ADD LEADERBOARD BASED ON USER'S EDITS USING COMMONS ANDROID APP

Mentors : Ujjwal Agrawal, Vivek Maskara

—

Silky Priya

Outreachy applicant

Personal Information

Name : Silky Priya

IRC nick : Silky44

Email : silkypriya64@gmail.com

Github : https://github.com/silkypriya

Location : India

Time Zone : UTC +5:30

Typical working hours : Between 2 pm and 3 am UTC +5:30

About the App

The Wikimedia Commons Android app allows users to upload pictures from their Android phone/tablet to Wikimedia Commons. Wikimedia Commons is online repository of freely-licensed multimedia files and an independent project that seeks to document the world with photos, videos and recordings.

About the project

The project is about enhancing the already working Wikimedia Commons App by adding a leaderboard based on user's edits made with the mobile app using Commons Android App.

What to implement

❏ Leaderboard displaying user's Avatar and rank.

❏ Leaderboard showing 3 different results based on uploads, nearby and used.

❏ Filters in leaderboard screen and show results based on selected filter (weekly, monthly, yearly, all time)

❏ Open the gallery of that user's pictures by tapping the name of a user in the leaderboard

Benefits of the feature

❏ A leaderboard would motivate many people and can be a fun way to drive competition among users.

❏ Quantity and quality of uploads will increase by letting users see other people's contribution.

How it can be implemented?

1. Leaderboard based on uploads

- ❑ Make a screen with two tabs first tab as achievement activity and second tab is Leaderboard(Project) using APIs for tabs with Fragments and ViewPager.
- ❑ Design UI of the Leaderboard fragment and display user's Avatar and rank on Leaderboard screen fetched from API.

2. Fetch user list to display on leaderboard screen

- ❑ Use the APIs to fetch leaderboard (user list) based on uploads via mobile app (all time) and display it in Leaderboard screen.
- ❑ The list can be calculated on the server where currently achievement calculations are performed.
- ❑ Cache user's score for 1 hour.Only users who keep looking at the leaderboard would have their score grow at smaller intervals than 1 hour.

3. Add Pagination

- ❑ Add pagination in the leaderboard screen to load more users on scrolling down.
- ❑ This can be done by creating class PaginationAdapter extending RecyclerView.Adapter.

❑ PaginationAdapter helper methods will be useful for added data fetched via Pagination.

4. Add duration filter

❑ Add duration filters in leaderboard screen and show results based on selected filter (weekly, monthly, yearly, all time).

❑ This can be done using Spinners which provide a quick way to select one value from a set.

❑ In the default state, a spinner shows its currently selected value(for example "all time"). Touching the spinner displays a dropdown menu with all other options (weekly, monthly, yearly, all time), from which the user can select a new one. Spinner can be implemented using three key classes Spinner, SpinnerAdapter, AdapterView.OnItemSelectedListener.

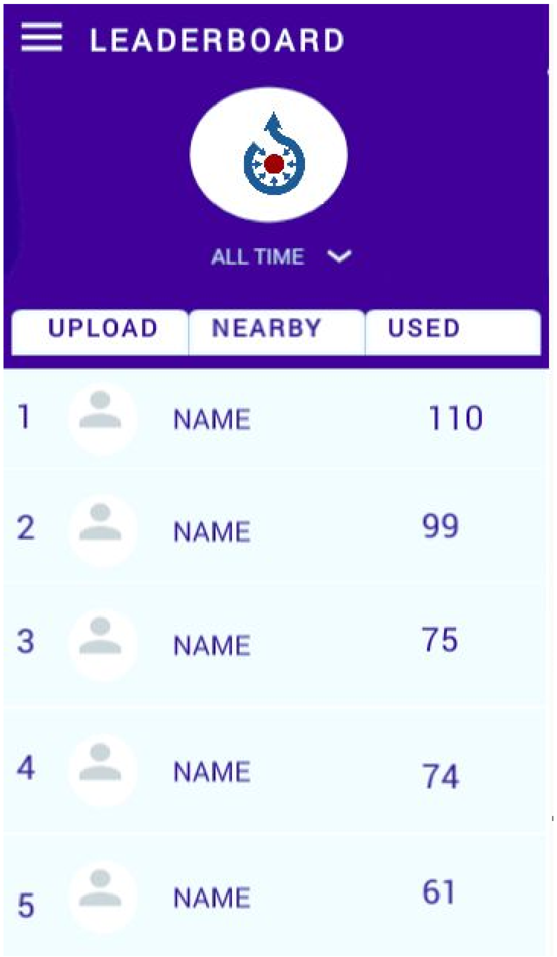
5. Show results based on uploads, nearby, used.

❑ Change the UI to add 2 more tabs in order to show 3 different results based on (uploads, nearby, used) using APIs for tabs with Fragments and ViewPager.

6. Open user’s gallery

- ❏ Tapping the name of a user in the leaderboard should open the gallery of that user's pictures.
- ❏ First fetch the URL of the gallery of the user's pictures using API call and open URL in android web browser.

7. Screenshot of the prototype



1	NAME	110
2	NAME	99
3	NAME	75
4	NAME	74
5	NAME	61

Time Table

Time Period	Task	Deliverables
May 20 to May 26 (7 Days)	Community bonding period ➢ Communicate and bond with students and mentors. ➢ Create specific issues for the project. ➢ Getting familiar with server where we currently perform achievement calculations and related APIs.	➢ Blogs about the initial experience, app architecture,community bonding ➢ Decide and design UI of the Leaderboard
May 27 to June 6 (10 Days)	➢ Add Leaderboard tab (based on uploads). ➢ Display user's Avatar and rank on Leaderboard screen fetched from API.	➢ Leaderboard displaying user's Avatar and rank.
June 7 to June 13 (7 Days)	➢ Use the APIs to fetch leaderboard (user list) based on uploads via mobile app (all	➢ Leaderboard displaying user list based on

	time) and display it in Leaderboard screen.	uploads.
June 14 to June 20 (7 Days)	➢ Add pagination in the leaderboard screen to load more users on scrolling down.	➢ Leaderboard with dynamic scrolling.
June 21 to June 30 (10 Days)	➢ Change the UI to show and select filter. ➢ Add duration filters in leaderboard screen and show results based on selected filter (weekly, monthly, yearly, all time).	➢ Leaderboard show results based on selected time frame.
	MID-EVALUATION	
July 1 to July 15 (15 Days)	➢ Change the UI to add 2 more tabs in order to show 3 different results based on (uploads, nearby, used)	➢ Leaderboard based on uploads, nearby and used.

July 16 to July 25 (10 Days)	➢ Tapping the name of a user in the leaderboard should open the gallery of that user's pictures.	➢ Open user gallery in android web browser.
July 26 to August 9 (15 Days)	Improvements based on the feedback received from mentors and other community members. Writing unit test for above implemented modules. Manual testing on different devices, emulators. Bug fixes, Writing documentation.	➢ Documentation of above modules. Unit test for implemented modules.
August 10 to August 20 (11 Days)	➢ Avatar is the user's most successful picture (the one which is used the most in the wikipedias/etc, or a random one if the user's pictures are not used yet anywhere). ➢ Add option to select one of the user's uploads as an avatar picture.	➢ Add option (an privilege to unlock) for users to personalize their Avatar.

	FINAL EVALUATION	

Participation

Workflow

- ❏ I will document detailed progress through weekly blog posts.
- ❏ I will be working on a separate branch on git and uploading code to the forked repo almost on a daily basis, will be Creating pull requests when a complete feature is done.
- ❏ During testing and debugging Period, I will request the beta testers to test the app and get feedback from community.
- ❏ Can be contacted via hangouts, mails in my non working hours

Communication

- ❏ Flexible with working hours
- ❏ Can be contacted via hangouts, mails in my non working hours
- ❏ Will remain online on IRC, Hangouts during my working hours
- ❏ Between 2 pm and 3 am UTC +5:30

About Me

Education and interest

Currently in the final year of B.Tech in Computer Science and Engineering in Indian Institute of Technology, Roorkee. I will graduate in May 2019. I have interest in Android App development.

How did I hear about this program?

A friend of mine, Ankita Saxena, encouraged me to participate in this program.She is an Outreachy 2017 alumuna.

Time during Summers

I will have my final exams over in April and will graduate in May 2019. I have no other commitments this summer, so I will work for full time for the whole internship period.

Are you planning to apply to both Google Summer of Code and Outreachy programs, if so, with what organization(s)?

I am only applying for outreachy.

What does making this project happen mean to you?

This was one of the first community I started contributing to. The helpful and highly skilled mentors motivated me to contribute and explore open source development. It is a huge opportunity to android development enthusiasts like me. Learning and contributing to the project that seeks to document the world with photos is great.

Past Experience

- ❏ I have gained experience with JAVA, PYTHON, HTML, CSS, JAVASCRIPT, GIT by working in college Technical teams.
- ❏ I've been doing Android development for the last one year.
- ❏ I have experience of clear and concise writing of communications and development documentation.

Contributions to Wikimedia Commons App

Merged Pull Requests:

https://github.com/commons-app/apps-android-commons/pull/2602

https://github.com/commons-app/apps-android-commons/pull/2559

https://github.com/commons-app/apps-android-commons/pull/2577

https://github.com/commons-app/apps-android-commons/pull/2574

https://github.com/commons-app/apps-android-commons/pull/2570

https://github.com/commons-app/apps-android-commons/pull/2557

Contributions to Wikimedia

API:Block: Improved the documentation and wrote sample code to demonstrate its use in blocking and unblocking the user

Documentation: **Approved**

Sample code: **Merged**

API:Emailuser: Improved the documentation and wrote sample code to demonstrate its use in sending email to other user

Documentation: **Approved**

Sample code: **Merged**