Google Summer of Code

AOSSIE

Google Summer of Code Project Proposal for AOSSIE

Project Details

Project: Resonate

Idea: Resonate Version 3

Project Size: Large (350 Hours)

Project Length: 18 Weeks

Personal Information

Contact Information:

Name: Madhav Gupta

Discord Username: M4dhav

GitHub Profile URL: https://github.com/M4dhav

Education

University: Bennett University

Degree: B.Tech CSE

Expected Graduation Date: August 2026

Other Commitments

I have Semester Exams in May 2025, however Bennett University is supportive of students pursuing programs like GSoC, and as such, they will allow me to be evaluated through alternative means if my proposal is accepted. I am also interning at a startup, however the work expected of me there is very less (~10 hours/week). Also, the startup also supports Open Source culture, and they will allow me to step back on temporary leaves to focus on my contributions to GSoC, if required.

Project Vision and Motivation

Vision:

Social Voice platforms are like a bastion against the suppression of Free Speech under the guise of moderation that we now see in so many places.

An Open Source Social Voice Platform like Resonate would bring freedom and power back to the people, and as long as we people have tools like this, we will have Freedom.

An App like Resonate has the potential to be used the world over, and the more people use it and spread the word, more people will become aware of how so many big tech companies are selling User Data for profit and how much better using FOSS Alternatives can really be.

Users often have to make compromises with data safety simply because of the lack of good enough alternatives, but Resonate is different—as good as its competitors, if not better.

Motivation:

I am extremely passionate about FOSS and have been developing in Flutter using BaaS services like Appwrite for well over a year now. I have also worked *with LiveKit.*

I have been contributing to Resonate since September 2024, and as a result, understand the codebase and coding conventions of Resonate. I also have experience contributing to other Open Source Projects as well.

I believe I am the best person to help take Resonate forward, as I have experience in all of the various fields that Resonate is currently being built with, as well as a passion to see a truly free Social Media Voice platform realized and used by people.

Additionally, contributing to Resonate will also help me improve my own Mobile Development Skills, give me an opportunity to connect with great Developers and write better, maintainable and scalable code.

Past Experience in Software Development

Project Name	Project Description	Project Language/Framework	Contributions
Infinite Horizons	Infinite Horizons is an Open Source Project being built to help students increase their study and work efficiency using Scientific Research Based Methods. It contains numerous features like a Pomodoro clock and tips to improve your effectiveness.	Flutter	• Implemented DND (Do not Disturb) Support • Complete In-app link handling integration - Video Players, WebViews, PDFs and other assets • UI Improvements - Added new elements, handled overflows, made app responsive
Bitcoin DevKit Flutter SDK	Bdk-flutter is the official Flutter Library for the Bitcoin Development Kit. The bdk library aims to be the core building block for Bitcoin Applications of any kind.	Flutter, Rust	• Migrated library to AGP 8.x+ • Fixed a critical path naming issue that was causing git clone, and subsequently pub get, to fail (yet to be merged)

Project Name	Project Description	Project Language/Framework	Contributions
Mapbox Maps Flutter SDK	This SDK is the officially developed Mapbox SDK for Flutter which allows users to use Mapbox products.	Flutter	Added an Example for a newly introduced API which required specific configuration steps not mentioned elsewhere in the documentation
Alpha Go	Under development DeFi App to connect Bitcoin Enthusiasts and incentivize attending Blockchain related events. Also includes features like an NFT Marketplace.	Flutter	• Implemented all Web 3 Features - In-App Wallet, Transactions, Displaying Assets, Minting Ordinals, etc. • Implemented Authentication, Chat Functionality, 3D Maps (using Mapbox), Deep Linking, Biometric Authentication • Working on a Internal Chatbot to integrate with the app and perform DeFAI services
Unier	Under development accessibility app ideated to help specially abled Indian Consumers, with support for Regional Languages. It will help them integrate better in society by converting daily conversations into Text-based conversations. Currently uses	Flutter	Added the Speech Transcription and Synthesis services that form the core of the app using the Vosk family of Models for transcription and On-Device functionality for synthesis.

Project Name	Project Description	Project Language/Framework	Contributions
	LiveKit for calling purposes but will soon be integrated with PSTN Gateways		

Contributions to AOSSIE (Resonate)

Pull Request #	Pull Request Title	Pull Request Contributions	Status
PR #456	Complete Chapter Player UI Overhaul	The previous UI/UX of the Chapter Player Screen was suboptimal and unintuitive, which was apparent when the description of the Chapter was long enough to exceed the bounds of the screen. This PR performed a complete overhaul of the screen and improved the UI/UX with the use of Custom Scroll Views	Merged
PR #457	Added input length validation and handled overflows throughout the app	This PR introduced Input Validation and handle any Text Overflows throughout most of the app, increasing robustness while trying to not hinder User Functionality in any way.	Merged
PR #461	Add tags for genres in explore screen	The previous representation for the Genres of different Stories was not very beautiful, and only was mentioned as a part of the text in the Tile displaying the	Merged

Pull Request #	Pull Request Title	Pull Request Contributions	Status
		information. This PR converted that representation to showing Genres as a “Tag” which was color coded for different genres and improved the overall UI of the app.	
PR #449	Added Extension Method for Color Class to convert Color to Hexcode	Resonate used the Color.toString method to convert a Color to it’s hexcode for storage and future recreation of the color. However the behavior of this method instead produced a String Representation of the Color Object and not the hexcode. This PR Added an Extension Method to the Color Class that converted the color into a hexcode	Merged
PR #424	Changes to Color API usage and removal of deprecated code	This PR replaced the use of the Color.withOpacity method which was deprecated in favor of Color.withValues method from Flutter 3.27	Merged
PR #453	Made floating action button icon change dynamically	The Bottom Nav Bar used throughout Resonate has a central Floating button which performs different actions depending on what screen the user is on. Previously the icon for this button was the same which is unintuitive and confusing for the end user. This PR changed the	Merged

Pull Request #	Pull Request Title	Pull Request Contributions	Status
		icon for said button dynamically based on the visited page.	
PR #450	Correct the ID of the Covers being uploaded for the Story Chapters	This PR fixed what might have been a regression in the codebase that caused the entire Add Story flow to break	Merged
PR #422	Fix incorrect IP address in ONBOARDING.md	This PR fixed a mistake in the instructions in the Onboarding Guide which caused developers using a Physical Device to be unable to test the app.	Merged
PR #80 (Backend)	Increased length of creatorName in Stories Collection	This PR to the backend setup repository included a minor change related to #457 that increased the Max Length of creatorName in the Stories Collection.	Merged

Detailed Proposal Description

My primary aim would be to get Resonate out to the public and have people actually use this product that AOSSIE has been developing for over two years now, so we can use the resulting community feedback to further improve the application. Following that, I will add some more features to the app which I believe are necessary and focus on improving existing functionalities and fixing bugs that may be found post-deployment. This will all culminate with an objective to deploy a v2 of Resonate at the end of my Project.

Tasks:

1. *Improve the UI in some places of the app where it can be better and make it More Responsive by replacing statically assigned dimensions*

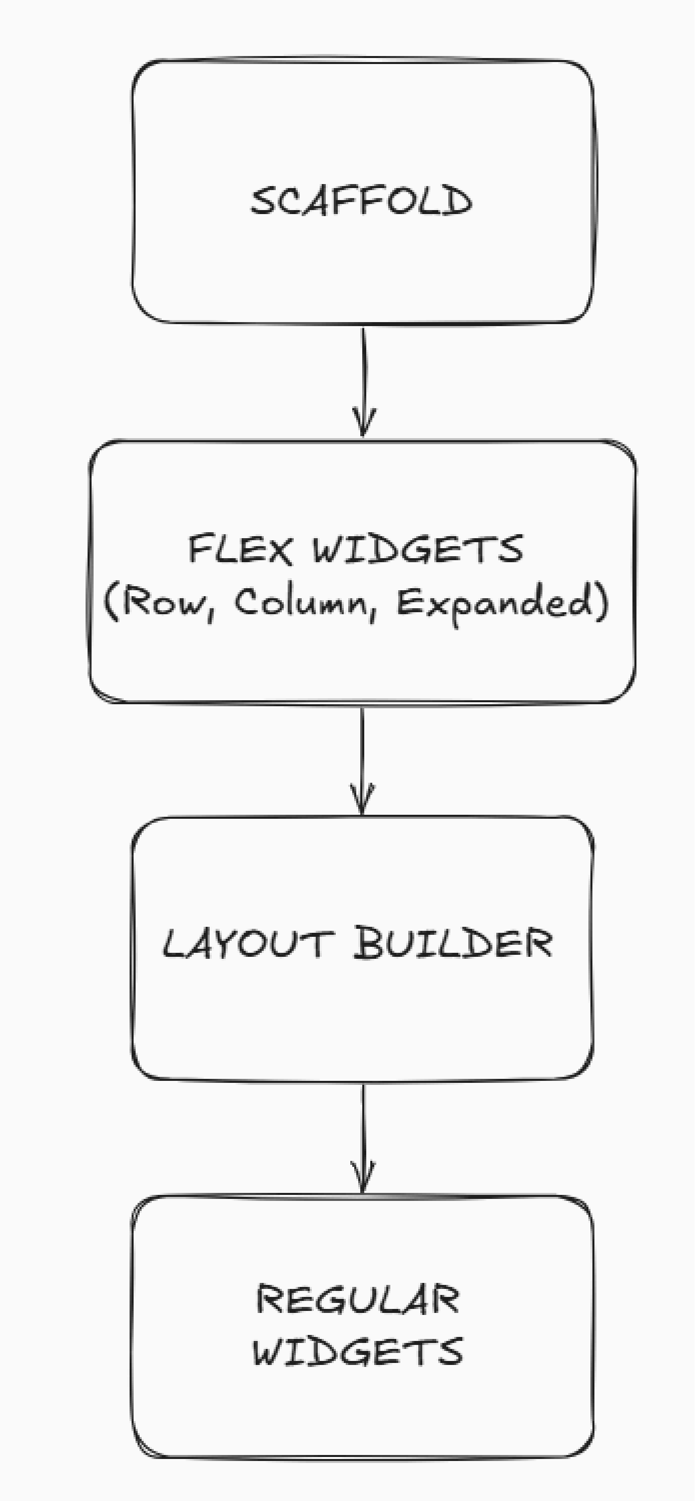
2. *Bring the current main branch to a stable state after looking over and merging recent changes to the dev branch and deploy a v1 of the Resonate App to the Google Play Store and the Apple App Store.*
3. *Write tests for key functionalities throughout the app so any subsequent PRs will be less likely to cause regressions.*
4. *Setup a CI/CD pipeline for the Main Branch to automatically compile and deploy updated builds to the respective App Stores.*
5. *Complete existing Chat Functionality implementation for Rooms and add features like notifications*
6. *Add Internationalization to the app by implementing support for various languages.*
7. *Improve Pair Chat Experience by listing available users and offering the option to choose who to chat with.*
8. *Add User Search functionality to Explore Page and implement functionality to follow creators and stay updated on their favorite ones.*
9. *Add "Friends" functionality to enable users to connect more with any added friends.*
10. *Improve Accessibility of the App by implementing TTS and STT features for specially-abled users. This functionality can either be achieved using device specific Speech to Text Capabilities or Open Source Models like Vosk .*
11. *Add Host Permissions to the discussion rooms*
12. *Add Live Chapter feature to allow users to live stream a new Story Chapter*
13. Add better search functionality using a proper search engine like Meilisearch
14. *Test all code thoroughly and ensure everything is stable before merging all changes with the Main branch to conclude the project with a v2 deployment of Resonate.*

Talking about them in detail:

1. Improve Responsiveness and look of the UI:

Throughout the app, the dimensions for various widgets have been defined statically, increasing the risk of the UI to break, especially on smaller screens. Making these sizes responsive would ensure that Resonate has pixel-perfect UI on every screen. Additionally, the app also has a lot of UISizes Class Objects which contain various Height and Width values calculated at runtime based on a MediaQuery. As UI Elements don't always necessarily use one of the sizes already defined in this Class, developers would either have to add more sizes to the class, which is cumbersome, or settle for using a predefined size which might not quite fit with the UI. This improvement can be implemented using a combination of methods:

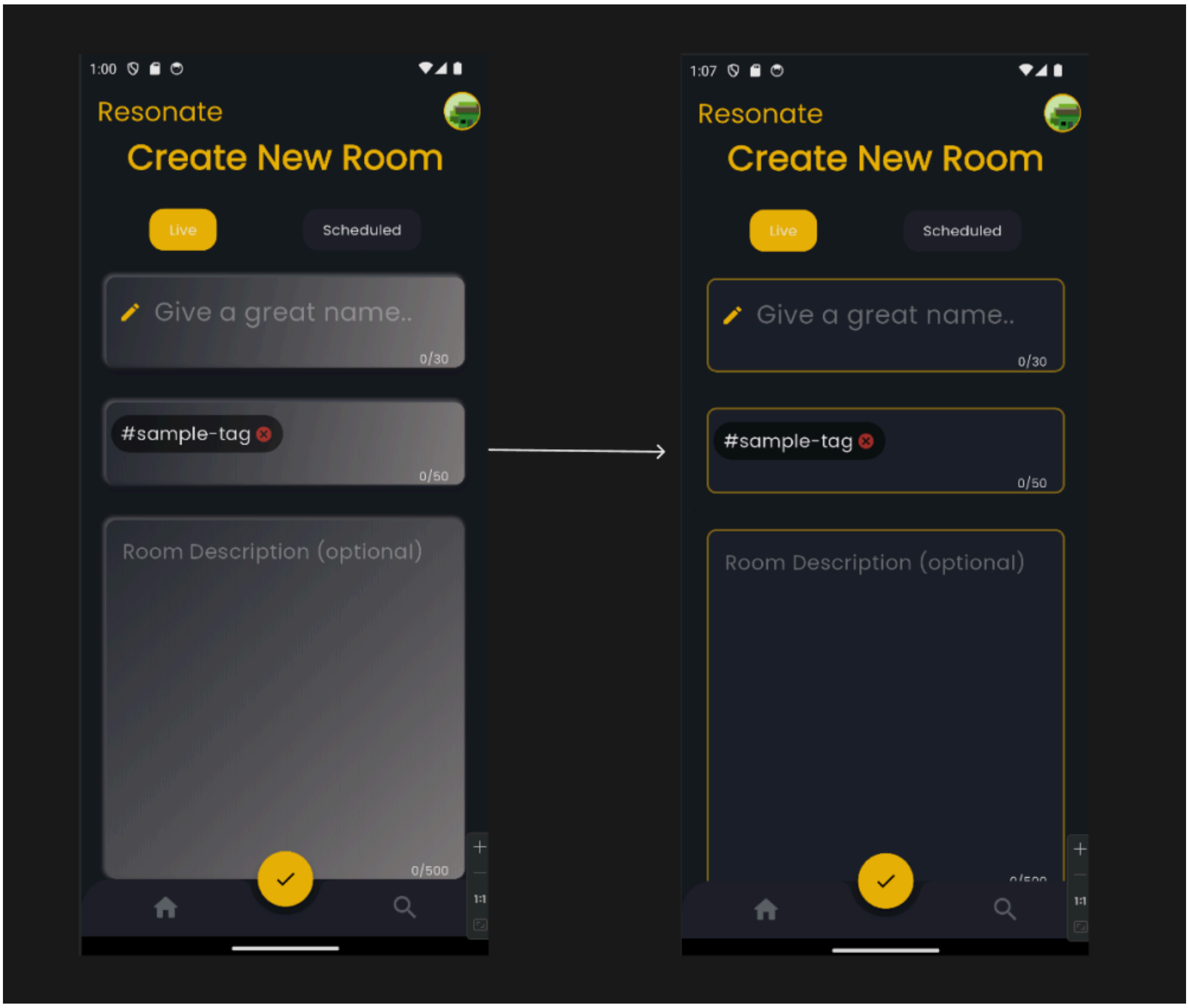
- *In most places, LayoutBuilder can be used to replace the sizes, which allows all children to reference the size of their parent to take their own sizes. The Top Level Widgets can then be placed in Flex Widgets like Row and Column to automatically fit within the screens.*

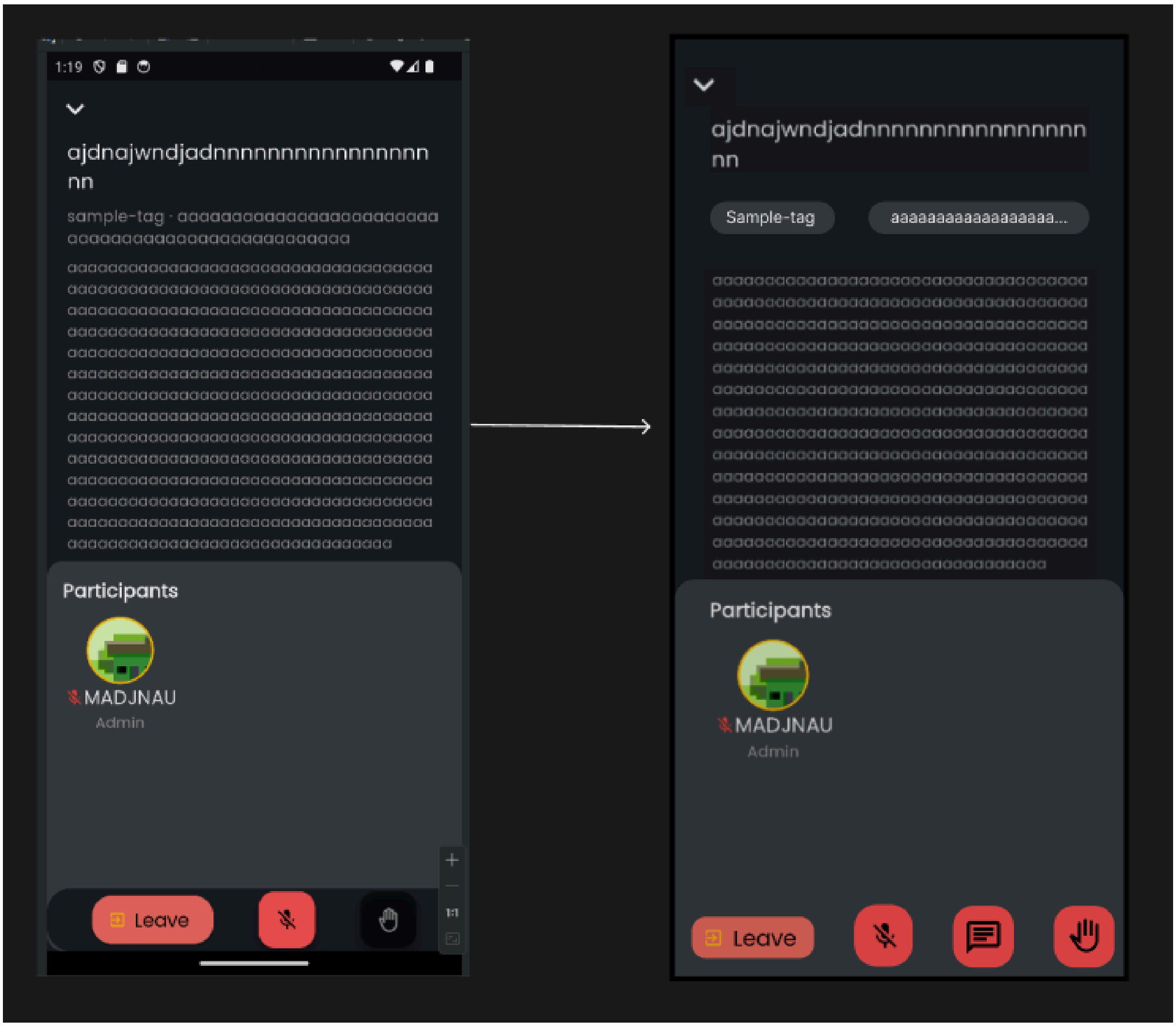


- *In places where defining a size without the use of LayoutBuilder is deemed necessary, we can use app wide packages like Responsive Sizer to assign responsive sizes to widgets*

This approach will both improve the developer experience and ensure that the app retains the look of it’s UI on every device.

Some examples of UI Improvements:




1:19
ajdnajwndjadnnnnnnnnnnnnnnnn nn
sample-tag · aaaaaaaaaaaaaaaaaaaaaaa
Participants
MADJNAU
Admin
Leave
ajdnajwndjadnnnnnnnnnnnnnnnn nn
Sample-tag
aaaaaaaaaaaaaaaaa...
Participants
MADJNAU
Admin
Leave

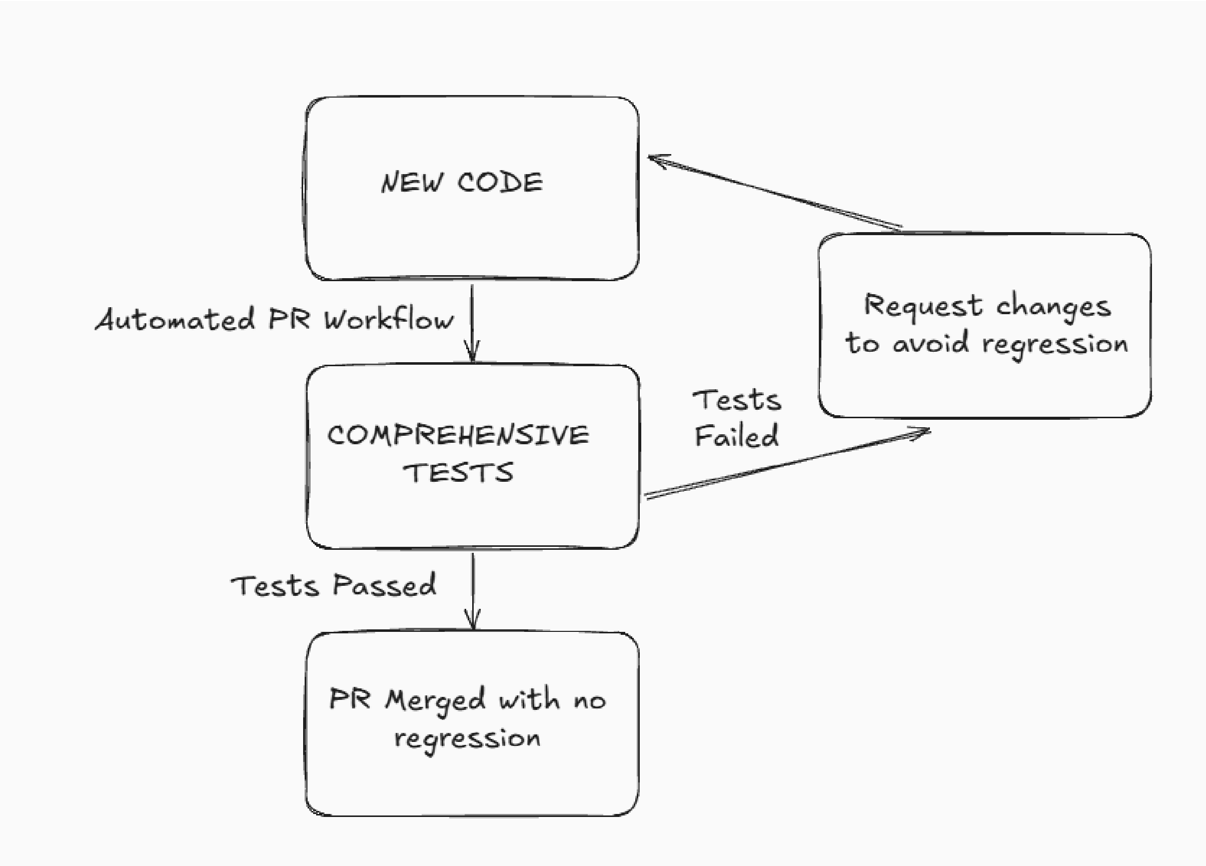
2. Merge all recent changes into main, test and deploy v1 of Resonate

The Resonate App has been in In-House Development Since March 2023, marking 2 years of development. However, it has not been released on any of the App Stores yet, and we don't even have a compiled build for download from GitHub Releases. Deployment of v1.0.0 of Resonate on the App Stores would allow the app to finally be used by users outside the developmental circle of Resonate, and provide us with valuable feedback to iterate on and improve the app, solving bugs, implementing requested features, and actually fulfilling the purpose for which Resonate was envisioned. I have experience deploying apps on both the Google Play Store and Apple App Stores, and although I do not own a MacBook, I can collaborate with anyone within the organization who does to deploy a build of the application for the first time, which later will be automatically handled by the CI/CD.

As Resonate collects and stores User Data, A privacy policy and Account Deletion Form will also need to be created and hosted on AOSSIE's website as required by Google Play Store policies. This sub-task is also included in this Task.

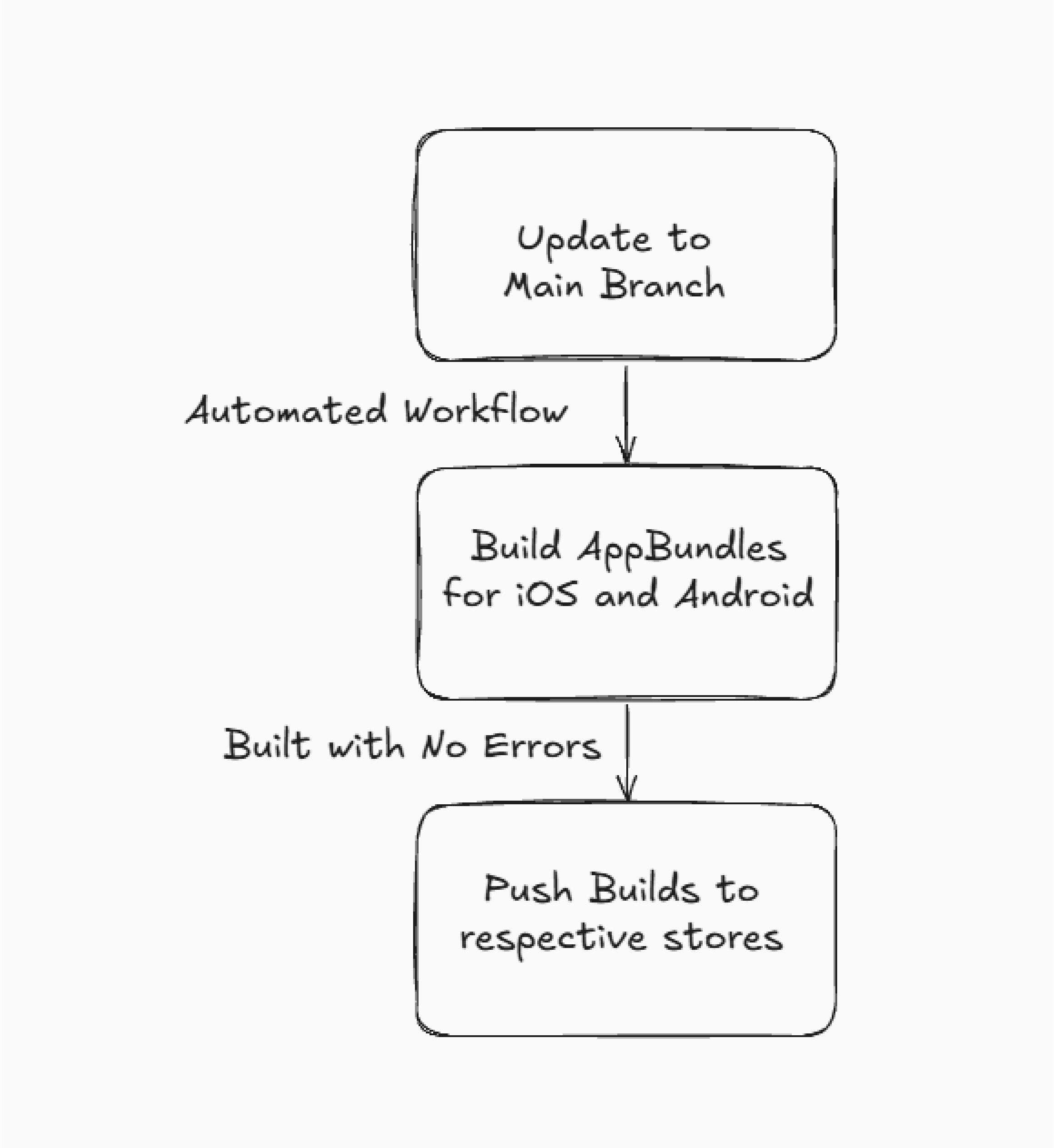
3. Write tests for key functionalities

As any app get's larger, it becomes harder to test it manually, and thus all major projects use automated tests to ensure that no regressions occur in the codebase, thereby hindering functionality. As Resonate will be entering production and be deployed on the App Stores, we should strive to avoid any chance of regressions as well, and the proper way of doing that would be to write tests using Flutter's baked-in testing functionality*. The target of this will be 90+% code coverage using a combination of Unit, Widget, and Integration Tests. Additionally, a CI/CD Workflow can be created to run these tests on every submitted PR to ensure everything works as intended. Future contributors will also be required to add appropriate tests for any implemented features.*



4. Setup CI/CD Pipeline for production

The presence of a CI/CD pipeline that can automatically build updates to the app when a new commit is made to the main branch and push an updated build to beta testing and internal testing on the App Store and Google Play Store which can then be later promoted to a production build if approved by the team. This would take some of the repetitive workload off project maintainers, leaving them to only perform the final steps of promoting the build. This workflow will include automated runs of the previously added testing to ensure that all code is working as intended. GitHub Actions and Fastlane would be used to accomplish this as described in the Official Documentation . Some of this workflow has already been implemented by a fellow contributor xkaper001 (build Android Apk on a Linux runner and upload to GitHub Releases). Similarly a workflow will be made to compile and deploy an Android App Bundle as required by Google Play Store and an iOS Signed Bundle (macOS runner) to Testflight

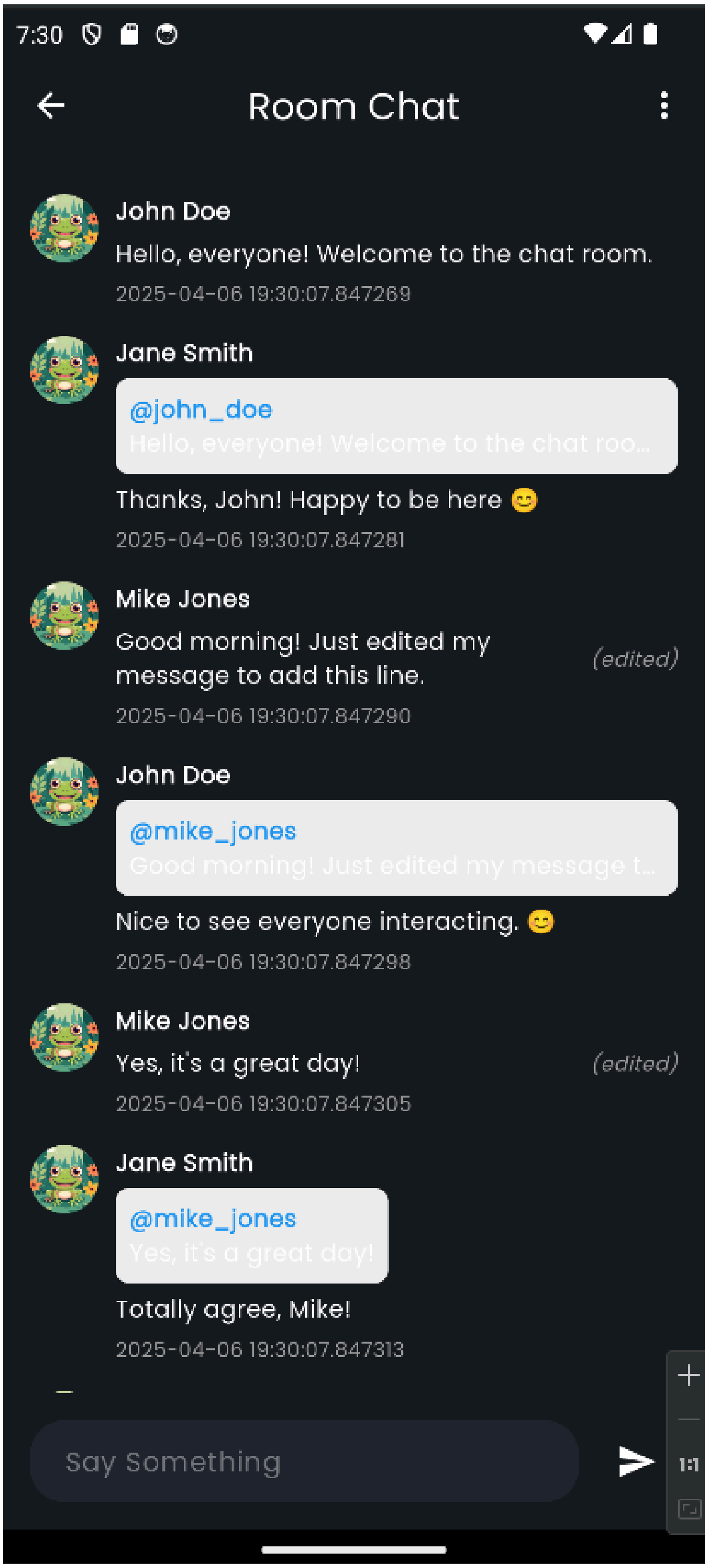


5. Complete existing Chat Functionality implementation for Rooms and add features like notifications

Resonate is currently limited by the only method of communication being audio, which some people might not always feel comfortable with. Chat functionality can be added to the Rooms to improve UX and give them more functionality. This is already partly implemented as part of the work done by previous GSoC Contributor Aarush Acharya.

The sub-tasks already completed are:

- Basic UI Implementation



John Doe

Hello, everyone! Welcome to the chat room.

2025-04-06 19:30:07.847269

Jane Smith

@john_doe
Hello, everyone! Welcome to the chat roo...

Thanks, John! Happy to be here 😊

2025-04-06 19:30:07.847281

Mike Jones

Good morning! Just edited my message to add this line. *(edited)*

2025-04-06 19:30:07.847290

John Doe

@mike_jones
Good morning! Just edited my message t...

Nice to see everyone interacting. 😊

2025-04-06 19:30:07.847298

Mike Jones

Yes, it's a great day! *(edited)*

2025-04-06 19:30:07.847305

Jane Smith

@mike_jones
Yes, it's a great day!

Totally agree, Mike!

2025-04-06 19:30:07.847313

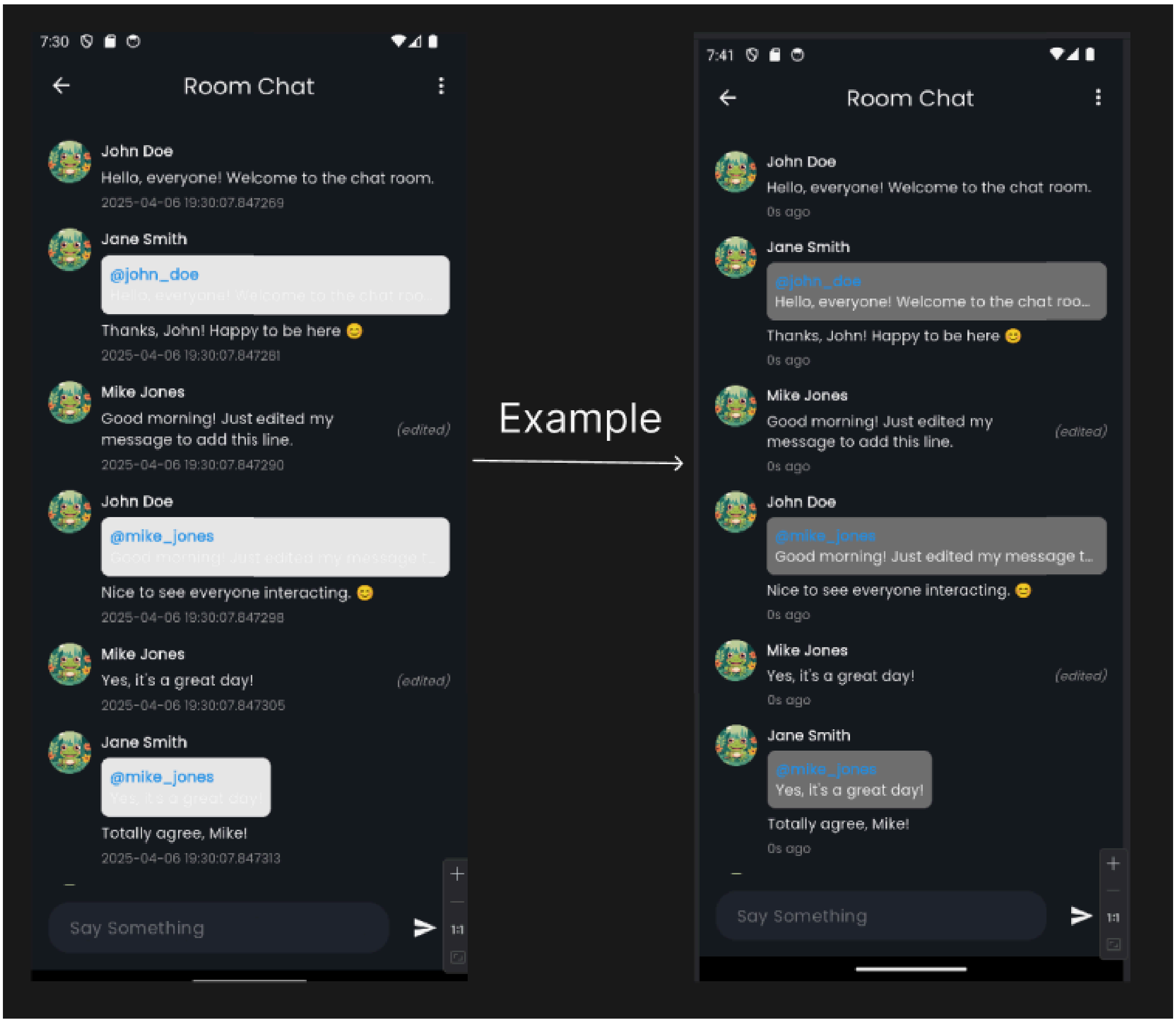
- Creation of Message Model with the following fields:

```
{
   "roomId": String,
   "messageId": String,
   "creatorId": String,
   "creatorUsername": String,
   "creatorName": String,
   "creatorImgUrl": String,
   "hasValidTag": Boolean,
   "index": Integer,
   "isEdited": Boolean,
   "content": String,
   "creationDateTime": DateTime,
   "replyTo": ReplyTo{
                         "creatorUsername": String,
                         "creatorImgUrl": String,
                         "index": String,
                         "content": String
                         }
}
```

- Basic Functionality using dummy data to test.

This task would include the following sub-tasks to complete the functionality:

- *Improve the UI for the Chat Screen*



7:30

Room Chat

John Doe
Hello, everyone! Welcome to the chat room.
2025-04-06 19:30:07.847269

Jane Smith
@john_doe
Hello, everyone! Welcome to the chat roo...
Thanks, John! Happy to be here 😊
2025-04-06 19:30:07.847281

Mike Jones
Good morning! Just edited my message to add this line. (edited)
2025-04-06 19:30:07.847290

John Doe
@mike_jones
Good morning! Just edited my message t...
Nice to see everyone interacting. 😊
2025-04-06 19:30:07.847298

Mike Jones
Yes, it's a great day! (edited)
2025-04-06 19:30:07.847305

Jane Smith
@mike_jones
Yes, it's a great day!
Totally agree, Mike!
2025-04-06 19:30:07.847313

Say Something

7:41

Room Chat

John Doe
Hello, everyone! Welcome to the chat room.
0s ago

Jane Smith
@john_doe
Hello, everyone! Welcome to the chat roo...
Thanks, John! Happy to be here 😊
0s ago

Mike Jones
Good morning! Just edited my message to add this line. (edited)
0s ago

John Doe
@mike_jones
Good morning! Just edited my message t...
Nice to see everyone interacting. 😊
0s ago

Mike Jones
Yes, it's a great day! (edited)
0s ago

Jane Smith
@mike_jones
Yes, it's a great day!
Totally agree, Mike!
0s ago

Say Something

- *Modify the backend structure to accommodate ChatModel as an Array of JSON Objects in the rooms collection and connect frontend and backend to make functionality work*
- *Add Notifications for messages and tags using the flutter_local_notifications package .*

6. Add Internationalization

As Resonate will be used by a wide global audience, and currently only supports English, it is important to add various languages and localizations so that users can use the language in an app that they are comfortable with. This can be achieved using the flutter_localizations package as outlined in the official docs . Initially the language added will be Hindi (My Bilingual Language) and other Indian Languages like Punjabi. Proper documentation will be added to the project so future contributors can add localizations to the app based on their own native languages.

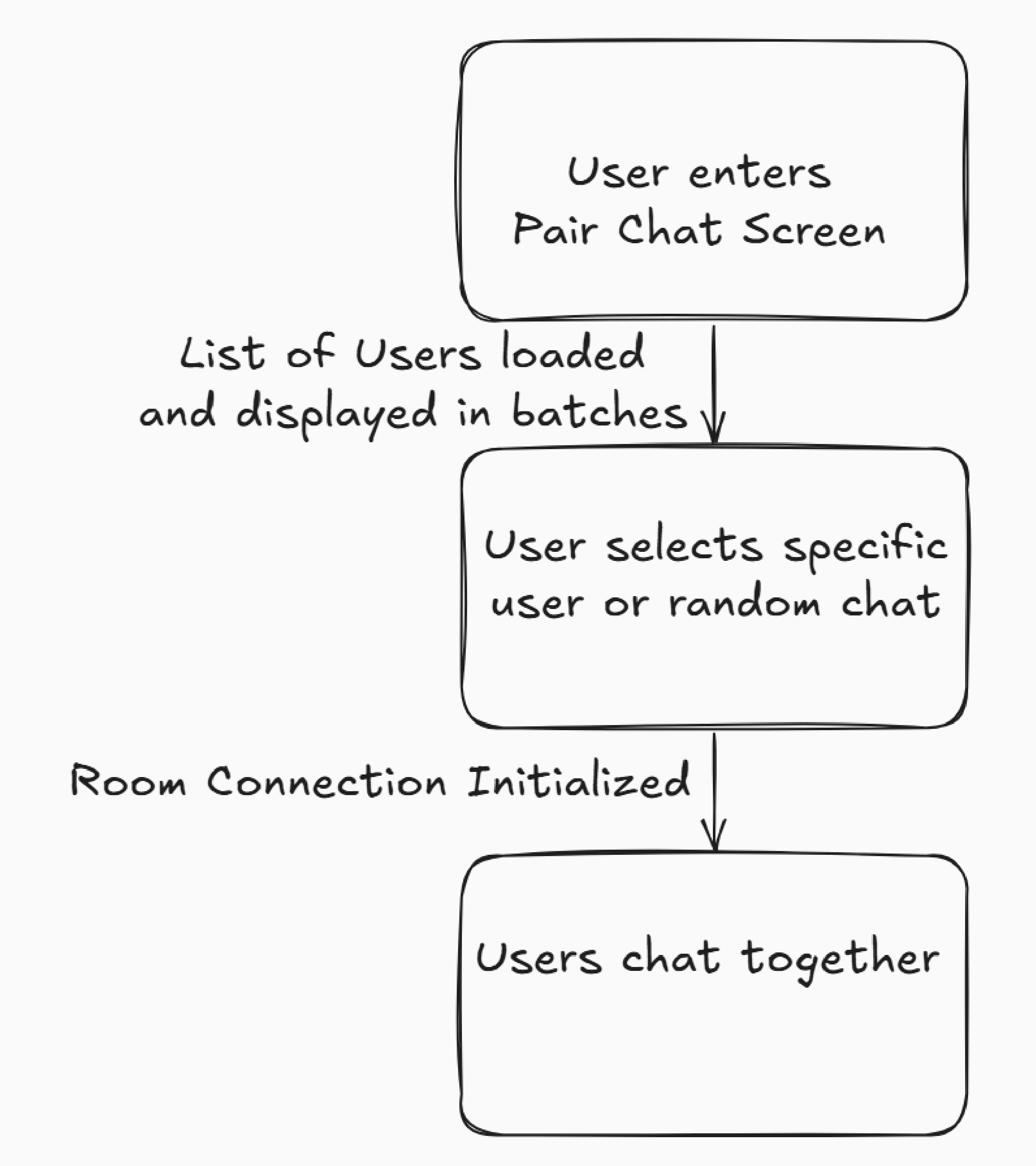
Sub-Tasks:

- Add Supported Locales and integration with *flutter_localizations* in the application
- Create files for English and Hindi Locales
- Add options to settings and home screen for user to change locales
- Add Documentation for future users to contribute and add additional Locales

7. Improve Pair Chat Experience

The current Pair Chat functionality randomly pairs up two random users for an instant chat, although it does not offer any estimates for when it will be able to do so. This experience can be improved in the following ways:

- *List all available users in the Pair Chat flow so user can select if he wants to chat with a particular user, while still keeping the option to randomly select one*
- *Implement functionality to connect with specific users*



- *Optionally also implement a "Rating system" that can act as a passive moderation system for users misusing the app for anti-social activities.*

This feature can also be reused in the "Friends Functionality" which will be discussed further below.

8. Add User Search functionality to Explore Page and implement functionality to follow creators and stay updated on their favorite ones.

The Search bar on the Explore page currently only searches through any uploaded Stories. This functionality can be expanded to search for active users and creators. This will allow the User to search for specific creators and look for new Stories and Chapters from them. This can be further expanded by adding a functionality to allow Users to follow their favorite creators and being updated as soon as the creators upload new content. This will use Appwrite Messaging to configure push notifications.

To add the Creator Follow functionality, we will add JSON Arrays to the user collection which will have the following formats:

```
"followers":{

"follower_id":<"user uid">,

"follower_username":<"user username">,

"follower_photo_url":<"user picture url">

}

"following":{

"follower_id":<"user uid">,

"follower_username":<"user username">,

"follower_photo_url":<"user picture url">

}
```

Sub Tasks:

- Add the collections to the backed and the new variables to the ResonateUser Model
- Implement buttons to enable Follow Functionality and add a Screen to show following users
- Extend search bar functionality to also look for matches in Users

- Configure push notifications for Following and Followers

9. Friends functionality

Resonate should include the functionality to “befriend” users. Implementing this functionality also gives the option to expand Pair Chats from random users to specific users you want to talk to. As an Open Source Social Voice Platform, Resonate will also become a way for even more people to connect with each other as people start using a Friends Functionality as a VoIP calling alternative as well. Unlike the improved pair chat experience, Friends will be available to call regardless of them being online on the app or not, and any calls to them will result in pop ups/notifications from the app informing them of the incoming call.

This task would involve the following sub-tasks:

- *Creation of the UI Elements for this feature, namely a screen to view friends and a screen to observe and accept/decline any received friend invites.*
- *Modifications to backend to accommodate the feature:*
  - *A new collection called “requests” will be added to the backend which contains the following fields:*

```
{

"sender_id": "<sender uid>",

"sender_username": "<sender username>",

"sender_display_name": "<sender display name>",

"sender_photo_url": "<sender picture url >",

"reciever_id": "<reciever uid>"

}
```

- *To keep a track of a user’s friends, a “friends” field of the JSON Array type will be added to the user collection with the following format:*

```
{
"friend_id": "<user uid>",
"friend_display_name": "<user display name>",
"friend_photo_url": "<user picture url>",
"friend_username": "<user username>",
}
```

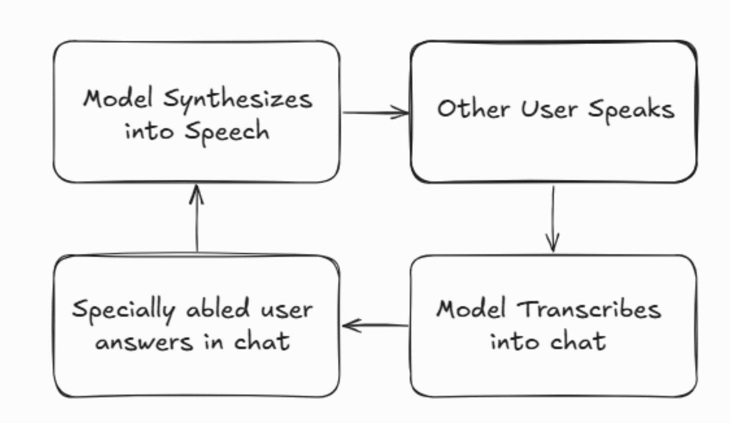
- *Addition of the buttons and business logic to send and accept friend requests*
- *Integration with Pair Chat feature to enable direct calling functionality*
- *Addition of notification on call functionalities*

10. Improve Accessibility of Resonate App

The Resonate App, as a Social Voice Platform has the majority of it's features based on Audio. This means that specially-abled non-verbal people will not be able to use the app to it's full extent. As all software is moving towards accommodating special needs, Resonate should also follow this trend. We can use State of the Art Speech Transcription and Synthesis technology to facilitate specially abled people to use Resonate. By modifying the screens and logic designed previously in this project for Chat Functionality, we can integrate these accessibility features using either Device Specific Speech to Text Capabilities or models from the Vosk family of models. Device Specific options would be preferred as that would allow us access to a greater number of languages. These audio conversations now converted to chat based conversations for the user who opts in for this feature during onboarding do not need to be stored in the database and as a result require no additional setup on the backend. This conversation history will be maintained within the app locally and be deleted once a room is closed. This same feature can also be implemented in Pair Chats and the Friend Calling systems. The Stories and Chapters already provide lyrics so Transcription will not be needed there.

The subtasks involved will be:

- Discuss with mentor on whether to use Vosk Models or On-Device Functionality
- Create Pipelines for data processing
- Add Options to the Settings and Onboarding for user to enable accessibility mode
- Integrate Transcription and Synthesis into Calls and Rooms (Can be integrated into the chat functionality for a seamless experience)


Model Synthesizes into Speech
Other User Speaks
Specially abled user answers in chat
Model Transcribes into chat

11. Add Host Permissions to Discussions

The Resonate App currently offers no options for Hosts to control participants in rooms. These features are prominent in all software that offer audio room features. These features should also be added to Resonate using the *Participant Management API* *. This can be done using the features from the Livekit SDK and the ParticipantPermissions class to manage participants and can also be used to "moderate" the platform, wherein room hosts can mute or even remove any troublemakers in their rooms. The use of this in moderation can be extended to including a functionality to report a user for misconduct.*

Subtasks:

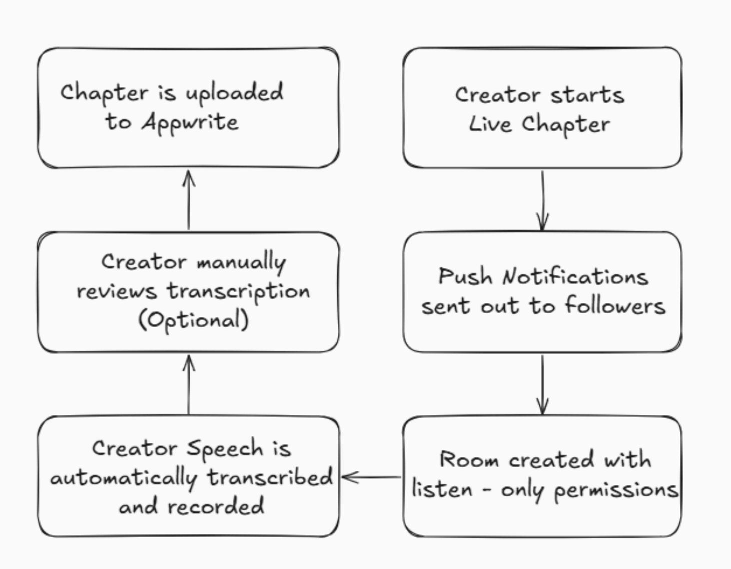
- Write functions to change Participant Permissions
- Add Buttons to Chat Screen to control audience for hosts
- Add appropriate reporting of events to users

12. Add Live Chapters Feature

Currently the only way to upload Story Chapters to Resonate is to have them pre recorded, but what if an author wants to live stream a chapter to their *audience and then add it to their story. This feature addition would make that possible, using the host permissions, creator follow functionality and speech transcription functionalities to build this feature where a room would be created in which only the author has the permission to speak. All users following the creator will receive a notification on their devices to tune in, and Captions for the chapter may automatically be generated in real time, which the speaker can then edit after ending his live stream. This stream will be recorded and the resultant audio can then be added as a chapter to a previously uploaded Story along with the generated transcriptions.*

Subtasks:

- Add option for User to start live chapter
- Using previously implemented features like Host Control and Creator Push Notifications initiate a room with only listening and chat permissions for general users and send notifications to all followers
- Re-using the previously implemented Accessibility features channel audio stream through the transcription pipeline to transcribe the chapter and create lyrics.
- Handle closing of the room by creating and uploading the chapter automatically after giving user an option to review transcription.


Creator starts Live Chapter
Push Notifications sent out to followers
Room created with listen - only permissions
Creator Speech is automatically transcribed and recorded
Creator manually reviews transcription (Optional)
Chapter is uploaded to Appwrite

13. Add better search functionality

Currently Resonate uses Appwrite Queries to search for things in the database. This approach work's fine for now but it will not scale well when the number of documents climbs into the thousands. Additionally it is not real-time or built to be used as a quick search tool and lacks support for advanced searching techniques. Instead we can use a Search Function that uses Meilisearch which is an Open Source AI Search Engine which can be easily linked with Appwrite using the process detailed in this blogpost . This will allow us to have a Search Function that scales appropriately and predictably, and allows us to make use of it's more advanced Search Functionalities like semantic search.

The sub-tasks for this task will include:

- Self-host an instance of Meilisearch
- Automate the self hosting procedure and add it to the backend setup repo along with proper documentation
- Write the cloud functions for integration with Meilisearch and deploy them on the backend. Also add these to the backend setup repo
- Replace existing Query based searches with new Meilisearch based search.

14. Merge all changes to main branch and deploy v2 of Resonate App

At this point Resonate would have all the Quality of Life Improvements and New Features implemented throughout the duration of my project and the Tests would be ensuring that everything is working. With a final run of all the tests and thoroughly testing all user flows, we would merge all changes to the main branch and the previously added CI/CD would compile and push a build to the app stores. Then we would go through the developer portals on both the stores, complete the formalities and push an updated v2 of the Resonate App to the stores, marking the end of my Project.

Timeline

Date/Week	Task
Bonding Period May 8 - June 1	• Connect with project mentor and other maintainers to confirm and discuss key details about implementation of various features • Bring the App to a stable state and work on UI Improvements and fixes • Perform initial setup for App Deployment on App Stores in collaboration with Mentor (Data Safety Questionnaires, etc.) • Upload app builds to internal testing and submit for review

Date/Week	Task
Week 1 June 1 - 7	• Implement App Store and Play Store review feedback, if any, and promote builds to open beta testing. • Write comprehensive tests to the app • Add documentation specifying contributors to include tests with all future PRs
Week 2 June 8 - 14	• Write CI/CD Pipeline for uploading builds to internal testing and deploy on GitHub Actions • Write a GitHub action to run all tests on any new PRs and to check for addition of new tests • Open Sample PRs to check workflows and debug any issues
Week 3 June 15 - 21	• Start Implementing Chat Functionality • Modify and improve UI of Chat Screens • Add new variables in backend and connect Frontend Code with Backend
Week 4 June 22 - 28	• Test Chat flows • Add Notification on message functionality
Week 5 June 29 - July 5	Implement complete Internationalization functionality in the app.
Week 6 July 6 - 12	• Implement UI for improved Pair Chats, i.e. Load and display list of available users • Implement functionality to pair with selected user
Week 7 July 13 - 18	Implement User Rating system for pair chats

Mid-Term Evaluation (Date: July 18th, 2025)

Date/Week	Task
Week 8 July 20 - 26	• Add User Search functionality to explore screen • Add Creator Follow functionality • Add push notifications for following and followed
Week 9 July 27 - August 2	• Add Variables and Functions for Friends Functionality to Backend • Implement UI Screens for Friends Functionality • Write business logic for friend functionality
Week 10 August 3 - 9	Add Direct Calling Feature for Friends Functionality
Week 11 August 10 - 16	• Add Notification and ringing on direct calls • Implement UI Elements for offering accessibility options in

Date/Week	Task
	Settings and onboarding
Week 12 August 17 - 23	• Create pipelines for data processing for accessibility features • Integrate Accessibility features with chat feature
Week 13 August 24 - 30	Add Host Controls for Rooms
Week 14 August 31 - September 6	Add Support for Live Chapters
Week 15 and 16 September 7 - 20	Add Better Search Functionality
Week 17 and 18 September 21 - October 4	• Merge all recent changes into main • Test the app and all flows thoroughly • Check Test Coverage • Deploy Resonate v2 to Stores • Gather User and community feedback and iteratively work on integrating feedback to improve on added features

Final Evaluation (Date: October 6th, 2025)

Any Other Relevant Information

I'm a huge FOSS enthusiast, and I've been actively working to spread awareness about Free and Open Source Software in my college. Along with a batchmate, I started the FOSS United Chapter here, there wasn't much of an open-source culture and most people didn't even know what FOSS really stood for. Now, we have a thriving community where people are learning, contributing, and actually understanding why open-source matters.

I also really Resonate (pun intended) with Richard Stallman's views on what makes software truly free. It's not just about the code being open but about ensuring users actually have control over the software they use. That's something I strongly believe in, and I want to do my part in adding value to the FOSS ecosystem.

So trust me when I say Resonate isn't just another project to me and that it represents something much bigger. I want to see it grow into a truly free and open voice platform that people actually use, not just as an alternative, but as the go-to choice.