finishing the Ride:**

* After completing a ride successfully and the user stops the recording, the app asks for a name already filled with a name in it. The name being “currentdate\_time\_start\_destination”.
* If the user wants to change the name, he/she can change it. If they don’t have the time to think of a name, they can just click on the save button and the track gets saved.
* This way, the user will later be able to access the recorded file with the date, time, start & destination points giving the file a unique name. (Note: The dialog box for name suggestion will come with the extension itself and even if the user doesn’t give the extension, the file gets saved with the extension.)

## **Track recorder storage:**

* After saving the track either by giving it a name or just clicking on the save button in the dialog box, the recorded file gets saved in the Bookmarks page.
* The user can access the files in the bookmarks page.
* The UI of the bookmarks Icon will be changed so that the User will be able to identify the icon being used for both tracks as well as bookmarks lists.

## **Recorded File Page UI:**

* Once the track has been saved, a page will be created for each file.
* The user can open the file and go through the recording, add highlights, pictures taken during the ride etc.
* This will give the good User Experience and allows us to add some more features to the Recorded file in the future.

## **Sharing the Recorded File:**

* Users can share the recorded file with the timestamps on each recorded point being included (either through the UI page of the file or directly from the Bookmarks List).
* It is useful to enable seamless integration with open mapping platforms for sharing and collaboration. Since OpenStreetMap rely on standardized track file formats like GPX, KML for importing and displaying track data.
* Including timestamps in the exported track files ensures compatibility with this as well as other platforms, allowing users to overlay their recorded tracks onto open maps for visualization and analysis.

## **Accurate track recording precautions:**

Sometimes when cyclists, hikers start the recording and go through the track, there are gaps found in the track when the recording is completed, this is due to weak signals in the hilly, mountain areas and trails. To avoid these, we make the track smooth using adjacent points, fill in the gaps and bring accuracy to the track so that the user can rely on the track without a doubt.

## **Section 2.1: How?**

* **First, we must retrieve the Previous Track recording feature from the Android code and then develop this feature on top of it.**
* **For the UI part we use Java, xml code and for the core functionality of the feature, we use JNI, C++ to avoid code duplication between Android and iOS applications.**

|  |  |  |
| --- | --- | --- |
| **Topic** | **How?** | **Reference UI screens from other apps/prototypes & screenshots of code snippets** |
| Start Button | The Start Button will be added to the regular as well as the navigation page as shown in the screenshots indicating that the recording will happen in both the cases. | Start Button UI in the regular layout:  A screenshot of a map  Description automatically generated  Start Button UI in the Navigation layout:  A screenshot of a map  Description automatically generated |
| Pause/Resume Button | The Pause button just like the start button will be placed in the regular layouts. This will replace the Start button once clicked on the Start button to start the recording.  The Resume button is used to resume the recording again and start the collection of data from the Location provider. | Pause Button UI    Resume Button UI |
| Stop Button | When the recording starts, below the pause button the stop button will appear. Which is used to both stop the recorded track as well as prompt the track name dialog. | Stop Button UI |
| Notification Bar | The Notification Bar consists of a pause button and a stop button when the app is running in the background. This will help avoid the permissions for Background Services, since the user will be accessing the notification which is in the foreground. | Example UI from Trekarta App. A screenshot of a phone  Description automatically generated |
| Automatic/Manual Track recording switch | Once the user starts a ride by fixing the destination and his relative position changes. The start button turns to pause, and a stop button will be appeared below the pause button.  Manual:  If he/she clicks the button manually, the same thing happens but the user should opt for the Manual recording setting in the general settings | Switch for the Automatic/Manual recording setting in organic-maps general settings. |
| Naming a Track | When the user finishes a ride and clicks on the stop button, it prompts a dialog box by suggesting a name with the **currdate\_time\_start\_destination** as the name of the track.  If the user wants to change the name, they can change and save the file.  I have created an algorithm which will generate a name for the recorded file with date, time, start point and destination point. | Example dialog box from organic-maps Bookmarks List.  A screenshot of a phone  Description automatically generated    Please Refer to Label 2.1.1  for the clear code snippet in the page 14. |
| Track recorder storage | When the user saves the dialog box, the file automatically gets saved in the Bookmarks and Tracks page.  The UI will be available parallel to the Bookmark lists and the user can access the track file by clicking on it to further see what happens inside the file.  I have come up with an icon to symbolize both tracks and bookmarks page so that the users will easily locate it. | Placement of Tracks in Bookmarks List page    UI Icon for Track in Bookmarks page:  A white star with green arrow and a green line  Description automatically generated |
| Recorded File Page UI | After testing an app called Komoot which is a track recording app I have observed that it has a UI page for the recorded track. User can add Highlights, add participants, edit the file name, see the track, select people who can view the track etc. | A screenshot of a map  Description automatically generated |
| Sharing the Recorded File | Sharing the Track file is a big priority for the organic-maps users. So, we record each point along the recorded track which includes a timestamp indicating when it is was recorded. The data provided from the location provider will be loaded and be saved in GPX format, so the user can export or share the file to other apps like open maps and access the track. | Example UI from Komoot app  A map with a blue line  Description automatically generated |
| Battery Life Conservation  **(Optional)** | Going a bit off topic, to save the battery life and get the track recording. We can use the Criteria class which will be used by LocationManager to automatically select the best provider based on our requirements. We use properties such as Criteria.POWER\_LOW which suggests the LocationManager to prefer a location provider that consumes less power.  Given that GPS is known for its relatively higher power consumption, setting the power requirement to low might lead the system to prioritize other location providers over GPS (like Network based location provider using cellular network towers etc,), since they consume less power. | Please refer to Label 2.1.2  for the clear code snippet  in the page 14.    Dialog box to disable the Battery saving mode to get the Accurate data from GPS(which is a power consuming Location provider)  A screenshot of a phone  Description automatically generated  Note : This is added surely . |
| Accurate track recording precautions | For accurate track recording, I will use an algorithm called Moving average. It will use the track points and a window size to optimize the track after recording. These two parameters will be passed in the java code to JNI method which acts as bridge between Java and C++.  Let’s name this Java/JNI method as **calculateMovingAverage** in Java and the same will be in C++. The C++ code will perform the algorithm provided the parameters, and we assign the output to another parameter called **smoothTrack** which contains all the track points and data to be collected in the GPX file. This is how we can avoid gaps due to signal issues in mountain roads and trails etc. | Please refer to Label 2.1.3  for the clear code snippet  in the page 15. |

Label 2.1.1

Code snippet for track Name generator:

A screen shot of a computer program

Description automatically generated

Label 2.1.2

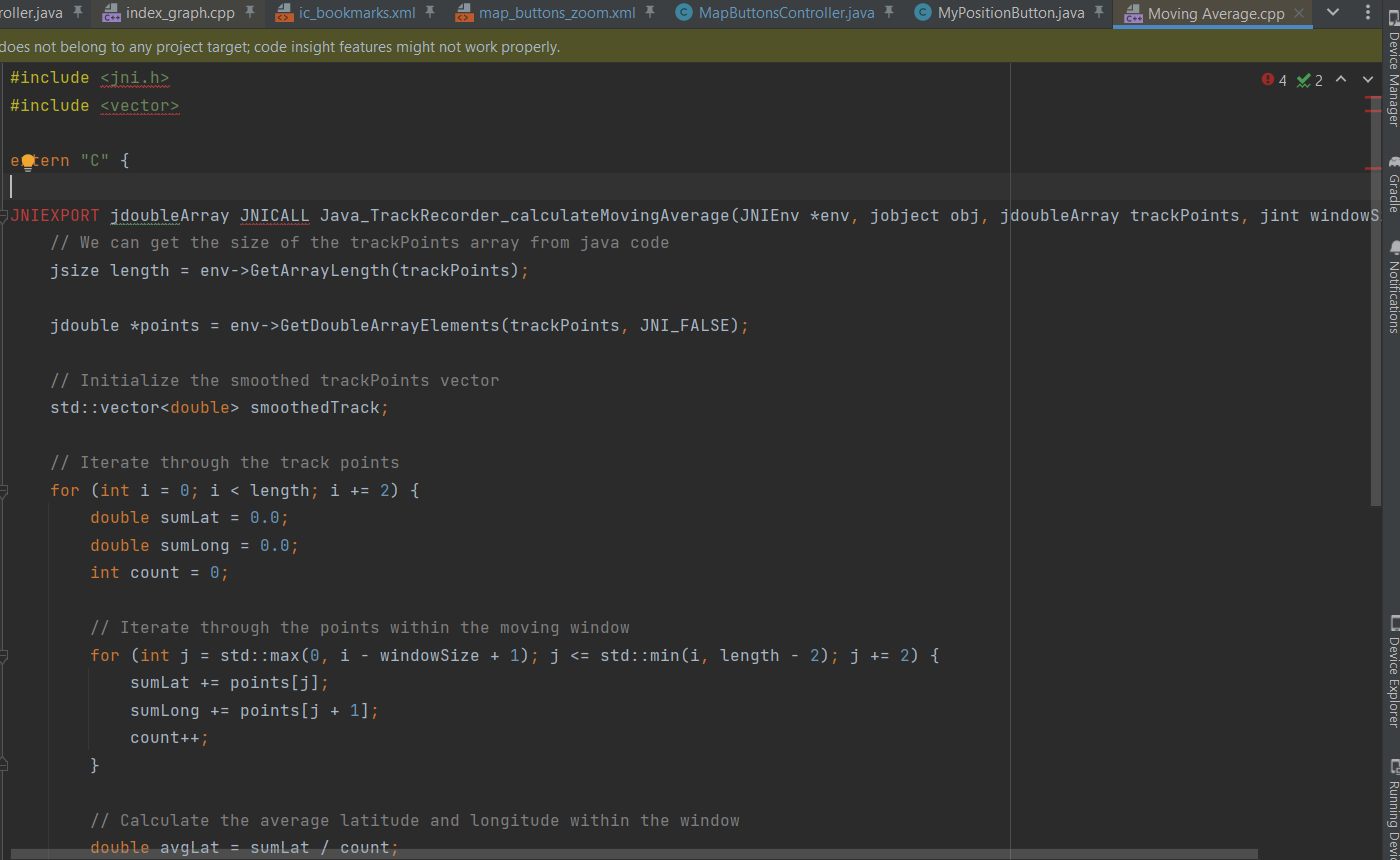
Code snippet for Batter saving:

A screen shot of a computer

Description automatically generated

Label 2.1.3

Code snippet for Moving average algorithm:



A screenshot of a computer program

Description automatically generated

## **Section 3: Milestones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Description of PR / action** | **Target date for work start** | **Target date for PR creation** | **Target date for PR to be merged** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1.1 | Retrieve the previous Track recording feature that was made for Android. We will retrieve all the files and functionalities in the commit [2bfb7ff](https://github.com/organicmaps/organicmaps/commit/2bfb7ffc0b1a060cf72d929ffd5d18779553d1aa) and develop from that.  Add unit tests | June 15 | June 15 | June 18 |
| 1.2 | Complete the discussion with the Mentors and Add additional UI components for Start, Pause, Resume, Stop, Automatic/Manual Switch, Notification Bar | July 8 | July 8 | July 12 |
| 1.3 | Complete the discussion with the Mentors and finalize the implementation of track design, colour of the track, background functionalities for all the UI components.  Add unit tests | July 23 | July 27 | August 2 |
| 1.4 | Adding the dialog box, implementing the naming algorithm, UI for the storing the track file in Bookmarks page, changing the icon of bookmark by discussing with the mentors.  Add unit tests | August 10 | August 10 | August 15 |
| 1.5 | Implementation of UI for Track File Page and algorithm for moving average  Add unit tests | August 17 | August 21 | August 24 |
| 1.6 | Add the functionalities for UI page of the recorded file, implementing the UI related to dialog box for battery saving, quality of tracking etc.  Add unit tests | August 26 | August 28 | September 4 |
| 1.7 | Add functionalities like file sharing and other miscellaneous  tasks that require sharpening the feature more and more, implementing new ideas etc. After discussing with the mentors | September 5 | September 8 | September 13 |

## **IDEs and codebases used for this development:**

* **Android Studio**
* **Apache NetBeans to test JNI and C++**
* **GitHub & git**

## **Future Scope:**

* I will be continuing to work on this feature with organicmaps and add few other functionalities like “**Spotlights for new users to identify the Track recorder UI precisely”.**
* UI for getting feedback from the user about track recording for further improvements.
* Upload Video recordings of users for better revision of the ride.