SoundPalette Project Progress Report 2

Brock University

Course: COSC 4P02 - Software Engineering II

Professor: Naser Ezzati-Jivan

Date: March 23rd, 2025

Parth Chauhan
pc19kt@brocku.ca
6860795

Victoria Danh
vd19qe@brocku.ca
6851547

Prab Khokhar
pk21le@brocku.ca
7348915

Kaija Sproxton
ks21lk@brocku.ca
7267131

William White
ww18fj@brocku.ca
6714091

James Windjack
jw17rm@brocku.ca
6366371

Table of Contents:

Our Mission:

SoundPalette is a dedicated social media platform designed to revolutionize the way artists and musicians connect, collaborate, and grow. Traditional social media often fails to bring together creatives with complementary skills, leaving artists struggling to find the right collaborators who match their style and vision. SoundPalette solves this problem by providing a tailored networking experience where lyricists, musicians, producers, and visual artists can seamlessly discover and engage with like-minded creators. With tools to showcase work, connect based on artistic compatibility, and even monetize content through premium subscriptions and exclusive events, SoundPalette empowers artists to turn inspiration into reality. Our mission is to break down barriers, foster meaningful connections, and build a thriving creative community.

Team Structure:

Team Member	Role
Parth Chauhan	Product Owner (Test Developer)
Victoria Danh	Scrum Master (Front End Developer)
Prab Khokhar	UX Designer (Front End Developer)
Kaija Sproxton	Documentation Specialist (Front End Developer)
William White	Development Lead (Full Stack Developer)
James Windjack	Assistant Developer (Full Stack Developer)

GitHub Repositories:

The following GitHub repositories contain all pertaining code, assets and documents related to the project:

Main Repository: https://github.com/vd19qe/SoundPalette

API Server Repository: https://github.com/WilliamSEWhite/SoundPaletteApiServer

UI Repository: https://github.com/WilliamSEWhite/SoundPaletteUI

Timeline:

The following is our tentative task schedule:

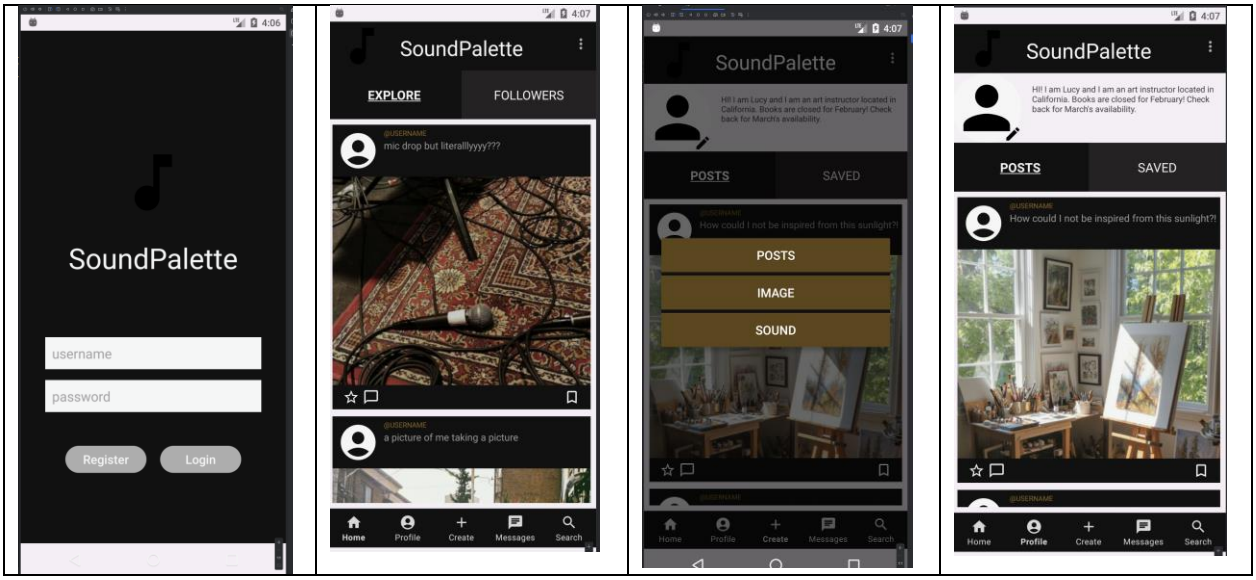
Dates	Task
Jan 6th – Jan 12th	~~☐ Create Group~~ ~~☐ Create Project Proposal – Due Jan 12th~~
Jan 13th – Jan 19th	~~☐ Create User Stories~~ ~~☐ Create Product Backlog~~ ~~☐ Create Sprint Backlog~~

	☒ ~~Create Release Planning Doc – Due Jan 19th~~
Jan 20th – Jan 25th	☒ ~~Sprint 1~~
Jan 26th – Feb 8th	☒ ~~Sprint 2~~ ☒ ~~Sprint 2 Retrospective Meeting~~
Feb 9th – Feb 23rd	☒ ~~Sprint 3~~ ☒ ~~Sprint 3 Retrospective Meeting~~ ☒ ~~Create Progress Report 1 – Due Feb 23rd~~
Feb 24th – March 8th	☒ ~~Sprint 4~~ ☒ ~~Sprint 4 Retrospective Meeting~~
March 9th – March 22nd	☒ ~~Sprint 5~~ ☒ ~~Sprint 5 Retrospective Meeting~~ ☒ ~~Create Progress Report 2 – Due March 22nd~~
March 23rd – April 5th	☐ Sprint 6 ☐ Sprint 6 Retrospective Meeting
April 6th – April 11th	☐ Sprint 7 ☐ Sprint 7 Retrospective Meeting
April 12th – April 25th	☐ Sprint 8 ☐ Final Presentation ☐ Submit Final Report

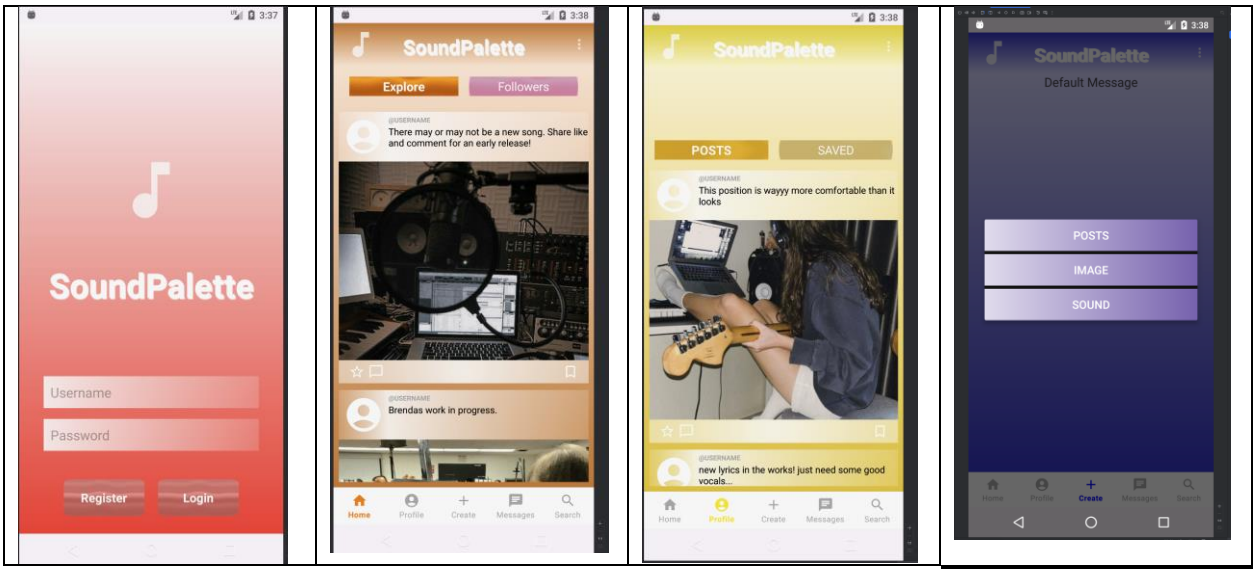
**Note – Our Detailed Sprint Schedule and Backlogs can be found under the Project's tab of our Main GitHub Repository.*

Current Progress:

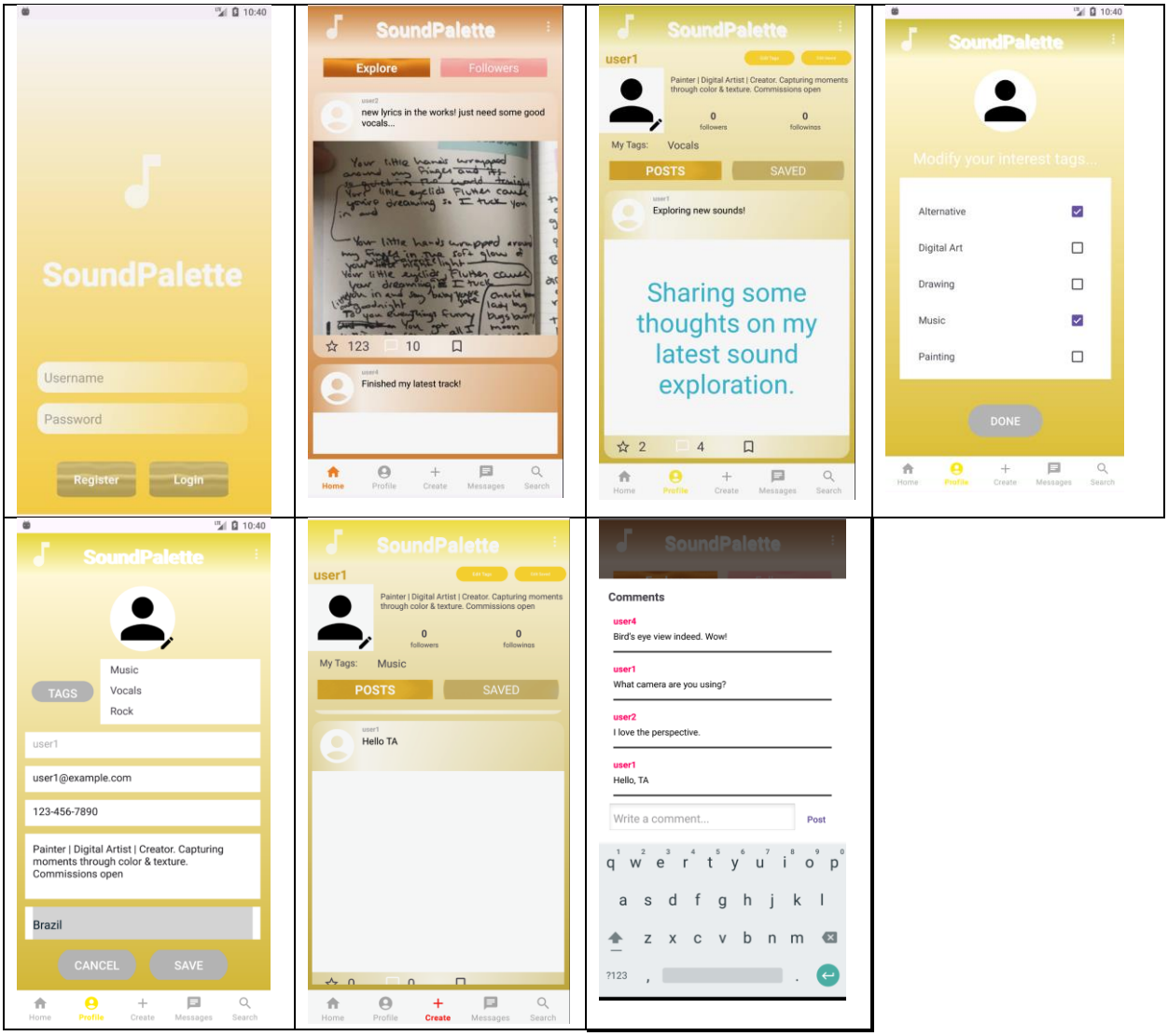
UI Progress:



**Note – Screenshots of Initial UI implementation*



**Note – Screenshots of Second Iteration of UI implementation*

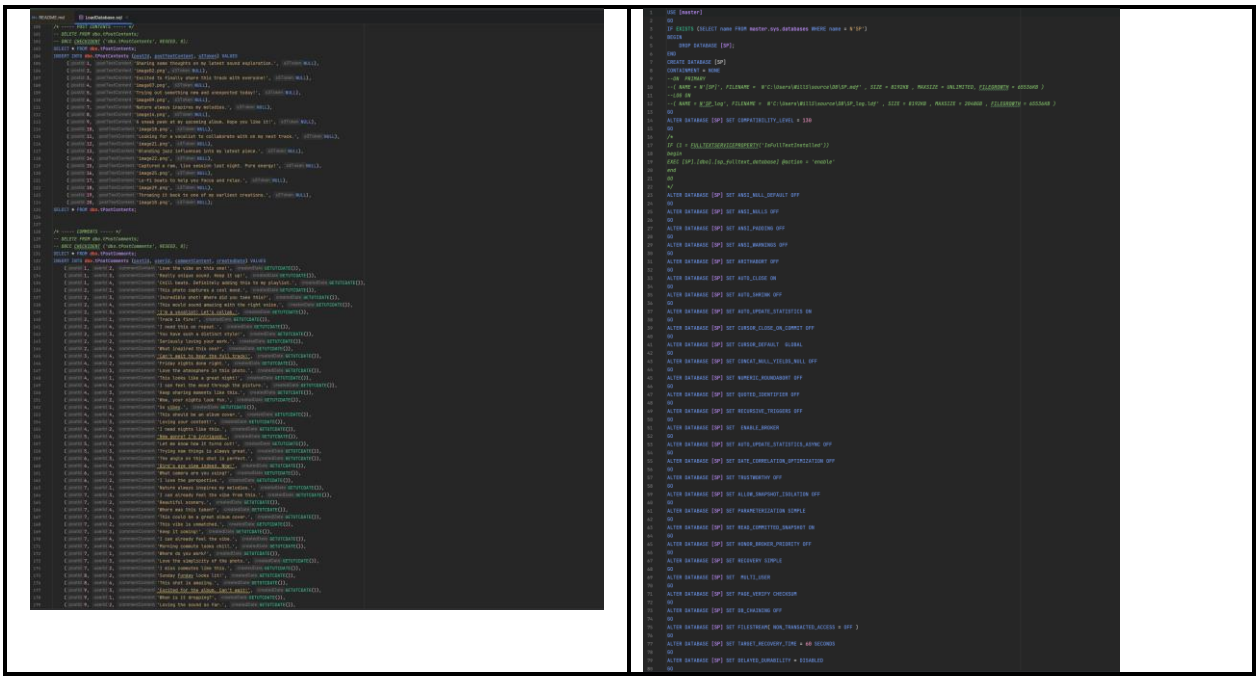


**Note – Screenshots of Current UI implementation*

The SoundPalette UI has undergone several meaningful updates that have brought core features closer to completion. One of the most significant milestones has been the successful implementation of the Posts functionality, which now allows users to create, view, and interact with content within the app. Additionally, the Tags system is now fully functional, enabling users to select, save, and display personalized tags during registration and within their profiles. The User Registration process has also been expanded to include profile biography fields, allowing for greater personalization, while Profile Edit capabilities have been added so users can modify their bios and tag selections after account creation. Comments are also fully functional.

Visually, the UI has seen improvements aimed at polishing the user experience and enhancing consistency. Rounded corners have been applied to containers and input fields across the app, reinforcing a clean and cohesive design language. While broader UI transformations have been slower due to both members of the main front-end team Kaija and Prab being sidelined with the flu for two weeks, small refinements have been made throughout the interface. These include improvements to layout spacing, button placement, and visual transitions, setting a strong foundation for the upcoming implementation of the planned glass morphism design. Overall, while some progress was delayed, the team has made steady strides toward a more functional and user-friendly interface.


Solution
SoundPalette · 1 project
SoundPaletteApiServer
Dependencies
Properties
launchSettings.json
Controllers
BaseApiController.cs
LocationController.cs
LoginController.cs
PostController.cs
PostInteractionController.cs
TagController.cs
UserController.cs
Data
LoadDatabase.sql
SoundPaletteDb.sql
SPContext.cs
DataModels
tLocation.cs
tPost.cs
tPostComment.cs
tPostContent.cs
tPostLike.cs
tPostSave.cs
tPostTag.cs
tPostType.cs
tTag.cs
tUser.cs
tUserFollower.cs
tUserInfo.cs
tUserProfile.cs
tUserTag.cs
DbHelpers
DbHelperBase.cs
LocationDbHelper.cs
LoginDbHelper.cs
PostDbHelper.cs
PostInteractionDbHelper.cs
TagDbHelper.cs
UserDbHelper.cs
Facade
FacadeBase.cs
LocationFacade.cs
LoginFacade.cs
PostFacade.cs
PostInteractionFacade.cs
TagFacade.cs
UserFacade.cs
Models
ChatRoomModel.cs
CommentModel.cs
LocationModel.cs
MessageModel.cs
NewChatRoomModel.cs
NewMessageModel.cs
NewPostCommentModel.cs
NewPostModel.cs
PostContentModel.cs
PostModel.cs
PostTypeModel.cs
TagModel.cs
UserInfoModel.cs
UserModel.cs
UserProfileModel.cs
appsettings.json
appsettings.Development.json
Program.cs
SoundPaletteApiServer.http
Startup.cs
app
manifests
AndroidManifest.xml
java
com.soundpaletteui
Activities
Home
Interactions
LoginRegister
Messages
Posts
Profile
Trending
MainScreenActivity
Infrastructure
Adapters
CountrySelectAdapter
MainContentAdapter
TagRowAdapter
TagSelectAdapter
UserTagAdapter
ApiClients
CommentClient
LocationClient
LoginRegisterClient
PostClient
PostInteractionClient
TagClient
UserClient
ApiEndpoints
CommentApiEndpoints
LocationApiEndpoints
LoginRegisterApiEndpoints
PostApiEndpoints
PostInteractionApiEndpoints
TagApiEndpoints
UserEndpoints
Models
ChatMessageModel
ChatroomModel
CommentModel
LocationModel
NewChatroomModel
NewMessageModel
NewPostCommentModel
NewPostModel
PostContentModel
PostModel
PostTypeModel
TagModel
UserInfoModel
UserModel
UserProfileModel
Utilities
AppSettings
Navigation
UISettings
SPWebApiRepository
com.soundpaletteui (androidTest)
ExampleInstrumentedTest



**Note – Screenshots of Database implementation*

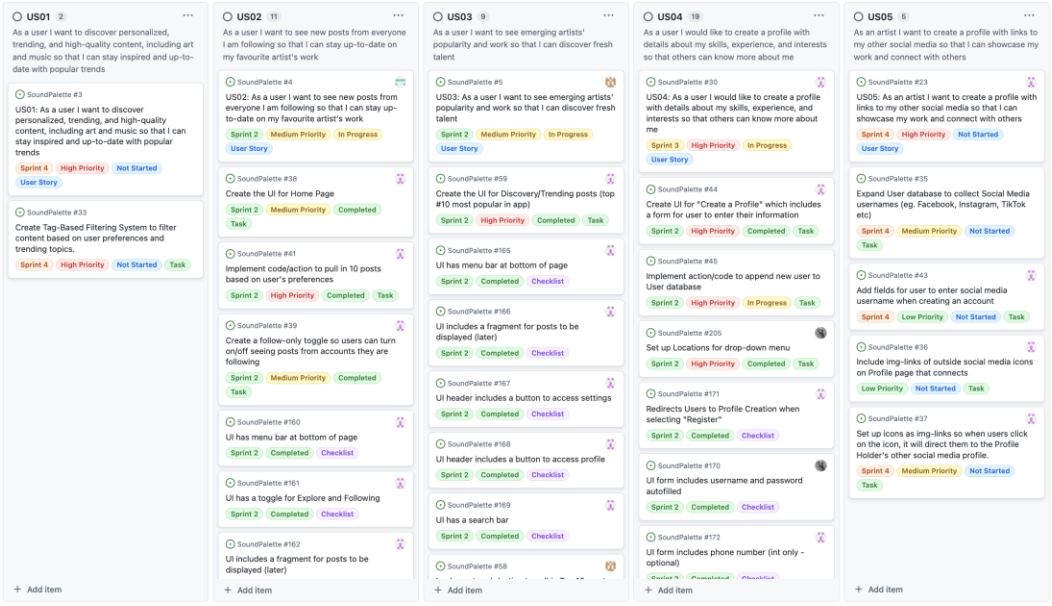
The backend of SoundPalette has expanded significantly to support a wide range of core features, transforming the app from a static prototype into a fully functional, data-driven platform. One of the major backend achievements has been the development of post interaction services, including endpoints for liking, saving, and commenting on posts. These were essential in enabling real-time engagement within the app and are now fully integrated with the front-end. Additionally, the backend now supports more complex user-related operations, such as following and unfollowing, retrieving user profiles, and viewing posts by user or saved status, all of which have laid the groundwork for a more connected user experience.

Another key advancement was the completion of the User Profile update and tag management systems. Functionality was added to allow users to edit their bios and profile tags, and new endpoints were created to read, write, and update this data in the database. The backend was also modified to handle null values more gracefully, particularly during registration to prevent crashes and ensure data integrity. Significant effort has gone into merging branches and resolving code conflicts, with James and William spending considerable time ensuring backend stability as new features were layered in. With the majority of core backend services now operational, the focus is shifting to optimization, cleanup, and final integration with the front-end for a seamless user experience. Will is currently working on and is almost done with the messaging features which will add an important functionality to our app.

Scrum Methodology:

Our team follows the Scrum methodology to ensure an organized agile development process for SoundPalette. We use User Stories to define features and improvements from an end-user perspective, which are managed in our Product Backlog. Each sprint, we prioritize tasks into a Sprint Backlog and work through them systematically. We hold regular meetings twice a week to discuss progress, address challenges, and adjust priorities as needed. At the end of each sprint, we conduct a Sprint Retrospective meeting to reflect on what went well and identify areas for improvement. All our sprint planning, progress tracking, and task management are documented in GitHub Projects. Below are the screenshots from our GitHub projects that can be found under the Main GitHub repository.

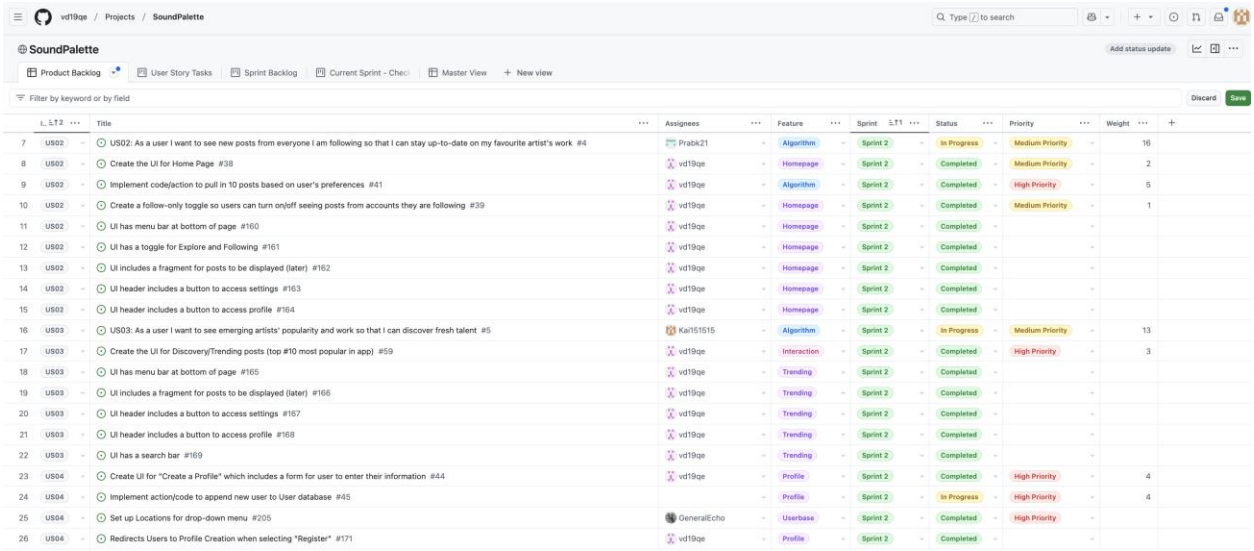
User Story Tasks:



Link: https://github.com/users/vd19qe/projects/5/views/5

There are 30 User Stories that we have created. Under each User Story are Tasks that are required to complete that User Story. Each Task has a Priority (Low, Medium, High), a Weight which indicates how much work that Task might take, and an assigned team member in charge of that Task.

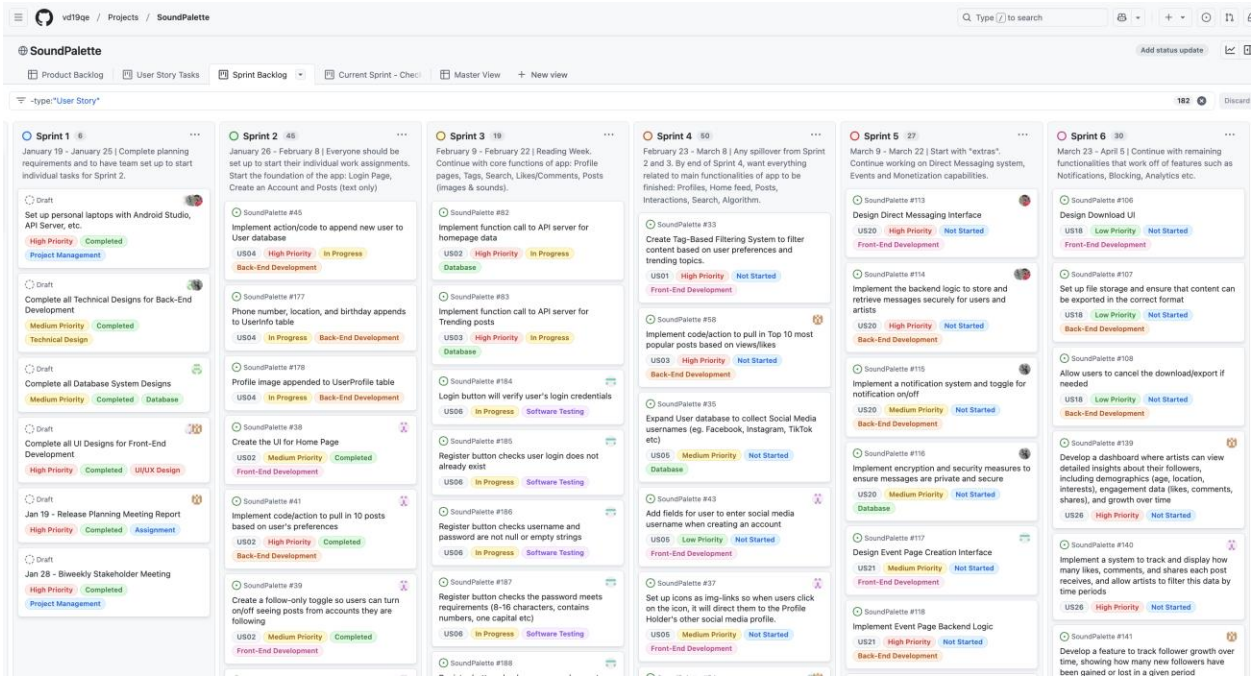
Product Backlog:



Link: https://github.com/users/vd19qe/projects/5/views/1?filterQuery=

Our Product Backlog contains 163 items, these items are in the form of User Stories, Tasks, or important deadlines/meetings. Each item has an assigned team member, the app feature and sprint it belongs to as well as a status, priority, category, and weight that it consists of.

Sprint Backlog:



Link: https://github.com/users/vd19qe/projects/5/views/9

Our Sprint Backlog details the total requirements of all of our 8 Sprints. Each Sprint consists of the User Stories, Tasks and important deadlines/meetings belonging to that Sprint.

We have also included a quick description of what the main goals of that sprint are as well as the timeline it falls under.

Testing:

This is the current Test Case Table which showcases which Test Cases to validate. This is still being planned and was created by Parth.

Test Case ID	Description	Boundary Cases	Test Type	Target
AUTH-001	User Registration with Valid Data	Max email length, max password length	Functional	Frontend/Backend
AUTH-002	User Registration with Missing/Invalid Fields	Empty fields, invalid email, weak password	Functional	Frontend/Backend
AUTH-003	Login with Valid Credentials	Max email length, max password length	Functional	Frontend/Backend
POST-001	Edit Post - Change Picture	Largest and smallest image sizes	Functional	Frontend/Backend
POST-002	Edit Post - Change Music	Largest music file, unsupported formats	Functional	Frontend/Backend
POST-003	Edit Post - Change Text	Max character limit, empty caption	Functional	Frontend/Backend
POST-004	Cancel Edit	Cancel with large images or changes	Functional	Frontend
SEC-001	Edit Post - Authorization Check	Different session, URL manipulation	Security	Backend
PERF-001	Post Edit Performance Under Load	100 concurrent users, media-heavy posts	Performance	Backend
USAB-001	Post Editing Interface Consistency	Different screen sizes, accessibility features	Usability	UI
FRONT-001	Navigation Bar Visibility	Proper display across devices	Functional	Frontend/UI
BACKEND-001	Database Validation for Post Edits	Proper update of text, media	Functional	Backend
UI-001	UI Consistency across Browsers	Test on Chrome, Firefox, Safari	Usability	UI
BACKEND-003	Post Ownership Validation	Attempt post edit by different user	Security	Backend

Challenges:

Since the last report, our team has encountered a new set of challenges that have impacted our development timeline and workflow. One major setback was an unexpected illness affecting both Kaija and Prab, who work closely together on the UI. As a result, progress on several front-end components was delayed, requiring the rest of the team to adjust task distribution and timelines accordingly. In addition, we faced recurring technical issues specific to macOS, which prevented some team members from running the app on their local machines. James dedicated an entire day to diagnosing the problem, eventually identifying an Android API

level mismatch and provided a custom virtual machine and step-by-step setup guide to resolve the issue for Mac users. Another ongoing challenge has been managing merge conflicts and ensuring the integrity of our codebase. With multiple team members working across different branches, James and William have spent significant time coordinating merges and resolving conflicts to prevent broken builds and maintain a stable development environment.

Team Contribution Summary:

*Note- Red has the newest changes since the last report

Parth Chauhan	▪ Refactored and cleaned up code to improve readability, maintainability, and scalability ▪ Implemented unit testing using JUnit and Mockito for user login, registration, and validation functions ▪ Managed dependencies and configurations to optimize performance and ensure compatibility with different Android versions ▪ Enhanced error handling and logging to improve debugging and app monitoring ▪ Added documentation and comments for cleaner codebase and README files with clear instructions ▪ Created additional unit tests for user session handling and input validation ▪ Developed a comprehensive Test Case Table to document and organize test scenarios
Victoria Dahn	▪ Scrum Master Responsibilities: o Organized and led weekly team meetings, including booking spaces, setting up invites, preparing objectives, and documenting key takeaways. Coordinated bi-weekly stakeholder meetings, prepared PowerPoint presentations and encouraged team participation o Led timeline reassessment at the end of Sprint 3, guiding the team in removing Events and Monetization features to keep scope manageable o Continued organizing weekly team meetings and coordinating bi-weekly stakeholder meetings, including presentation prep and documenting meeting minutes on GitHub ▪ Database & UI Design: o Assisted in brainstorming and refining database and UI designs, and digitalized whiteboard sketches into structured mock-ups o Created a script to pre-populate the Firebase database with users, posts, comments, likes, and saved content for testing and debugging purposes

	▪ GitHub Project Board Management: ○ Worked alongside Kaija and Parth to structure and transfer user stories into GitHub, ensuring a well-organized project backlog ○ Continues updating board and reassigning/rescheduling tasks ▪ Android App Front-End Development: ○ Implemented key front-end components, including button navigation menu, home page, profile page, trending page, and post display system. Also created placeholder data to structure pages and ensure smooth navigation. ○ Developed interactive features including likes, comments, saved posts, and editable profile/post fields ○ Built the comments UI using RecyclerView and custom CommentAdapter with placeholder logic to support later API integration ○ Started Direct Messaging feature by designing the initial UI and laying the groundwork for future back-end integration
Prab Khokhar	▪ GIF-Based Button Animations: ○ Created new GIF assets for Explore, Follower, Posts, and Saved buttons, replacing static icons with looping animations for more engaging user interactions. ▪ Translucency Gradients: ○ Developed a white translucency xml file to enhance the UI with layered transparency, creating a modern, glass-like effect. ○ Collaborated with Kaija to plan the integration of a glass morphism design system, including transparency layering and component structure ▪ Rounded Corners & Minimalism: ○ Applied curved input fields to registration and search pages, balancing vibrant colors with subtle overlays for a clean user-friendly design. ○ Applied rounded corner designs to containers and boxes across the app to reinforce the cohesive visual language ▪ Playful & Cohesive UI: ○ Maintained brand consistency across tabs by integrating transparent overlays and color-coded elements, ensuring unified visual style. ○ Refined the layout and usability of the Edit Profile and Edit Tags pages for a more intuitive user experience ▪ Asset Development: ○ Experimented with gradient variations across multiple screens, ensuring layered effects blend smoothly.

	○ Currently creating an asset for a background image that will animate giving the app more of a creative touch
Kaija Sproxton	▪ Documentation & Reports: ○ Solely responsible for writing the Proposal, Release Planning Document, and this report (Progress Report 1), handling all formal documentation. ○ Sole author of the current Project Progress Report 2, ensuring all recent updates, challenges, and contributions were professionally documented ▪ Color-Coded Navigation: ○ Assigned distinctive hues for Home, Profile, Messages, and Search tabs, ensuring each section had a unique color identity ▪ Text Animation: ○ Designed a custom animated text effect for the brand's name, utilizing timed keyframes for smooth fade and scale transitions. ▪ Login & Register Color Transitions: ○ Introduced a gradient effect for a welcoming and modern first impression, ensuring readability of input fields. ▪ Overall Design Direction: ○ Ensured visual consistency across the UI, unifying elements like typography, colors, and animations to maintain a cohesive look. ○ Collaborated with Prab to plan the integration of a glass morphism design system, focusing on playful, creative layering and transparency effects ▪ Communication: ○ Maintained clear and consistent communication with the team throughout her recovery, sharing concerns and updates to keep everyone aligned
William White	▪ Project Setup & Infrastructure; ○ Created the base Android project with a login page ○ Developed basic Kestrel API server for seamless connection between front-end and backend ○ Created Git repositories for both UI and API server. ▪ API & Database Development: ○ Integrated Microsoft Entity Framework tand created the initial database version with scripts to support user creation, profiles, and posts. ○ Added services to the API server for user registration, profile creation, and login functionality. ○ Developed functionality to add posts to the database with the API server, as well as create and retrieve posts from the API server.

	○ Implemented API services for post interactions, including comments, likes, and saved posts ○ Added backend support for following and unfollowing users ○ Created services to retrieve various post sets (e.g. by current user, by username, and saved posts) ○ Developed backend functionality for viewing other users profiles ○ Built out complete backend services to support direct and group messaging features ▪ Front End Functionality: ○ Updated UI to connect post interaction features (comments, likes, saves) with live backend data ○ Integrated new post retrieval services into the front-end to dynamically populate content ○ Replaced placeholder UI elements with real-time functionality for a responsive user experience ▪ Merging Changes: ○ Merged code contributions from all team members into a unified build, resolving conflicts and ensuring project stability
James Windjack	▪ Proposed the SoundPalette app idea ▪ Set up Development Environments for Mac Users: ○ Spent Significant time configuring C# API server and MSSQL database using Docker, ensuring compatibility across Intel and Apple devices ▪ Assisted Team Members in Troubleshooting: ○ Helped resolve issues related to Docker images and shell command errors ▪ Created Initial UI mock-up: ○ Did this for the main screen providing a foundation for the UI team ▪ Implemented Register Activity UI: ○ Added user profile fields and photo upload functionality ▪ Connecting App & API Server: ○ Learned C# and Retrofit to ensure smooth communication between the app and server. ▪ Fixed API Server Issues: ○ Corrected code for retrieving location data and resolved issues with saving user profiles to the database. ▪ User Profile Update Functionality: ○ Refactored ProfileEditFragment for modularity and maintainability ○ Fixed location spinner by modifying table data and resolving API server issues with returning location IDs ○ Created backend procedures and API endpoints for reading/writing user profile data

	○ Connected front-end UI to backend for real-time profile updates, including bios ▪ User Tags System: ○ Designed and implemented UI for selecting and editing user tags during registration and profile editing ○ Built layout adapters and API clients to send/receive tag data between UI and server ○ Developed API endpoints and database procedures for adding, updating, and retrieving tags ○ Created new UI fragments for tag display and integrated them into the profile interface ▪ Bug Fixes & Stability Improvements: ○ Fixed crashes related to null bio data and missing database columns by adding default values and updating schema ○ Rewrote navigation system to manage fragment stack and prevent duplicate entries causing app crashes ○ Debugged post navigation issues and implemented logic to return users to correct fragment after actions ▪ Environment Setup & Team Support: ○ Resolved critical issue preventing Mac users from running the app by identifying Android API level conflicts ○ Built and shared a compatible virtual machine, including detailed setup instructions for team members ▪ API Server Maintenance: ○ Resolved merge conflicts and fixed issues that temporarily broke the API server ○ Spent hours troubleshooting async UI update problems to ensure smooth data transfer from API to UI ▪ Infrastructure Research: ○ Investigated cloud hosting options and identified Amazon S3 as the most suitable platform for backend deployment

Team Contribution (Old) In-depth:

Parth Chauhan:

Parth worked on refactoring and cleaning up the codebase to improve readability, maintainability, and scalability. He removed unused imports, optimized methods, and simplified logic in key areas such as user login and registration. Additionally, he focused on implementing unit testing using JUnit and Mockito to ensure the reliability of the logic. Parth created test cases for user registration, login logic, and validation rules, allowing the team to verify core functionality without relying on backend systems or UI components.

To maintain compatibility across different Android versions, Parth updated project dependencies and optimized build.gradle files for better performance. He also enhanced error handling to ensure the app could gracefully handle API failures and unexpected user interactions. To support better monitoring and debugging, he integrated logging mechanisms to track user activity and app performance. Finally, Parth contributed significantly to documentation, adding detailed comments throughout the codebase and updating README files with clear setup instructions, test execution guidelines, and build processes, making it easier for future developers to work on the project. He also created an extensive list of regions by continent as well as many popular tags for posts.

Victoria Dahn:

As the Scrum Master, Victoria played a crucial role in organizing and leading weekly team meetings by booking meeting spaces, setting up calendar invites, and preparing structured objectives. She ensured meetings remained productive by facilitating discussions and documenting key takeaways for future reference. Additionally, she coordinated bi-weekly Stakeholder meetings, preparing PowerPoint presentations to showcase the team's progress. Victoria encouraged active team participation in these meetings, strengthening engagement with stakeholders and ensuring all contributions were acknowledged. All meeting records are available in our GitHub repositories.

Beyond her Scrum Master duties, Victoria contributed significantly to the UI and database design process, working closely with the team to brainstorm and refine initial concepts. She digitized whiteboard sketches into structured mock-ups, which were shared with the team and uploaded to GitHub for easy reference. She also played a key role in managing the GitHub project board, collaborating with Kaija and Parth to structure and transfer user stories effectively.

In addition to her leadership responsibilities, Victoria actively contributed to the development of the SoundPalette Android app. During Sprints 2 and 3, she implemented essential front-end components, including the button navigation menu, home page, profile page, trending page, and post display system. Thanks to James and Will who handled setting up the main activities and API connections, Victoria was able to structure the UI using placeholder data,

ensuring smooth navigation and interaction between screens. Her work provided a strong foundation for future database integration, allowing Prab and Kaija to refine the UI for an improved user experience.

Prab Khokhar:

Prab played a significant role in enhancing the visual and interactive elements of SoundPalette. One of his primary contributions was developing GIF-based button animations for key user interactions, including Explore, Follower, Posts, and Saved buttons. These animations replaced static icons, creating a more dynamic and engaging user experience. His goal was to introduce playful yet polished transitions that maintained the app's vibrant and modern aesthetic.

Additionally, Prab focused on implementing translucent gradient overlays across multiple screens, including Home and Profile pages. He developed white_translucency_gradient.xml and its variations to introduce 50–80% opacity overlays, ensuring a subtle, glass-like effect that allowed background colors to shine through. To unify the app's look, he applied rounded corners to input fields on the registration and search pages, reinforcing a minimal yet modern UI style.

Prab also played a key role in refining UI consistency across tabs by aligning transparent overlays with color-coded elements (orange, yellow, blue, pink). This approach ensured that while each page retained its distinct feel, it still adhered to the overall SoundPalette theme. Furthermore, he iterated on gradient assets, experimenting with different variations to ensure seamless layering and a luminous visual effect. His meticulous work helped establish a refined and cohesive interface, improving the app's user experience and aesthetic appeal.

Kaija Sproxton:

Kaija is the Documentation Specialist for the team, taking full responsibility for writing all project reports, including the Proposal, Release Planning Document, and Project Progress 1 Report. Her meticulous approach ensures that the team's work is well-organized and professionally documented, keeping the project on track and aligned with its goals.

Beyond her documentation role, Kaija played a central part in shaping SoundPalette's visual identity, leading efforts in color-coded navigation and gradient-based design elements. She assigned signature hues to key app sections, such as Home, Profile, Messages, and Search, ensuring that each area maintained a distinct yet cohesive color scheme. One of her major implementations was animated gradients, which introduced smooth color transitions on major screens, reinforcing SoundPalette's playful, energetic, and artistic aesthetic.

To further enhance the app's visual appeal, Kaija designed a custom SoundPalette text animation that appears on the splash and primary screens. Using timed keyframe effects, she created a fade-and-scale transition, giving the brand name a dynamic and memorable entrance.

She also applied color transitions to the Login and Register pages, blending white backgrounds into soft pastel hues to create a welcoming user experience that flows seamlessly from first login to everyday interactions.

William White:

William served as the foundational developer for the SoundPalette project, taking charge of both front-end and back-end setup. He began by creating the base Android project, including the initial file structure and a simple login page, providing a strong starting point for the development process. On the backend, he built the basic Kestrel API server, establishing the framework needed to connect the front-end and back-end systems seamlessly. This took a very long time, and he spent days trying to configure the systems so they would work seamlessly.

He also created both of the Git repositories for the UI and API server, ensuring effective version control throughout the project. He also implemented Retrofit, an HTTP client library, to enable smooth communication between the UI and API server. To further streamline the system, William incorporated Microsoft Entity Framework to connect the API server to the database, providing the necessary tools to handle data efficiently.

William created the initial version of the database, including the necessary scripts to support user creation, profiles, and posts. He then added essential services to the API server, allowing for user registration, profile creation, and login. Moreover, he integrated functionality for users to add posts to the database and retrieve them, setting up both text post creation and post retrieval systems via the API server. He ensured that users could create text posts within the UI, providing a seamless front-end experience.

James Windjack:

James has played a crucial role in both the conceptualization and technical implementation of the SoundPalette app, contributing to a range of tasks from ideation to problem-solving. He was the one who initially proposed the idea for the app during the first group meeting, offering a solution for artists in the music industry to collaborate more effectively on their works. James brought personal experience and insights into the challenges of current social media platforms, laying the foundation for the app's purpose. He followed this by helping to define user stories and add significantly to the design process.

James also set up the development environment for the team’s Mac users, dedicating significant time (days) to researching and troubleshooting configurations to ensure smooth operation for all team members. His research and troubleshooting proved crucial for the team to have a coherent setup. His perseverance and technical problem-solving led to a working setup on both Intel and Apple Silicon Macs, which he documented and shared with the team. He provided

hands-on assistance, helping team members troubleshoot and get their local development environments running.

In addition to his work on the environment setup, James contributed actively to the UI and database design meetings. He helped shape the UI flow and contributed to database design, working with the team to ensure data was represented correctly within the app. He took the initiative to create an initial UI mock-up for the main screen, which served as a framework for the team to build on. James also spent considerable time (days) learning C# and RetroFit, allowing him to bridge the gap between the Android app and the API server. He worked extensively to get the two systems to communicate, ensuring smooth data retrieval and updates between the app and database.

James made significant contributions to the Android app's user interface, implementing key features such as the user registration screen and profile picture upload functionality. He tackled a series of technical issues, such as resolving broken API connections after updates, fixing issues with retrieving location data from the API server, and ensuring user profile information was correctly saved and updated in the database.

Team Contribution (New) In-depth:

Parth Chauhan:

Parth deepened the project's test coverage by developing additional unit tests using JUnit and Mockito, targeting key areas such as input validation, authentication, and user session handling. He also created a Test Case Table as seen under the Testing header in this report, organizing and documenting test scenarios for easier reference and consistent quality assurance. This table serves as a valuable resource for the team, ensuring test coverage aligns with the app's functional requirements. It is not fully complete and is still being added to.

Victoria Dahn:

Victoria has continued to actively fulfill her responsibilities as Scrum Master, ensuring consistent team communication and workflow management. She has remained diligent in organizing weekly team meetings, addressing concerns raised by team members, and facilitating productive discussions. Toward the conclusion of Sprint 3 and the start of Sprint 4, Victoria led the team in reassessing the project timeline, leading to a collective decision to remove the Events and Monetization features in order to maintain a realistic and achievable scope. She promptly updated the team's roadmap to reflect these changes and ensured that all adjustments were communicated clearly across the team and to stakeholders. Victoria curated focused demos by preparing PowerPoint decks. She also reflected on completed and pending User Stories to facilitate alignment between sprint goals and stakeholder expectations. Meeting summaries continue to be carefully documented and uploaded to the team's GitHub repository for transparency and future reference.

On the development side, Victoria has been instrumental in advancing the front-end features of the SoundPalette Android app. In Sprints 4 and 5, her focus shifted toward enhancing user engagement through the implementation of interactive features such as likes, comments, saved posts, and editable user content. She designed the basic layout and logic for the comments section, constructing a RecyclerView and corresponding CommentAdapter to dynamically manage and display user comments. Victoria also implemented core logic to iterate through comment lists and bind them to the UI.

Victoria is currently working with Will to implement the Direct Messaging system. Her primary focus has been building out the initial UI for the messaging interface, with plans for further refinement by Prab and Kaija. She continues to anticipate team needs by structuring her components to support efficient integration. She created a comprehensive script to populate the database with data ranging from user accounts to posts, comments, likes, and saved content. This enabled the team to test features against a robust and realistic dataset, supporting Parth's early rounds of functionality testing and bug tracking.

Prab Khokhar:

Prab and Kaija were both impacted by a severe flu that sidelined them for nearly two weeks, causing a temporary delay in their front-end development progress. Despite this setback, Prab has resumed work and made notable contributions toward refining and modernizing the UI. He continued enhancing the app's visual style by implementing rounded corner designs across various boxes and containers, further reinforcing SoundPalette's clean and minimal aesthetic. He also focused on improving key UI elements, specifically making enhancements to the edit tag and edit profile pages to ensure a more intuitive and visually cohesive user experience. In collaboration with Kaija, Prab began planning the integration of a new glass morphism design system, discussing layout adjustments, translucency layering, and component hierarchy to achieve a futuristic, polished look throughout the app. Although recent illness created some delays, Prab remains actively involved in elevating the UI's visual identity and ensuring that new changes align with the team's overall design direction.

Currently, Prab is developing a new animated visual asset, a stylized music score animation that will appear in the background of various app pages. This feature aims to add a playful and creative touch to SoundPalette, reinforcing its music-focused identity while subtly enhancing the visual experience without distracting from core functionality. The animation is designed to be lightweight and seamlessly integrated into the existing layout, offering movement and charm that supports the app's modern and expressive aesthetic.

Kaija Sproxton:

Kaija experienced a setback after catching a severe flu alongside Prab, which left her unable to contribute for nearly two weeks and temporarily delayed her progress on UI-related tasks. Despite this, she has resumed work and is actively catching up, maintaining strong communication with the team throughout to relay any concerns or issues as they've come up.

Kaija has continued to lead UI planning efforts in close collaboration with Prab, focusing specifically on how to implement a playful and creative glass morphism design that aligns with SoundPalette's artistic and modern aesthetic. Together, they have been discussing component layering, light opacity gradients, and animated transitions that bring the new design concept to life without compromising functionality. As the team's Documentation Specialist, Kaija also took full responsibility for writing this current project report independently, ensuring that all recent developments, challenges, and contributions were thoroughly documented. Her commitment to organization, creativity, and open communication continues to play a central role in keeping the team aligned and the project on track.

William White:

William has significantly expanded the functionality of the SoundPalette app by developing and integrating a wide range of backend services. He implemented essential API endpoints to support user interaction with posts, including commenting, liking, and saving. These services allow users to engage with content dynamically and were a key part of enhancing user experience. In addition, he created backend support for following and unfollowing users, enabling social connectivity within the app. William also developed services for retrieving various sets of posts such as posts by the current user, posts by specific usernames, and saved posts providing the flexibility needed to display tailored content throughout the app. All of these services were built on the existing Kestrel API server and were carefully structured to integrate smoothly with the existing database and Entity Framework setup.

On the front-end, William updated the UI components to connect with the new backend services, replacing static or placeholder elements with live data. For instance, the previously implemented buttons for post interactions were updated to reflect real-time status changes, such as toggling likes or displaying live comment threads. These changes helped transform the app from a static interface to an interactive experience. He also added backend functionality for viewing other users profiles and laid the foundation for the direct and group messaging system by implementing all necessary API services to support real-time communication. Additionally, William managed the complex task of merging the team's various code changes, which required several hours of coordination and debugging to ensure a stable, unified build. His comprehensive backend work and seamless front-end integration have been critical in moving the project toward a fully functional and user-responsive application.

James Windjack:

James has taken on an increasingly critical role in refining and expanding the user profile system, troubleshooting major bugs, and ensuring overall system stability. One of his primary focuses was developing and debugging the update user profile functionality. He began by refactoring the ProfileEditFragment to improve modularity, code readability, and long-term maintainability. When faced with issues related to the spinner not displaying location options, James traced the problem to missing or misformatted location data in the database. He modified the relevant table structures, fixed the API server so it could correctly return location IDs, and verified that the updated location list rendered properly in the UI. He went on to create the necessary procedures in the API server to read from and write to the user profile table and added models and endpoints on the UI side to handle profile data updates. Throughout this process, James performed manual testing, identified inconsistencies, and adjusted logic across both the server and client sides to ensure reliable data flow.

In the midst of integrating these changes, James encountered several merge conflicts while pushing updates to Git. He resolved the conflicts carefully and rebuilt the API server, which temporarily broke during the merge. After identifying and fixing the root causes, he committed his changes and created a safe working branch to prevent further instability. However, issues still arose with the updated data not displaying correctly in the UI, despite being written to the database. He spent hours identifying and fixing the asynchronous update issues and made sure data from the API server correctly appeared in the app with a single save, instead of requiring multiple actions.

Alongside profile editing, James led the full development of the user tag feature, which allows users to select, edit, and save personalized tags on their profile. He created the layouts, UI adapters, and a new API endpoint to handle the tag logic. He also built the API client on the UI side to communicate with the API server and wrote the necessary backend procedures for adding, updating, and retrieving tags. During testing, he discovered that crashes would occur after user registration due to missing bio data, which led him to add placeholder values during registration to prevent null-related backend crashes. He continued to iterate on the tag editing interface, creating additional UI fragments for tag editing, integrating them with the main profile screen, and ensuring the buttons for tag editing correctly navigated to their respective screens. To prevent repeated crashes related to null bio entries and invalid column references, James modified the database schema by adding the required columns and setting sensible default values.

In terms of stability and navigation, James discovered that navigating between fragments caused instability and crashes due to the FragmentManager call stack accumulating duplicate entries. To fix this, he rewrote the app’s navigation logic, implementing a tagging system that tracked fragment states, prevented duplicates, and allowed for smoother transitions. He also identified a crash when posting from any fragment other than the home screen and addressed it by modifying the post-navigation logic to correctly return the user to their last active screen.

Additionally, he made layout improvements, such as enabling automatic horizontal scrolling of user tags in the profile view, enhancing usability.

Beyond development features, James took the initiative to address a major compatibility issue where Mac users were unable to run the app after a recent pull. He dedicated an entire day to manual testing and log analysis, eventually identifying a mismatch in the Android API levels as the source of the problem. To resolve this, James built a working virtual machine, exported it, and created a step-by-step guide to help Mac users configure their environments correctly by importing the VM and matching the correct OS and API versions. Finally, he began exploring cloud hosting solutions for the app's backend infrastructure and concluded that Amazon S3 would be the most appropriate choice for hosting the API server and database in future production stages.

GitHub Logs:

I have provided the URLs to the Commits of each GitHub Repository so they can be viewed as there are quite a bit. Kaija and Prab have adopted a pair programming approach, working side by side at the same workstation to collaboratively write code, brainstorm ideas, and build out the UI together. Since they are in close proximity to each other and are both focused on front-end development, they found this method to be the most efficient resulting in all commits being pushed from Prab's computer. Parth does not have too many commits as he has focused on writing test cases and the majority of the testing.

Main Repository:

https://github.com/vd19qe/SoundPalette/commits/main/

UI Repository:

https://github.com/WilliamSEWhite/SoundPaletteUI/commits/master/

API Server Repository:

https://github.com/WilliamSEWhite/SoundPaletteApiServer/commits/master/