Wikimedia Commons App

Allows users to browse commons repository from Commons android app

Mentor : Neslihan

Co-Mentor : Nicolas Raoul

27.03.2018

Ujjwal Agrawal
IIT ISM Dhanbad
Dhanbad, Jharkhand
India — 826004

Contents

Personal Details

Name	Ujjwal Agrawal
Email	ujjwal.iitism@gmail.com
Mobile	+91 87 707 76846
Time Zone	UTC +5:30
GitHub Handle	@ujjwalagrawal17
IRC Nick	ujjwalagrawal17
University	Indian Institute of Technology (Indian School of Mines), Dhanbad
Blog	https://ujjwalagrawal.wordpress.com/
Typical Working Hours	Between 1pm and 11pm UTC +5:30
Other Contact Methods	Google Hangouts, Gitter Chat

Synopsis

About the App

The Wikimedia Commons Android app allows users to upload pictures from their Android phone/tablet to Wikimedia Commons. Wikimedia commons accepts only freely licensed media files (that are not subject to any copyright). Users can upload images and then add various tags specific to them such as category, title, description, and license.

About the Project

The goal of this project is to enhance the already working Wikimedia Commons App by implementing an option to browse/search the Wikimedia Commons Repository.

Why this feature is needed?

- ❏ More and more users are asking (as evident by Play Store feedback)to be able to also browse existing Commons images via the app
- ❏ Quality of uploads will increase by letting users see other people's work

What can be implemented?

- ❏ Image Search on Wikimedia Commons using a text
- ❏ Restrict the search results to images
- ❏ Category Search
- ❏ Previous Search queries of images and categories (Recent Search history)
- ❏ Shows the sub-categories and the images present in that category
- ❏ Allow users to skip login and browse and search in the app

How can it be implemented?

- ❏ Explore Mediawiki APIs to search images using titles, search categories (Probably using Mediawiki Search API)
- ❏ Implement search activity which will contain 2 tabs
  - ❏ tab 1 for showing image list (if possible reusing featured images fragment)
  - ❏ tab 2 for showing category list
- ❏ Implement async call requests to request search pictures, search categories dynamically on page scroll
- ❏ Develop feature for displaying search results in search activity
- ❏ Saving the clicked category/images in the local database
- ❏ Showing in the recently searched list (both categories and images) if a search query is empty
- ❏ Implement the category activity which will contain 2 tabs
  - ❏ tab 1 for showing sub-category list
  - ❏ tab 2 for showing images inside category (if possible reusing featured images fragment)

- ❏ Improving category activity by showing best images (images inside subcategory) using FastCCI
- ❏ Add an option to skip login in Login Page which redirects to Featured Image Activity

Prototype

I've been working on this feature for some time now. I made a small prototype for the same.

- ❏ **Description**
  - ❏ *Added a Separate activity for search in which an edit text is present.*
  - ❏ *Implemented a viewpager along with a tab layout in the activity*
  - ❏ *Added Browse Image fragment, Browse Category Fragment in the item of the viewpager.*
  - ❏ *Now if a text is changed in the activity I call update list method in both the fragments with search query as a parameter.*
  - ❏ *Method update list updates the list by fetching details from Wikimedia API: Search*
  - ❏ *For updating the list in recyclerview I had added AdapterFactory, Renderer similar to other modules.*
- ❏ **Video Link:** https://goo.gl/7XUv9w
- ❏ **GitHub Link:** https://github.com/ujjwalagrawal17/apps-android-commons/tree/browse

Timeline

Period	Task	Deliverables
(Community Bonding Period) *April 23 - May 14*	➔ Communicate and bond with students and mentors ➔ Create specific issues for the project ➔ Getting familiar with the app architecture and Wikimedia APIs ➔ Learning RxJava ➔ Getting Familiar with featured images scrollable activity, fragment implementation	➔ Community bonding report ➔ Blogs about the experience, app architecture
(Week 1) *May 14 - May 20*	➔ Add search button/layout in featured image activity ➔ Decide and create search activity with mock data.	➔ Search layout in featured image activity ➔ Search activity with mock data

	➔ Link Search activity, Design XML for the activity	
(Week 2) *May 21 - May 27*	➔ Add a new instance of featured image scrollable fragment in search activity/fragment. ➔ Use the Mediawiki Search API to search image list results using text (search query). ➔ Modify featured image scrollable fragment to show search image results. ➔ The list will load Search results dynamically on page scroll. Using recyclerview onScrollChangeListener callback we will fetch new image list with offset and add it to previous list.	➔ Images search feature with dynamic scrolling

(Week 3) *May 28 - June 4*	➔ Create tables in local database (SqLite) for storing recently searched images ➔ Store search results in the local database to show search history (previously opened images). ➔ Restrict the search results to images ➔ Decide and create UI for recent images searches (image title and timer icon) ➔ Use content providers similar to already existing Category ContentProvider to Show image search history (previous search queries) if the search query is null	➔ Previously searched images history (recently searched images)
(Week 4) *June 5 to June 11*	➔ Improvements based on the feedback received from mentors, other community members,	➔ Documentation of above modules ➔ Unit test for image search, search history

	→ Writing unit test for above implemented modules → Manual testing on different devices, emulators → Bug fixes, Writing documentation.	
Phase I Evaluation *June 11 – 15*		
(Week 5) *June 12 to June 18*	→ Modify layout of search activity to show both image list, category list fragment in it.(using viewpager and tablayout) → Decide and create category search fragment using mock data	→ Improvements in Search Activity UI → Category Search with Mock Data
(Week 6) *June 19 to June 25*	→ Implement category search using <u>Mediawiki Search API</u> → Show it in category search fragment.	→ Category search feature with dynamic scrolling

	➔ Make the search dynamic similar to image search.	
(Week 7) *June 26 to July 2*	➔ Create tables in local database (SqLite) for storing recently searched categories ➔ Store search results in the local database to show search history (previously opened categories). ➔ Decide and create UI for recent category searches (category name and timer icon) ➔ Show category search history (previous search queries) if the search query is null	➔ Previously searched category history (recently searched categories)
(Week 8) *July 3 to July 9*	➔ Start exploring APIs that can fetch category details (list of subcategories, media inside that categories).	➔ Link to finalized API to fetch category details and fields that will be fetched, ➔ Category activity with mock data

	➔ Decide and Create UI mock up for Category Activity (2 tabs with viewpager) ➔ One fragment for showing subcategory list, other fragment for showing images	
Phase II Evaluation *Aug 9 – 13*		
(Week 9) *July 10 to July 16*	➔ Implement category details using relevant APIs. ➔ Implement the category activity, which shows the sub-categories and the images present in that category (using featured image fragment to show category image list here also will reuse the code and the functionality of viewing media details)	➔ Fragment 1 to show list of subcategories in fragment one showing all at once as a category doesn't have more than 50 elements total (usually). Sorted alphabetically. ➔ Fragment 2 to show list of images in a category(reusing featured image fragment)

(Week 10) *July 17 to June 23*	➔ 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 category search, search history, sub-category list, images inside category.
(Week 11) *July 24 to July 30*	➔ Read about <u>FastCCI</u> / Search other APIs to show the best images first.(images inside subcategories) ➔ Add spinner/menu and add 4 options (Good Pictures, Featured pictures, Quality Images, Valued Images)	➔ Link to finalized API to fetch images inside subcategories ➔ Improvements in Category Activity UI by adding spinner/menu

(Week 12) *July 31 to August 6*	➔ Modify existing APIs for fetching new image list filtered according to item selected in menu/spinner ➔ Add skip Option in Login which redirects to Featured Image Activity ➔ Making changes in navigation base activity to hide items (my contributions, logout, other items which needs login) in navigation drawer and showing a login option in navigation drawer which takes the user to login page if user is not logged in (by checking from shared preferences)	➔ Filtered Image list in category activity according to FastCCI ➔ Skip Option in Login
(Final Week- Students submit their final work product and their final mentor evaluation) *August 7 to August 13*	➔ Write Unit test for FastCCI implementation ➔ Improvements based on the feedback	➔ Unit test for FastCCI ➔ Project Presentation

	received from mentors, other community members ➔ Manual testing on different devices, emulators ➔ Writing documentation for final submission. ➔ Working on Project Presentation.	
Final Evaluation *Aug 14 – 21*		
Final results of Google Summer of Code 2018 announced *Aug 22*		

Other Deliverables

❏ Weekly report and blog

Participation

Progress Report

- I will remain online on IRC, hangouts in my working hours (1pm to 11pm UTC +5:30)
- I will write weekly blog posts at (https://ujjwalagrawal.wordpress.com/)
- I will share my blogs on twitter

- Write weekly scrum reports and update it to our mailing list [commons-app-android@googlegroups.com]
  - What did I do last week?
  - What will I do this week?
  - What is currently preventing me from reaching goals?
- I will submit a Project Presentation

Where I plan to publish my source code

- I will be working on a separate branch on git and uploading code to the forked repository almost on a daily basis, will be Creating pull requests when a complete feature is done.

Communication on task

- I will use GitHub to manage bugs and task.

About Me

Personal Background

I am a third-year B. Tech. Undergraduate at Indian Institute of Technology (ISM), Dhanbad. I am pursuing Electronics and Communication Engineering as my Major. I have a keen interest in Android App development (I have been doing Android App Development from last 2 years). I use git and GitHub every day, and I am well acquainted with using them for version control.

How did I hear about this program ?

I heard about GSoC an year back in a college meetup. I am an open source enthusiast from past 1 year and I always wanted to take part in Google Summer of Code.

Time during Summers

I have no other commitments this summer. So I'll be able to give 40 hours or more per week. My summer break starts from 27th April, so I can start working full time from that day on. I'll not be

taking any vacations. My classes start on around 20th July but I will be able to commit enough time for the project as there are no exams during the period

Eligible for Google Summer of Code and Outreachy ?

I am applying only for GSoC as I am not eligible for Outreachy program. I am applying under Wikimedia Commons Android only.

What excites me about this project

I had developed a lot of apps during my college days, but I always wanted to develop apps that really help people. I loved the goal of Wikimedia Foundation "Global movement whose mission is to bring free educational content to the world".

It would be really great for me to apply my skills and contribute to such an organization.

Past Experience

I've been doing Android development since my first year at the college. My first project was a home automation app to control LED lights using our mobile phone (Raspberry Pi behaving as a server) that I built in a hackathon. After that, I developed various apps for my learning purpose. Apart from the prototypes, I had also developed 2 apps that are on Google Play Store.

Relevant Projects

Spectrum 18:

I had developed this app for a departmental fest of the Dept. of Electronics Engineering (Spectrum), at the college. It has features like login, OTP verification using the phone number, show list of events day wise, can register users for events, send notification for events.

- ❏ **Google Play Link:**

  https://play.google.com/store/apps/details?id=com.ujjwalagrawal.spectrum&hl=en

- ❏ **GitHub Link:** https://github.com/Tech-ISM/spectrum-android-app

Brand Store:

It is an app where users can get offers from nearby shops of different categories across their city. Shops can also register them and show their offers to the users of the app. Although the startup is closed now, the app is still on google play.

- ❏ **Google Play Link:** https://play.google.com/store/apps/details?id=com.brandstore

- ❏ **GitHub Link:** https://github.com/CodeNicely/OfferCartApp

Contributions to Wikimedia Commons App

It has been an enriching experience contributing to the app and I look forward to continuing contributing to it. For all I have learned so far is that your contributions are not just the number of Pull Requests that you've got merged. But contributions can be in various forms like:

- ❏ Creating New Issues

- ❏ Finding Bugs

- ❏ Helping New Contributors

- ❏ Communicating with Moderators and helping them

Pull Requests:

- **Merged Pull Requests (8/12):**

  - Change toast to snackbar in App

  - Update Issue Template

  - Improvement in Login UI

  - FAQs added in About Activity

  - Rate Us feature

  - Changed WebView to CustomTabs in App

  - Created Template for Pull Request

  - Created Template for Issue

- **Closed Pull Requests:**

  - Change Toolbar color

  - Add Option to delete the contributed image

- **Open Pull Requests**

Issues:

- **Closed Issues (10/16):**

    - Add FAQs page in Commons App

    - Add Option to donate to Wikimedia Foundation

    - Change toolbar color to primary color

    - Rate Us feature in navigation drawer

    - Adding PULL_REQUEST_TEMPLATE.md file

    - Adding ISSUE_TEMPLATE.md file

    - Change Web views to Chrome Custom Tabs in App

    - Option to Change Recent Uploads Design

    - Add Option to delete the uploaded image

    - *Change LoginActivity Textview to Snackbar*

- **Open Issues**