



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

Contents	2
Personal Details	4
Synopsis	4
About the App	4
About the Project	4
Why this feature is needed?	4
What can be implemented?	5
How can it be implemented?	5
Prototype	6
Timeline	7
Community Bonding Period	7
Week 1	7
Week 2	8
Week 3	8
Week 4	9
Phase I Evaluation	9
Week 5	9
Week 6	10
Week 7	10
Week 8	10
Phase II Evaluation	11
Week 9	11
Week 10	11
Week 11	12
Week 12	13
Final Week- Students submit their final work product and their final mentor evaluation)	13
Final Evaluation	13
Final results of Google Summer of Code 2018 announced	13
Other Deliverables	14

Participation	14
Progress Report	14
Where I plan to publish my source code	14
Communication on task	14
About Me	14
Personal Background	14
How did I hear about this program ?	15
Time during Summers	15
Eligible for Google Summer of Code and Outreachy ?	15
What excites me about this project?	15
Why should I be selected for the project ?	15
Past Experience	16
Relevant Projects	16
Contributions to Wikimedia Commons App	17

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

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. We want to provide a streamlined and consistent user experience across the website and the app.

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 and categories dynamically on page scroll and displaying results in the respective fragments.
- ❑ 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 the category (if possible reusing featured images fragment)
- ❑ Add an option to skip login in Login Page which redirects to Featured Image Activity

- ❑ If time allows, improving category activity by showing best images (images inside subcategory) using FastCCI

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

<i>Period</i>	<i>Task</i>	<i>Deliverables</i>
(Community Bonding Period) <i>April 23 - May 14</i>	<ul style="list-style-type: none"> → Communicate and bond with students and mentors → Create specific issues for the project → Getting familiar with the app architecture and Wikimedia APIs → Getting Familiar with featured images scrollable activity, fragment implementation → Link featured images fragment with already existing media details page → making changes in media details fragment for extra details like author name 	<ul style="list-style-type: none"> → Community bonding report → Blogs about the experience, app architecture
(Week 1) <i>May 14 - May 20</i>	<ul style="list-style-type: none"> → Add search button/layout in featured image activity → Decide UI and create search activity with mock data. → Link Search activity, Design XML for the activity 	<ul style="list-style-type: none"> → Search layout in featured image activity → Search activity with mock data

<p>(Week 2)</p> <p>May 21 - May 27</p>	<ul style="list-style-type: none">→ Add a new instance of featured image scrollable fragment in search activity..→ 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 <code>recyclerview onScrollChangeListener</code> callback we will fetch new image list with offset and add it to previous list.	<ul style="list-style-type: none">→ Images search feature with dynamic scrolling
<p>(Week 3)</p> <p>May 28 - June 4</p>	<ul style="list-style-type: none">→ 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	<ul style="list-style-type: none">→ Previously searched images history (recently searched images)

	<ul style="list-style-type: none"> → 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 	
<p>(Week 4)</p> <p><i>June 5 to June 11</i></p>	<ul style="list-style-type: none"> → Improvements based on the feedback received from mentors, other community members, → Writing unit test for above implemented modules → Manual exhaustive testing on different devices, emulators → Bug fixes, Writing documentation. 	<ul style="list-style-type: none"> → Documentation of above modules → Unit test for image search, search history
<p>Phase I Evaluation</p> <p><i>June 11 – 15</i></p>		
<p>(Week 5)</p> <p><i>June 12 to June 18</i></p>	<ul style="list-style-type: none"> → Modify layout of search activity to show both image list, category list fragment in it.(using viewpager and tablayout) 	<ul style="list-style-type: none"> → Improvements in Search Activity UI → Category Search with Mock Data

	<ul style="list-style-type: none"> → Decide and create category search fragment using mock data 	
<p>(Week 6)</p> <p>June 19 to June 25</p>	<ul style="list-style-type: none"> → Implement category search using <u>Mediawiki Search API</u> → Show results in category search fragment (only category name) → Make the search dynamic similar to image search. 	<ul style="list-style-type: none"> → Category search feature with dynamic scrolling
<p>(Week 7)</p> <p>June 26 to July 2</p>	<ul style="list-style-type: none"> → 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 	<ul style="list-style-type: none"> → Previously searched category history (recently searched categories)
<p>(Week 8)</p> <p>July 3 to July 9</p>	<ul style="list-style-type: none"> → Start exploring APIs that can fetch category details (list of 	<ul style="list-style-type: none"> → Link to finalized API to fetch category details

	<p>subcategories, media inside that categories).</p> <ul style="list-style-type: none"> → Decide and Create UI mock up for Category Activity (2 tabs with viewpager) → One fragment for showing subcategory list, other fragment for showing images 	<p>and fields that will be fetched,</p> <ul style="list-style-type: none"> → Category details activity with mock data
<p>Phase II Evaluation</p> <p><i>Aug 9 – 13</i></p>		
<p>(Week 9)</p> <p><i>July 10 to July 16</i></p>	<ul style="list-style-type: none"> → Fetch category details using relevant APIs. → Implement Fragment 1 to show list of subcategories in fragment 1 showing all at once. No category has more than 50 elements in total, so displaying all at once would be better. Sorted alphabetically. → <i>Implement</i> Fragment 2 to show list of images in a category (reusing featured image fragment) 	<ul style="list-style-type: none"> → Category Activity with list of subcategories, images with dynamic scrolling feature.
<p>(Week 10)</p> <p><i>July 17 to June 23</i></p>	<ul style="list-style-type: none"> → Improvements based on the feedback received from 	<ul style="list-style-type: none"> → Documentation of above modules

	<p>mentors and other community members</p> <ul style="list-style-type: none"> → Writing unit test for above implemented modules → Manual exhaustive testing on different devices, emulators → Bug fixes, Writing documentation. 	<ul style="list-style-type: none"> → Unit test for category search, search history, sub-category list, images inside category
<p>(Week 11)</p> <p>July 24 to July 30</p>	<ul style="list-style-type: none"> → 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) → Read about <u>FastCCl</u> / Search other APIs to show the best images first.(images inside subcategories) → Add spinner/menu and add 4 options (Good Pictures, 	<ul style="list-style-type: none"> → Skip Option in Login → Link to finalized API to fetch images inside subcategories → Improvements in Category Activity UI by adding spinner/menu

	Featured pictures, Quality Images, Valued Images)	
<p>(Week 12)</p> <p>July 31 to August 6</p>	<p>→ If time allows, Modify existing APIs for fetching new image list filtered according to item selected in menu/spinner</p>	<p>→ If time allows, Filtered Image list in category activity according to FastCCI</p>
<p>(Final Week- Students submit their final work product and their final mentor evaluation)</p> <p>August 7 to August 13</p>	<p>→ Write Unit test for FastCCI implementation</p> <p>→ Working on Project Presentation.</p> <p>→ Improvements based on the feedback received from mentors, other community members</p> <p>→ Manual exhaustive testing on different devices, emulators</p> <p>→ Writing documentation for final submission.</p>	<p>→ Unit test for FastCCI</p> <p>→ Project Presentation</p>
<p>Final Evaluation</p> <p>Aug 14 – 21</p>		
<p>Final results of Google Summer of Code 2018 announced</p> <p>Aug 22</p>		

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, 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 am an active member of the Cyber society at our institute. I organize weekly Android Development Workshops headed by the society.

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. I am ready to commit extra time if needed in order to finish up the goals of the project. 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?

The kind of exposure and experience a platform like this would provide me is a huge reason for me to want to be a part of it. I have developed a lot of apps during my college days, but I always wanted to develop apps that really help people at a global scale. I find nothing more exciting than working for a company like Google with a goal as impactful as this one. I sincerely love the goal of Wikimedia Foundation "Global movement whose mission is to bring free educational content to the world" and would be more than happy to work towards it.

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

Why should I be selected for the project ?

I have always been interested in open-source projects and have been passionately working on them. I can push myself to the boundaries, come out of my comfort zone and work things out. I am a very good team-player and can learn thing quickly and adapt myself.

I have been developing Android apps for more than 2 years now and having contributed to various open source projects, I believe I have the required skills to finish the proposed goals of the proposal. Moreover, having enrolled in Google's Android developer nanodegree in Udacity, I believe I will follow the best practices in implementing the goals of the proposal.

I have been contributing to Wikimedia Commons app for over 45 days now and will be contributing to it even after the GSoC period. I have even made a working prototype which required the knowledge of app architecture and MediaWiki API.

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 several prototypes, I also published four apps on the Google Play Store. I have open-sourced two of the above apps. These projects required skills ranging from server-end development (Django) to database management (MySQL). I use git and GitHub every day, and I am well acquainted with using them for version control.

Relevant Projects

Easy Invoice Manager App :

GST Invoice Creator provides an Easy way to create Invoices after GST Launch. Users can add products, buyers, consignee which will be saved for making the next invoice. Users can generate pdf for the invoice. Also they can download, share or mail the pdf of invoice. The features now in the app is not only limited to invoice but also includes quotation, Purchases, Expenses. I was the only developer working on this app. It has more than 25k downloads in Google Play now. I improve the app on a regular basis. (We are using razorpay payment gateway for getting payment)

Server End (Team Project): We were 3 members in the team working for server end. My work here included creating database design, writing APIs for different modules.

Google Play Link : [Easy Invoice Manager](#)

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.spectrum>
- ❑ **GitHub Link:** <https://github.com/Tech-ISM/spectrum-android-app>

Brand Store:

Brand Store is a Indore based startup. It is an app where user could get offers from shops of different categories across their city. Shop can also register them and show their offers to the users of the app. Shop needs to pay subscription money to register the shop with the app (We were using razorpay payment gateway for getting payment).

Server End: I have used Django for the server end for Brand Store Android App. The app is made completely dynamic from cities, categories, shops, offers. Whenever a shop is registered both user and admin is notified of the subscription.

- ❑ **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**

Please see my contributions after 27th of March [here](#).