

# It's Urgent

**Mentor - Akshat Tripathi**

**Contributor - Harkirat Singh**

**Org. - CCExtractor Development**

## Table of Contents

Note - To jump to a section, please click on any link below.

<b>PERSONAL INFORMATION</b>	<b>2</b>
<b>PROJECT – “It's URGENT”</b>	<b>3</b>
Simple Process Flow	3
Technologies & their Use	3
⇒ Why Firebase?	4
Pros & Cons of using Firebase (esp., Firebase Cloud Messaging - FCM)	5
Proof of Concept	6
Project Deliverables & their proposed Implementations:	6
<b>User Authentication and Management:</b>	<b>6</b>
<b>Notification System:</b>	<b>9</b>
<b>User Interface (UI) Implementation:</b>	<b>11</b>
<b>Good Documentation:</b>	<b>13</b>
<b>Testing:</b>	<b>13</b>
Timeline & Plan of action:	13
Help from the org.	14
Why am I confident about the project?	15
<b>Why Me?</b>	<b>15</b>
My Projects:	15
What motivated me the most towards applying for GSoC?	16
Qualification task	16
Commitments	16
Contributions (Issues, PRs & others) & Preparation:	17
<b>Post GSoC:</b>	<b>18</b>
Our own backend & database for the project:	18
Long term commitment, contributions & mentorship:	18
Publishing on the app stores:	18
One of the Project Ideas for future years:	19
<b>CREDITS &amp; ACKNOWLEDGEMENTS:</b>	<b>19</b>

## Personal Information

Legal Name	Harkirat Singh
GSoC Display Name	0xharkirat
Degree	Bachelor of Information Technology (Final Year)
Time zone	...
Till 7 April 2024	...
	...
After 7 April 2024	...
	...
Email	...
GitHub	<a href="https://github.com/0xharkirat">https://github.com/0xharkirat</a>
Slack Handle	@0xharkirat
LinkedIn	<a href="http://www.linkedin.com/in/0xharkirat">www.linkedin.com/in/0xharkirat</a>
Mobile Number	...
Postal Address	...

\*UTC – Coordinated Universal Time; #IST – Indian Standard Time.

## Project – “It's Urgent”

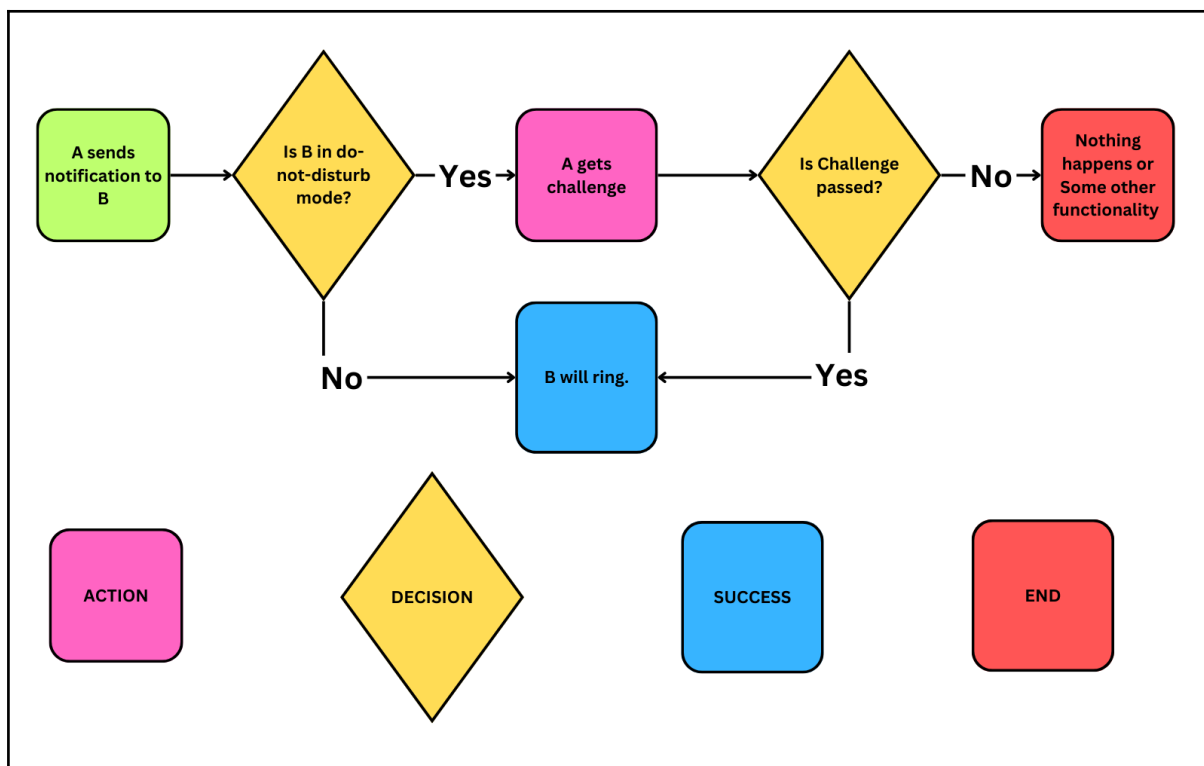
It's Urgent Project is a notification app, with the twist that allows the user trying to reach out decide how urgent it is (i.e. if they want to disrupt or not). It is not a messaging app or a phone app.

This proposal aims to build this project from scratch in this year's GSoC Program. Both Frontend & backend will be created.

**At least**, this project will have:

- A Functionality to see the device's contacts.
- Ability to check if the device's contacts are also on the app (like in WhatsApp it shows, “Yours contacts on WhatsApp”).
- Ability to send notifications to the contacts which are already on the app.
- User Account creation, linking with Phone number (act as unique Id), sign in & sign out, and request for deletion of User's data.
- (See Project deliverables section for detailed approaches.)

### Simple Process Flow



### Technologies & their Use

- ◆ **Frontend** – Flutter for UI & on-device functions.
  - State-Management – Riverpod.
  - Flutter/Dart packages to be used:

sqlite/shared_preferences	For storing data locally on device.
flutter_contacts	For getting all the data about the device's contacts.
permission_handler/ app_settings	For managing all device permissions required such as notification permission, dnd access, device contacts, local storage permissions etc.
http	For managing all the requests to & from Firebase cloud functions.
flutter_local_notifications	For managing Firebase notifications when the app is in Foreground (Open).
connectivity_plus	For managing & discovering the network connectivity.
flutter_sms	For sending SMS and MMS on Android and iOS to invite others. (Sends SMS through sim card)
Firebase related packages	For all the Firebase related functionality.
Other UI Packages	For making the app's UI beautiful, simple & minimalistic.

Note – These are not the final packages. While developing, if I find more useful & good alternatives, I will use those packages instead.

#### ◆ **Backend** – Firebase as a Backend Service & Python on Firebase Cloud Functions.

##### ⇒ Why Firebase?

Simple answer – It is very simple, easy to set up, has out-of-the box features, integrates seamlessly with Flutter. As the project grows in the future years, we can work on custom backend as per the requirements arises. But for the sake of starting the project from scratch this year, it seems the best option.

Firebase Features to be used:

Firebase Authentication	For authenticating the users.
-------------------------	-------------------------------

Firestore	For storing & managing all the data in real time.  Note: I had some bad experience in working with "Firestore" in the past. It does not work in some regions. So, Cloud Firestore is sufficient for our requirement.
Firebase Cloud Messaging (FCM)	For Sending notification to the users, both in-app & push notifications. It uses device token which is generated through & unique to every device to send the notification.
Firebase Cloud Functions	I can use Python on Cloud Functions to implement other backend requirements such as: <ul style="list-style-type: none"> <li>• Checking for DND status.</li> <li>• Sending &amp; validating the challenge task.</li> <li>• Sending notifications after the dnd &amp; challenge check.</li> <li>• Encrypting User data sent to Firestore like User's phone number.</li> <li>• It can also act as another security layer along with the inbuilt security rules between Cloud Firestore and the client app.</li> </ul>
Firebase App Check (Optional)	Once an app is in production mode, published on both the app stores, we can add Firebase App check as an extra layer of security for all the requests sent to Firebase.

## Pros & Cons of using Firebase (esp., Firebase Cloud Messaging - FCM)

The biggest advantage of using Firebase Cloud Messaging is how easy it is to integrate with Flutter Applications. Additionally, it comes with no-cost in both Firebase's plans. FCM also integrates seamlessly with Cloud Functions or any backend server (hosted anywhere) using Firebase Admin SDK for Python & other supported languages.

I researched other alternatives of Firebase Cloud Messaging, but all are somehow using the FCM device tokens needed to send notifications. See these posts:

- [https://www.reddit.com/r/aws/comments/14g42p2/please\\_help\\_me\\_understand\\_the\\_use\\_cases\\_for/](https://www.reddit.com/r/aws/comments/14g42p2/please_help_me_understand_the_use_cases_for/)
- [https://www.reddit.com/r/Firebase/comments/pp29sv/why\\_is\\_there\\_no\\_firebase\\_fcm\\_alternative/](https://www.reddit.com/r/Firebase/comments/pp29sv/why_is_there_no_firebase_fcm_alternative/)
- <https://onesignal.com/blog/firebase-vs-onesignal/#:~:text=OneSignal%20itself%20uses%20the%20FCM%20API%20internally%20to%20send%20messages%20to%20Android%20devices.>

Other alternatives to Firebase Cloud Messaging

- One Signal
- Pusher
- Twilio
- AWS SNS
- Pushy

So, it is a much simpler & easier choice to use Firebase's own messaging rather than trying hard to integrate the other alternatives at this starting point.

## Proof of Concept

GitHub: [https://github.com/0xharkirat/its\\_urgent\\_poc\\_public](https://github.com/0xharkirat/its_urgent_poc_public)

Demo Video (Watch with 1.5x speed & captions):

[https://www.youtube.com/watch?v=\\_nhs4On2zI](https://www.youtube.com/watch?v=_nhs4On2zI)

I have implemented a simple proof of concept for the Project. It is working fine on both iOS & Android Devices with the above-mentioned functionalities & developed with above mentioned technologies (except Firebase Cloud Functions).

The GitHub repository for the Proof of Concept contains a well written documentation & demo. Anyone can either download the .apk file for Android to test or watch the YouTube Demo video.

Or you can set up the project locally using the detailed instructions mentioned in the documentation & then run using Flutter.

## Project Deliverables & their proposed Implementations:

I think there are five main deliverables for this project. Everything else is related to these five main deliverables:

1. User Authentication and management.
2. Notification System.
3. Good & Simple User Interface (UI) Implementation.
4. Good documentation.
5. Testing (Unit, Widget & Integration)

### User Authentication and Management:

#### **Why do we need to implement the User authentication for our project?**

- In order to reach out to someone, they need to have the app too, so we need to implement user authentication to keep track of the users who have installed the app, currently signed-in or signed-out (i.e. users who are on the app).
- Authentication provides us with some kind of unique identifier (like Phone number or something we can set up on our own) to connect with others.
- Also, we need to authenticate all the requests to our backend (Firebase cloud functions) & our database (Firebase cloud Firestore)

#### **What needs to be implemented?**

- Implement account creation, sign-in, adding other users who are currently on the app, sign-out, and account deletion functionalities.
- Keep track of the Users' added friends even when users sign out.
- SMS Invite Functionality: Integrate SMS invite functionality to allow users to invite others by sending SMS messages.
- Show the app's frontend accordingly.

#### How - (Proposed Implementations):

1. **Authentication using Phone numbers** – (also mentioned on the project's webpage).
  - a) This approach is similar to WhatsApp's account management. Every user needs to have a phone number & they can verify the ownership of that phone number by entering the code sent to them through SMS.
  - b) User accounts are linked to phone numbers. All the data is linked to their phone number.
  - c) Other users can see in the app which of their contacts are on the app.
  - d) It is a very straightforward approach. Firebase has an inbuilt phone authentication method.
  - e) All the User data is stored in the Firebase Cloud Firestore once the verification is done with Phone number as a unique field.
  - f) We can use some kind of hashing algorithm on cloud functions to hash the phone number & then store it in the Firestore database as an extra layer of security.

#### ⇒ **Problems in this approach:**

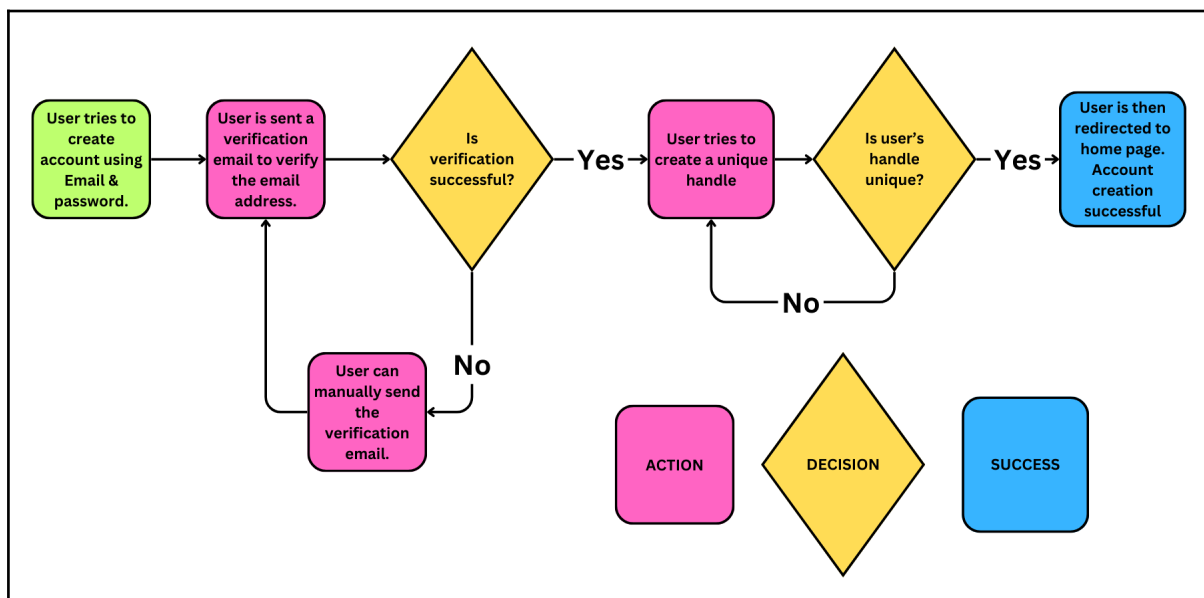
- o The main problem in this approach is that it is necessary to verify the phone number ownership, otherwise anyone can create an account using anyone else's phone number.
  - o When the real owner of the phone number will create an account & someone has already made an account using the number, it is going to clash in the database & (all sort of bad stuff starts happening – phone number is a unique field).
  - o If we use Firebase's phone number authentication – it is not very cheap for us at this starting stage.
    - Only 10 SMS sent/day are available in the free plan.
    - But we need to use the "Blaze plan" (see [Help from the org.](https://cloud.google.com/identity-platform/pricing#for_phone_authentication_and_multi-factor_authentication)) for using the cloud functions eventually. So, in the Blaze plan after 10 Free SMS sent/daily these are the current rates:
      - [https://cloud.google.com/identity-platform/pricing#for\\_phone\\_authentication\\_and\\_multi-factor\\_authentication](https://cloud.google.com/identity-platform/pricing#for_phone_authentication_and_multi-factor_authentication)
  - o Every Phone number verification provider charges money to verify the phone number.
2. **Authentication using Email & Password** – (instead of phone number, a **unique handle** like in Instagram is used as a unique field).



- Why a unique handle? Unlike phone numbers unique handles are not preowned by any people, anyone can make their own handle (if it's not in use).
- We can still verify the account creation by sending verification email to the email address on which the user is trying to create an account. It is also an inbuilt feature in Firebase Authentication and is free unlike phone verification.
- After users verify their email, they can then create a handle which is unique to their account. (We can add functionality to check whether this handle exists or not by querying our database).
- All the user's data is linked to this handle.
- Other users can use this handle to add anyone into their app. All the users' friends' handles are stored on the local device's storage just like the phone numbers are stored on the device.
- Users can use this handle to send notifications to other users who are on the app. (User's email is never shared to others, only unique handle is shared).
- When users try to sign out, then only their added friends' handles are stored in the database so that next time they login, they don't need to add again all their friends in the app.
- We can also update the database about whether a user is signed out or currently signed in. So, when someone is trying to send the notification, they can see that the user is currently signed out.

*"Talk is cheap – show us the workflow diagram".*

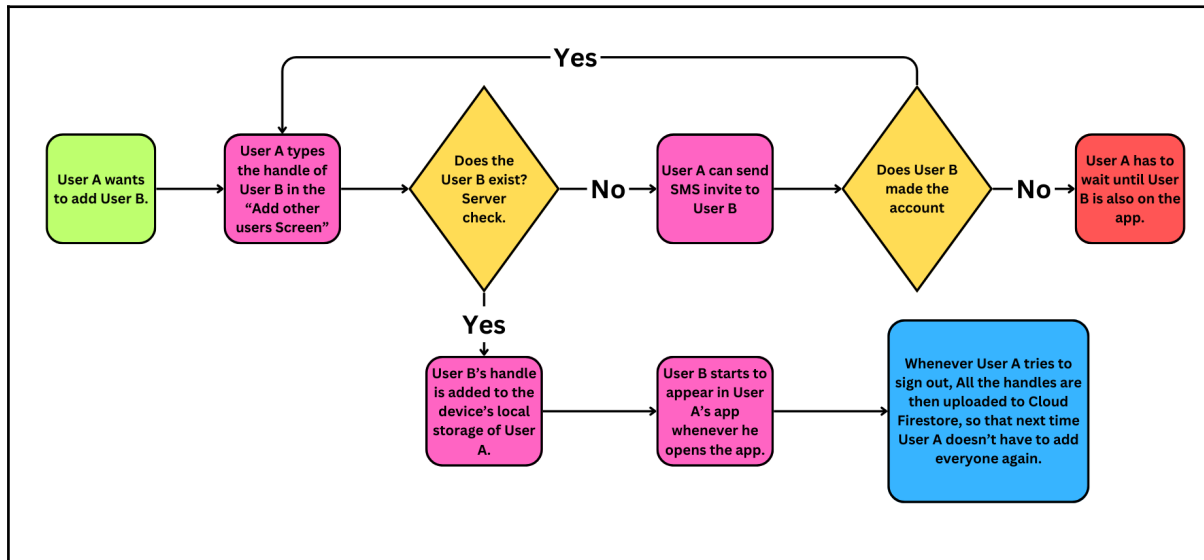
**Workflow diagram for the first-time account creation.**



**First-time account creation workflow.**

⇒ If you are still confused about how a user will add the other users without the phone number in the second approach, See this workflow diagram below.

**Workflow to add users (if they exist or not) in the app.**



**Add users (if they exist or not) in the app workflow.**

### My Opinion:

I think both the approaches are fine. I can implement both the approaches.

- For the first approach 10 Free Phone verifications /day is the limit, then it charges according to the region.
- For the second, no charges are required.
- I mean if we go with the first approach, we have to implement the verification eventually.
- Every other functionality remains the same for both the approaches.
- I am open to more suggestions.

### Notification System:

#### Why?

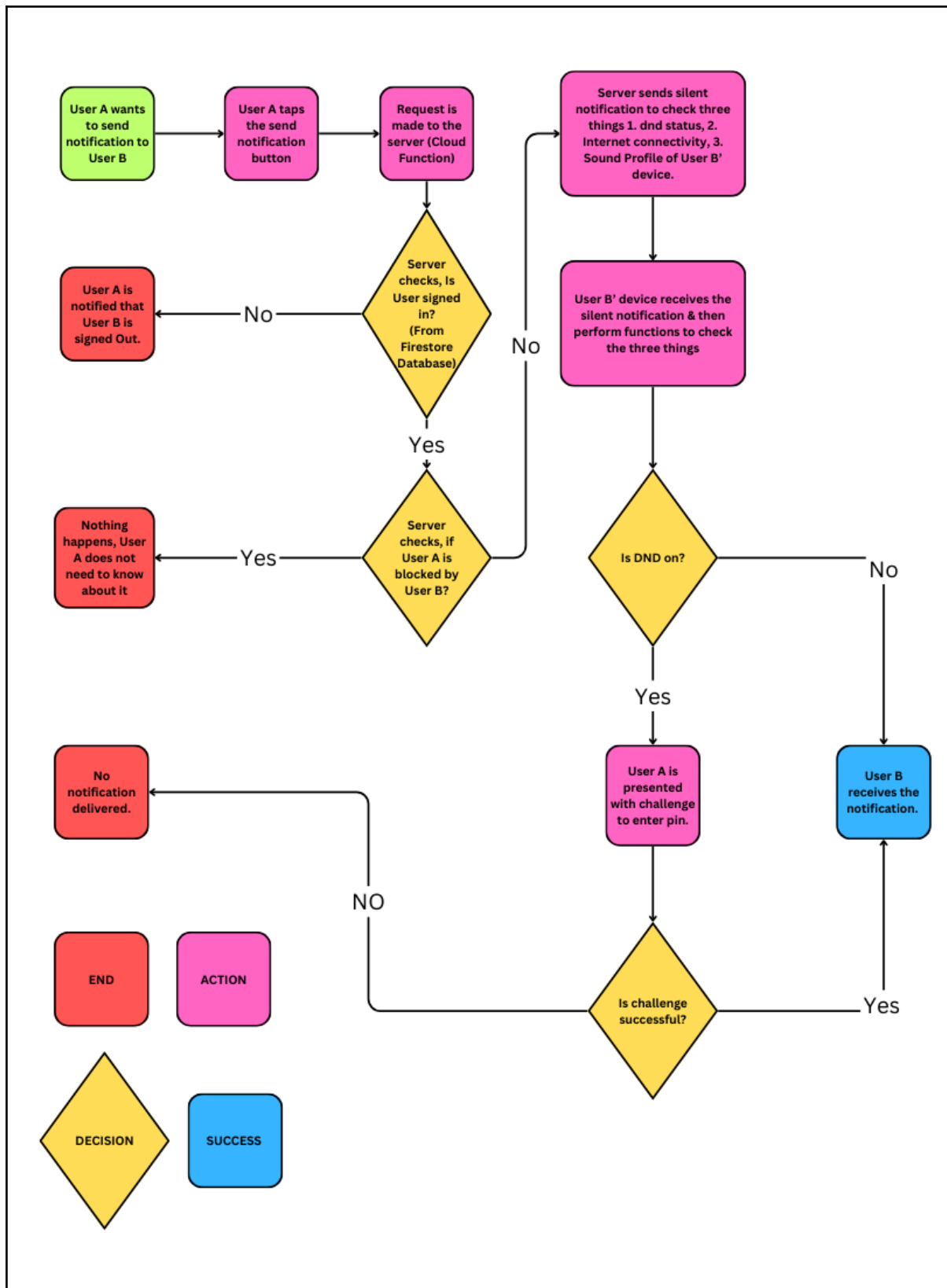
This is the core functionality of the app. Through a proper notification system, users can send the notifications to the other users added in their app accounts.

#### What needs to be implemented?

- Notification System:
  - Design and implement a notification system for sending alerts to other users.
- Do-Not-Disturb Mode & other checks:
  - Implement logic for handling notifications based on the recipient's dnd status.
  - Also add other checks such as:
    - Is the recipient signed out?
    - Is the sender blocked by the recipient?
    - Is the recipient's device connected to the internet?
    - Is the recipient's device on silent or vibration mode for sound?
- Challenge Mechanism:
  - Create a challenge mechanism for users in do-not-disturb mode to verify urgency.

- Implement a 4-digit pin challenge for user verification.
- Necessary Permissions:
  - Initial app setup to grant permissions for receiving notifications even when in dnd mode.
  - All other required permissions for internet connectivity check, local storage check & etc.

**How? See the Workflow diagram below.**



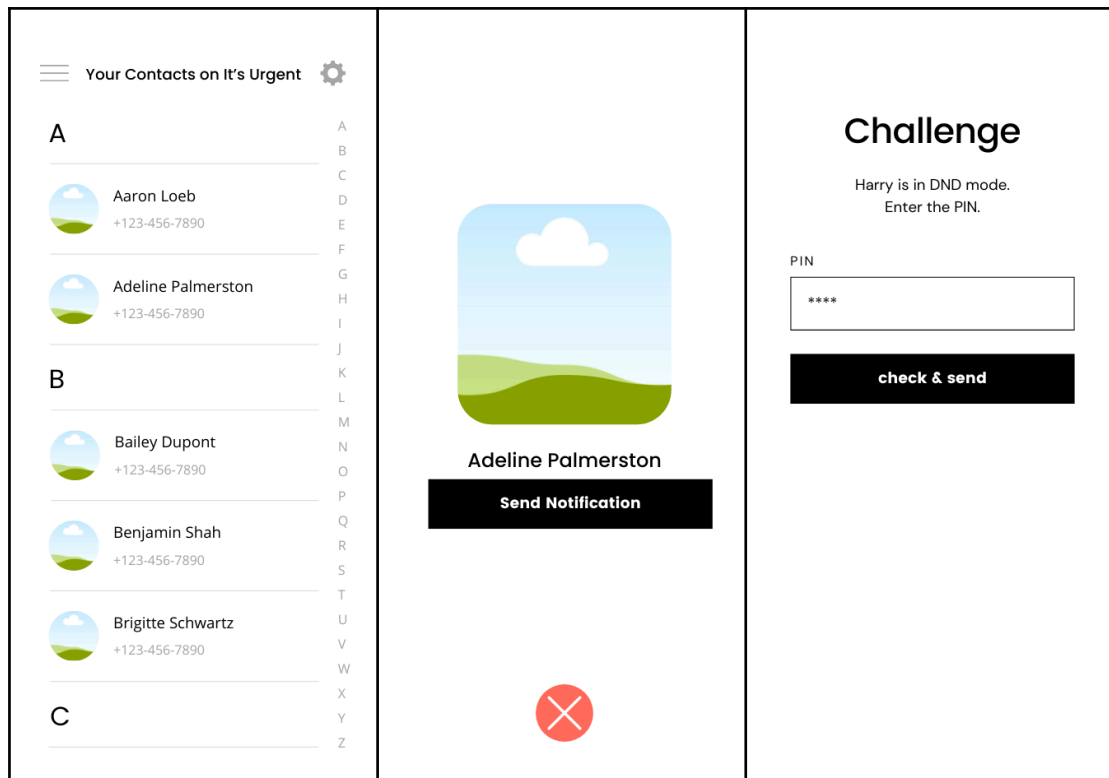
**Notification System Workflow.**

### User Interface (UI) Implementation:

- Currently I have planned 7-8 screens for the app's use case as follows:

- Login/Account Creation Screen.
  - Email Verification Pending/Successful Screen.
  - Create your unique handle Screen.
  - Home Screen (After login done) - all the other users' list shown here.
  - Add/invite others Screen.
  - Individual User Screen (with send notification button).
  - Challenge screen (when receiver is in DND mode).
  - Notification screen (any notification received).
- General UI development tasks:
    - Design and develop intuitive user interfaces for all app screens.
    - Ensure consistency in design elements and adhere to material design guidelines.
    - Implement smooth navigation between different sections of the app.
    - Prepare the app UI theme, Mock-ups, app logo & designs in the Community bonding period.
  - Some UI **wireframes** (not complete designs - see [timeline](#) section):

<h3 style="text-align: center;">Create new Account</h3> <p style="text-align: center;">Already Registered? Log in here.</p> <p>NAME</p> <input style="width: 80%;" type="text" value="Harkirat Singh"/> <p>EMAIL</p> <input style="width: 80%;" type="text" value="harkirat@email.com"/> <p>PASSWORD</p> <input style="width: 80%;" type="password" value="*****"/> <p>CONFIRM PASSWORD</p> <input style="width: 80%;" type="password" value="*****"/> <div style="text-align: center; margin-top: 10px;"> <div style="background-color: black; color: white; padding: 5px 20px; display: inline-block;">sign up</div> </div>	<h3 style="text-align: center;">Login</h3> <p style="text-align: center;">New? Create account here.</p> <p>EMAIL</p> <input style="width: 80%;" type="text" value="harkirat@email.com"/> <p>PASSWORD</p> <input style="width: 80%;" type="password" value="*****"/> <div style="text-align: center; margin-top: 10px;"> <div style="background-color: black; color: white; padding: 5px 20px; display: inline-block;">log in</div> </div>	<h3 style="text-align: center;">Forgot Password</h3> <p style="text-align: center;">Enter your email.</p> <p>EMAIL</p> <input style="width: 80%;" type="text" value="harkirat@email.com"/> <div style="text-align: center; margin-top: 10px;"> <div style="background-color: black; color: white; padding: 5px 20px; display: inline-block;">send</div> </div>
---	---	--



### Good Documentation:

- Write good technical articles about explaining my work.
- Keep the Readme updated about the project details such as features under development, setup guide & contribution guide etc.

### Testing:

- Write the Unit, Widget & Integration tests for all sets of scenarios.

### Timeline & Plan of action:

Phase 0 (May 1 – 26)	Community Bonding Period	<ul style="list-style-type: none"> <li>● Introductions &amp; bonding with mentors &amp; other fellow contributors.</li> <li>● Design the UI, mock-ups, choose app's theme, colours, fonts, app logo (all design related parts).</li> <li>● Setup the project Repo &amp; initial documentation.</li> <li>● Create a Firebase account &amp; upgrade to the Blaze plan (see <a href="#">Help from the org.</a>).</li> </ul>
Phase 1 (May 27 – Jun 16)	Week 1, 2 & 3	Implement the User Authentication and Management functionality.

Phase 2 (Jun 17 – Jul 7)	Week 4, 5 & 6	Implement the Notification system with Do-Not-Disturb Mode & other checks along with Challenge mechanism.
(Jul 8 – 14)	Midterm evaluation Week – Deadline (July 12)	<ul style="list-style-type: none"> <li>● <b>Evaluation:</b> Evaluate based on the app's main functionality with complete user authentication &amp; notification system.</li> <li>● Little relaxed week (break from coding).</li> </ul>
Phase 3 (July 15 – Aug 4)	Week 7, 8 & 9	<ul style="list-style-type: none"> <li>● Continue implementing Notification systems.</li> <li>● Improve UI, add animations &amp; logo.</li> <li>● Start testing (Unit) &amp; documentation.</li> </ul>
Phase 4 (Aug 5 – 25)	Week 10, 11 & 12	<ul style="list-style-type: none"> <li>● More testing (Widget &amp; integration).</li> <li>● More documentation.</li> <li>● Refactoring - Clean Code.</li> </ul>
(Aug 26 – Sept 2)	Final Evaluation Week	Final Evaluation based on the app's functionality with simple, beautiful UI enriched with smooth navigation & animations.

## Help from the org.

As mentioned on the organisation's website, I need to let you know, *"what kind of support I will need from the organisation"*,

- Firebase account (Blaze plan required for using Cloud Functions):
  - (Normal approach) I can setup the Firebase project using my own google account, even upgrade it to the "[Blaze plan](#)" for using the Firebase Cloud Functions which requires a credit card info.
  - (It is optional) Or an organisation can simply make a Firebase account & provide me access to it. Then I will set up everything using that Firebase account.
- Package name suggestion: For Firebase Cloud messaging to work on iOS, instead of using the default **com.example.its\_urgent** package name, we need to specify the package name using our own unique identifier like:
 

```
<string>$(EXECUTABLE_NAME)/</string>
<key>CFBundleIdentifier</key>
<string>com.hsiharki.itsUrgentPoc</string>
```

  - Either we can use, org's identifier if you have any like "**com.ccextractor.its\_urgent**",
  - Or I can use mine, like "**com.hsiharki.its\_urgent**" for iOS & "**com.0xharkirat.its\_urgent**" for Android.
- General Support, collaboration & feedback: Well, it comes with the program.

## Why am I confident about the project?

This project is a unique app idea. It is also quite useful for the common users. I want to choose this project because it is the only project idea listed in the Flutter section which is not implemented yet (now also the new Firebase editor project). With every other Flutter project getting its contributor, I think it is now the time to implement this project also. It will give me a great learning experience about developing an open-source project from scratch & then growing it through upcoming years as a contributor & mentor myself.

## Why Me?

Based on the skills required for the project, I am very confident that I will be able to complete the project with all its deliverables. I have a fair amount of experience working with Flutter, Firebase & Python. I am constantly learning & upgrading myself in all these technologies & also in other upcoming useful technological trends. My research skills about finding better solutions & workarounds if any limitations arise are very good. I also have experience publishing the production ready apps on both Google Play Store & Apple App Store.

You can see some of my projects made using Flutter, Firebase & Python here:

## My Projects:

- Live Darbar (Flutter) – A simple yet powerful & intelligent mobile app for live audio & video broadcast from Golden Temple, Amritsar. Both Android & iOS apps are fully published on their respective app stores.
  - GitHub Repo (with detailed documentation) - [https://github.com/Oxharkirat/live\\_darbar](https://github.com/Oxharkirat/live_darbar)
  - Google Play Store - [https://play.google.com/store/apps/details?id=com.hsiharki.live\\_darbar](https://play.google.com/store/apps/details?id=com.hsiharki.live_darbar)
  - Apple App Store - <https://apps.apple.com/us/app/live-darbar/id6449766130>
- anvaad-py (Python) - Transliteration toolkit for Gurmukhi (Punjabi) ASCII to Unicode, romanized and other character sets.
  - GitHub Repo (with detailed documentation) - <https://github.com/Oxharkirat/anvaad-py>
  - Published on Pypi - <https://pypi.org/project/anvaad-py/>
- A typewriter Flet app (Flutter, Python & Flet Framework) – A simple MacOS & Windows app to simulate typing on any input field, independent of the app, made using PyAutoGUI module with UI in Flutter, Python & Flet Framework. Useful in scenarios where in input fields copy paste is prohibited.
  - GitHub repo (With installation instructions) - <https://github.com/Oxharkirat/typewriter>



- Other collections of apps, websites, packages & tools that I've created to learn, be helpful, fun & sometimes just to show off my programming skills.
  - Can be found on my portfolio website (which is also made using Flutter for web) - <https://0xharkirat.com/> - (Currently Work in progress for mobile responsive design)
  - Also, on my GitHub profile - <https://github.com/0xharkirat>

## What motivated me the most towards applying for GSoC?

Until now, I've always developed projects alone. It is so boring & I am tired of coding alone. I want to be part of a good open-source organisation. I have a lot of other useful & cool project ideas, which I can't build alone, I want to be with likeminded people which is a serious lack that I currently have because of my location. I am in a remote location in Australia, with very few programmers or developers around me. This is my chance to connect with other programmers. I have looked into other orgs which use Flutter as technology. I liked this organisation more because it has the option to bring & explore new ideas from scratch rather than only working on the projects already started. Also mentioned on the org website, "this org is a small org, which means that my contribution will have a large impact. It's not going to mean a 0.5% improvement on a big project --- it's going to be more than 10% on a medium size one. If I like challenges and want a chance to shine this is my place". I want to become a contributor & then mentor in upcoming years.

## Qualification task

I also managed to complete the qualification task listed for the Flutter. A Flutter app which detects the objects using Tensor Flow's tflite\_flutter plugin & then zooms in when tapped on any object detected. Also saves the zoomed in image to the photos or DCIM folder on IOS & Android respectively.

- ⇒ GitHub repo (with detailed documentation) - <https://github.com/0xharkirat/autozoom-camera-flutter>
- ⇒ Available to test using Android .apk & iOS Test Flight: [https://github.com/0xharkirat/autozoom-camera-flutter/releases/tag/v1\\_beta](https://github.com/0xharkirat/autozoom-camera-flutter/releases/tag/v1_beta)
- ⇒ Android Demo Video - [https://www.youtube.com/watch?v=m4n8GqSb\\_yQ](https://www.youtube.com/watch?v=m4n8GqSb_yQ)
- ⇒ iOS Demo Video - <https://www.youtube.com/watch?v=AkChtczrh6g>

## Commitments

- ⇒ How many hours will you work per week on your GSoC project?
  - I plan to spend 25-30 hours per week on the project. As it requires both Frontend & Backend.
- ⇒ Other Commitments, if any?
  - No. I am a regular college student and attend 4 classes per week.

- ⇒ Do you plan to apply for any other organization for GSoC'24?
  - No, I am only applying to CCExtractor Development. (See the reason in answer to the "What motivated me the most towards applying for GSoC? Section")
- ⇒ If you get selected as a GSoC contributor, would you like to work on other tasks besides the project of your choice?
  - Well, it is a tricky one - for general purposes - I intend to focus on contributing to my project, but if some other contributor needs help or support, I can help with the best of the knowledge I have.
- ⇒ If you are not selected as a GSoC contributor, would you like to work on the projects as a general contributor?
  - Yes, & I think this is the whole motive behind the GSoC program to contribute meaningfully to the open-source community. Of course, if you get selected, you have a strong motivation to contribute but also in general developing the habit to contribute meaningfully to open source should be the main goal.
- ⇒ Would you like to contribute to CCExtractor Development for long-term even after the GSoC program ends?
  - Oh yes. That is my long-term goal. Taking this project to grow & also to work on other useful, fun & cool project ideas as a contributor & mentor myself is the motive behind my participation.

## Contributions (Issues, PRs & others) & Preparation:

As this project needs to be implemented from scratch, I spent the majority of my time researching the tools, technologies, edge cases, scopes & limitations to make this project successful.

I also managed to contribute to other Flutter projects, CCExtractor website, Flet-website & many more other projects outside our organisation. I am very new to open-source contributions, although I have been using GitHub since the beginning of my coding journey, I started contributing just recently. Moreover, other Flutter projects already have some good contributors, so instead of getting attention by competing with them on the same issues which they were solving, I tried my best to contribute meaningfully & respectfully.

I also like contributing in some other way, like solving other's queries regarding any projects on slack, replying to all the dms sent to me regarding any project, pointing them towards right resources & right mentors & even sharing the resources regarding any issue which arises. Even if they are planning to do the same project as me, I never see them as a

competitor, rather as a collaborator. That I feel is the true essence of GSoC & open source in general.

PRs merged & issues resolved.

- ⇒ [https://github.com/CCExtractor/ultimate\\_alarm\\_clock/pull/499](https://github.com/CCExtractor/ultimate_alarm_clock/pull/499)
- ⇒ [https://github.com/CCExtractor/ultimate\\_alarm\\_clock/issues/498](https://github.com/CCExtractor/ultimate_alarm_clock/issues/498)
- ⇒ [https://github.com/CCExtractor/ultimate\\_alarm\\_clock/issues/497](https://github.com/CCExtractor/ultimate_alarm_clock/issues/497)
- ⇒ <https://github.com/CCExtractor/taskwarrior-flutter/pull/331>
- ⇒ <https://github.com/CCExtractor/taskwarrior-flutter/issues/317>
- ⇒ <https://github.com/CCExtractor/website/pull/55>
- ⇒ <https://github.com/CCExtractor/website/pull/49>
- ⇒ <https://github.com/flet-dev/website/pull/244>

## Post GSoC:

It's very improbable that there will be any unfinished parts of the project within the 12-week GSoC timeframe. However, if by chance there are, I'm completely dedicated to wrapping up any remaining tasks during the extended timeline. During my time in the GSoC program, I'll work hard to contribute to CCExtractor Development and participate actively in community discussions to the best of my capability.

## Our own backend & database for the project:

In the upcoming years, we can write our own backend from scratch with our own chosen database for more functionality & more control. Or we can explore other Firebase alternatives such as Supabase & Appwrite, etc.

## Long term commitment, contributions & mentorship:

Moreover, I aspire to establish a long-term relationship with this project, other upcoming projects, and the organization, and I see myself becoming a mentor in the future, guiding others in their journey of contributing to open-source projects after gaining experience as a contributor this year.

## Publishing on the app stores:

I also have experience publishing the production ready apps on both Google Play Store & Apple App Store. So, when we feel the app is ready for production, I can help in publishing the app on both the app stores available.

One of the Project Ideas for future years:

...

## Credits & Acknowledgements:

At last, I am very thankful to these people, who are supporting, guiding & encouraging me in every step of making this proposal & eventually this project a success.

- **Carlos Fernandez:**
  - Thank you for making this platform & organisation which gives opportunities to contributors like me for learning, collaborating, sharing & developing useful projects.
  - Thank you for answering all my messages, resolving all my queries & guiding me at every step.
- **Akshat Tripathi:**
  - Thank you for helping, guiding, supporting me & also for replying to all my long questions.
  - Thank you for your upcoming support as a mentor if I get chosen.
- **Mabud Alam:**
  - Thank you for guiding me towards implementation of invoking the functions on notifications received.
- **All other fellow contributors:**
  - Thank you for taking part in discussions, sharing resources, ideas, proposal templates & encouraging each other & especially me to learn new technologies & think differently.

# THANK YOU.