SoundCloud for VS Code

Roniel Abreu, Owen Hunter, Michael McGilvray, Alina Ryan

April 14tht, 2021

1 Background and Overview

SoundCloud for VS Code is an extension for the VS Code editor that integrates SoundCloud playback features directly into the editor. SoundCloud [1] is a lesser-known music streaming platform compared to popular services like Spotify [2] and Apple Music [3]. They offer a place for small artists to share their creations and in 2020, had over double the total available tracks compared to Spotify [6]. Despite this, Spotify and Apple Music still dominate in ease of use, the number of mainstream artists, and the versatility of the platform.

The goal of SoundCloud for VS Code is to make SoundCloud a more versatile platform by increasing its usability and removing the need to switch between a music player and VS Code. The extension enables higher levels of productivity and reduces the number of distractions while using VS Code. SoundCloud offers a similar experience to any other music streaming platform but adds functionality from social media sites. A user can comment on songs, repost songs to their feed, and display their favorite music on their profile. These features are great for regular SoundCloud users, but casual users can get confused by the cluttered interface. SoundCloud for VS Code offers a user-friendly experience that limits distractions by showing the core playback controls and hiding the SoundCloud social features. The SoundCloud for VS Code team aims to focus on including the essentials of a music player first and foremost but also to not limit the unique experience presented by SoundCloud.

2 Importance and Alternatives

In software development, productivity is very difficult to manage with the endless number of distractions at one's fingertips. Whether it be checking social media, responding to an email, or a quick game of solitaire, distractions limit the capabilities of developers in their rush against the clock. To limit these distractions, developers often find that productivity increases while listening to music. Rather than using VS Code for development and a music streaming app or website separately, SoundCloud for VS Code combines the two by integrating music playback control directly into VS Code. Other music streaming services like Spotify and Apple Music have been integrated into VS Code, but the less mainstream SoundCloud has not.

SoundCloud has limited accessibility and officially supports use through a browser, mobile app, and a Microsoft Store app. There are other third-party options for users on macOS and Linux, but none target software developers. SoundCloud for VS Code allows users of VS

Code the ability to stream SoundCloud music directly within VS Code and avoid the hassle of switching between their code editor and music player.

As mentioned before, there are similar extensions in VS Code that support popular platforms like Spotify, but there are currently none that support SoundCloud. Music Time for Spotify is a VS Code extension that integrates Spotify into VS Code and allows users to control music playback, manage playlists, view music recommendations, and analyze listening statistics through the VS Code interface. SoundCloud for VS Code aims to provide a similar but more streamlined experience. VSCode-Spotify is another extension that merges Spotify with VS Code and unlike the previously mentioned Music Time for Spotify, it provides a minimalistic interface that doesn't interfere with productivity. Its functionality is limited to mainly music playback controls and is the perfect option for Spotify users. SoundCloud users are left without an extension and forced to abandon their favorite artists only streamable on SoundCloud. SoundCloud for VS Code fills this void and is an extension that enables easy music playback control and productivity.
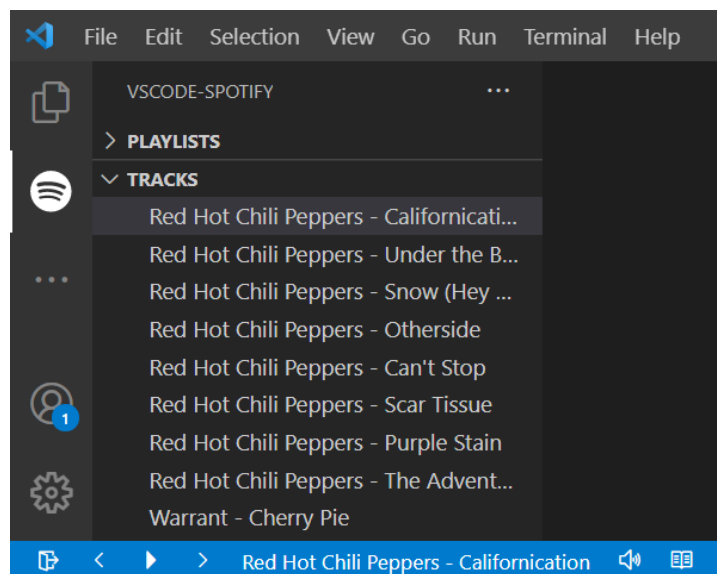

Figure 1: VSCode-Spotify interface

3   Approach and Limitations

SoundCloud for VS Code relies heavily upon using the SoundCloud API. The SoundCloud developer API uses OAuth 2.0, a standard authentication method with many API providers. This allows the extension to access additional features of SoundCloud such as playlists, likes, and other personalized data. Unfortunately, SoundCloud is no longer taking requests for developers to register their app with SoundCloud, preventing us from being able to use the SoundCloud developer API. A workaround we have decided is to create a SoundCloud account and use the account's associated API key instead of the developer API we could get from registering the app. With this method, our users will be able to input the client ID

associated with their personal SoundCloud account to be able to query and download tracks. Not having access to the developer API key greatly limits the available SoundCloud features such as OAuth support and the SoundCloud social features, but by using a single account's associated API key, users will at least be able to query and play songs from their favorite artists inside of VS Code instead of having to switch to another window.

Alongside the SoundCloud API, SoundCloud for VS Code also heavily utilizes the VS Code extension API. The extension API uses Electron JS, an API framework that merges Chromium and Node.js. This means that the extension can be written in either JavaScript or TypeScript (a superset of JavaScript). Since TypeScript can transcompile directly down to JavaScript, we decided it would be easiest to just use TypeScript. Due to the nature of working with the VS Code extension API, our design and features are further limited to what the extension API allows such as window, setting, button, and networking constraints.

Combining the SoundCloud and VS Code API, SoundCloud for VS Code incorporates features that make it function like a traditional music player. At its core, it includes controls to play and pause music, as well as skip the current track to the next in the queue or skip to the previous song in the queue. Tracks can be searched for by name and the currently chosen song will be displayed along with the name of the artist. Upon a track's completion, the next track in the playlist will automatically start playing or recommend a track when the playlist is empty. As previously explained, SoundCloud's lack of OAuth support will hinder the development of other account-specific features resulting in a focus on a more streamlined approach to a music player.

4   Tools and technologies

The tools and technologies SoundCloud for VS Code uses include our account's associated SoundCloud API, VS Code's extension API, Node.js, npm, Audic, TypeScript, and GitHub. SoundCloud for VS Code uses the associated SoundCloud API specifically to play and query songs from SoundCloud. Instead of using OAuth to access a user's account, SoundCloud for VS Code uses a single client ID from a SoundCloud account that we made, and in the future may have the option for users to input the client ID associated with their own SoundCloud account to query and download tracks. This may prove to be useful because in 2015 SoundCloud announced that there will be a limit to the number of streams per day outside of registered apps [8]. Despite the limitations, we think that being able to download, play and query songs will be enough to cover the use case. Someday, if SoundCloud begins accepting requests to register their apps we will have access to everything else that comes with the SoundCloud API including OAuth 2.0, embedding the SoundCloud widget in VS Code, and some social media features. We may limit the social media features available with the API to keep to the purpose of SoundCloud for VS Code - the ability to listen to SoundCloud without becoming distracted.

At the core of SoundCloud for VS Code's development we utilize the VS Code extension API, Electron JS. As mentioned in the previous section, Electron JS merges Chromium and

Node.js and allows for the creation of GUI applications using web technologies. For language, our group chose to write our extension in TypeScript.

Additionally, we make use of npm to acquire the necessary dependencies and run our extension before it is published in the VS Code Marketplace. If needed we will use npm for any other packages that may be useful in the further development of the extension. Specifically, we have used Audic, a Node.js module that allows for MP3 files to be played as our means of playing downloaded tracks. Finally, we have used a GitHub repository to contain our project and for source control. Our repository is open source and includes all of our implementation files, contribution guide, and instructions to run the project. See our current progress [here](#).

5   Interfaces

As mentioned previously, SoundCloud for VS Code works closely with the SoundCloud and VS Code extension API. For each of the main functionalities, a combination of API and function calls is made. Users can first find a song by searching in the query bar, which calls SoundCloud's track search method. Selecting a song returns that song's track ID, which can then be used to download the track so it can be played. Additionally, users can decide to queue a song, which adds that track ID to a list to be played later on. Once a song is playing, the user has access to a variety of controls such as play, pause, skip, and skip to the previous track.
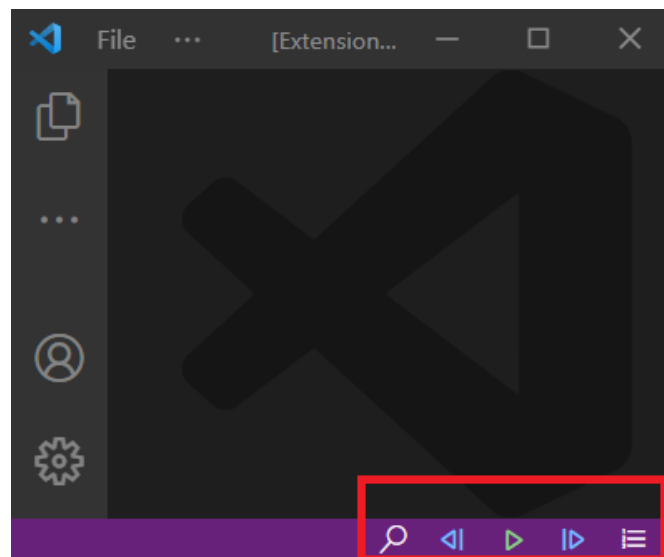

Figure 2: SoundCloud for VS Code startup
The area in red indicates the additional functionality that is presented by SoundCloud for VS Code. From left to right the buttons act as follows: song search, skip to the previous track, play/pause, skip to the next track, and view the songs in the queue.
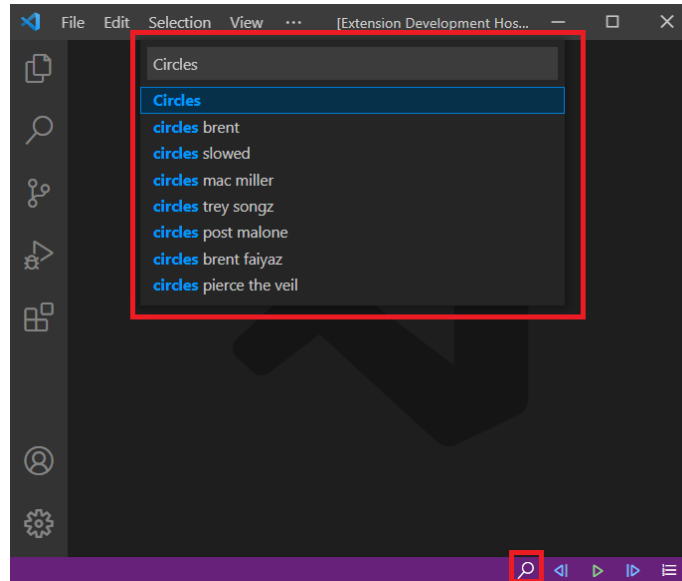
Figure 3: Song Search

The area in red indicates what should happen when the song search button is pressed. A dialog box is made available where the user can search for songs. During the search, the extension will attempt to autocomplete the desired track.
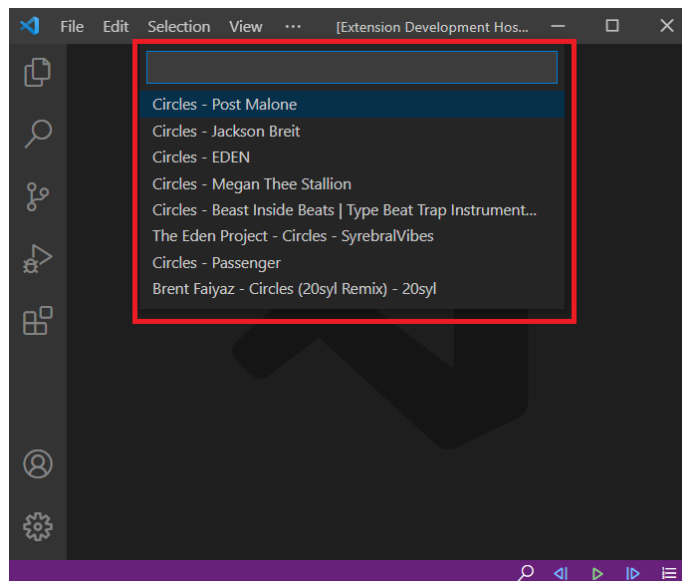

Figure 4: Song Selection

The area in red indicates the results after a song has been searched (in this case "Circles" was searched). The user can now select the desired track by clicking on it or entering it in the dialog box so it can get ready to be played.
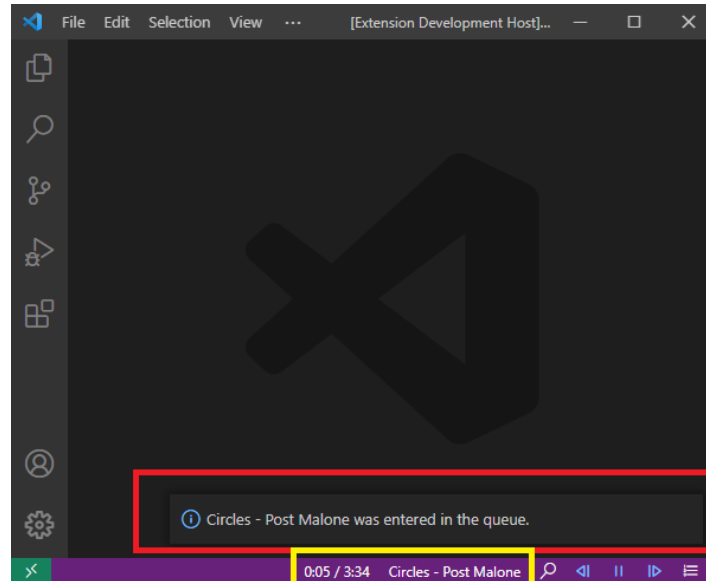
Figure 5: Song Playing

The area in red indicates a song being added to the queue after being searched and selected. The song then starts playing, the title and artist of the song, and the timer are highlighted by the area in yellow.

6   Architecture and Methodology

   To start using the extension, the user must open VS Code and enable the extension through the VS Code Marketplace by searching for "SoundCloud for VS Code." The user can then start playing music using the playback management buttons located in the lower right corner of the VS Code window. First, a song must be selected using the search functionality through a SoundCloud Request. This looks for the most relevant tracks related to the inputted search terms where the user can then select the desired track. Each track stores relevant information about its artists and how it can later be downloaded. The selected track is placed in the Track Player's queue and made the current track if there are no other tracks in the queue. Once at the front of the queue, the track can be downloaded via the SoundCloud Request and played. As it is being played the Timer keeps track of the time remaining on the current track and is stopped when the track is paused or has completed. Tracks can be switched by skipping to the next track or to the previous track if there is more than one track in the queue. The following diagram further illustrates these classes, the data they store, their methods, and how they interact.
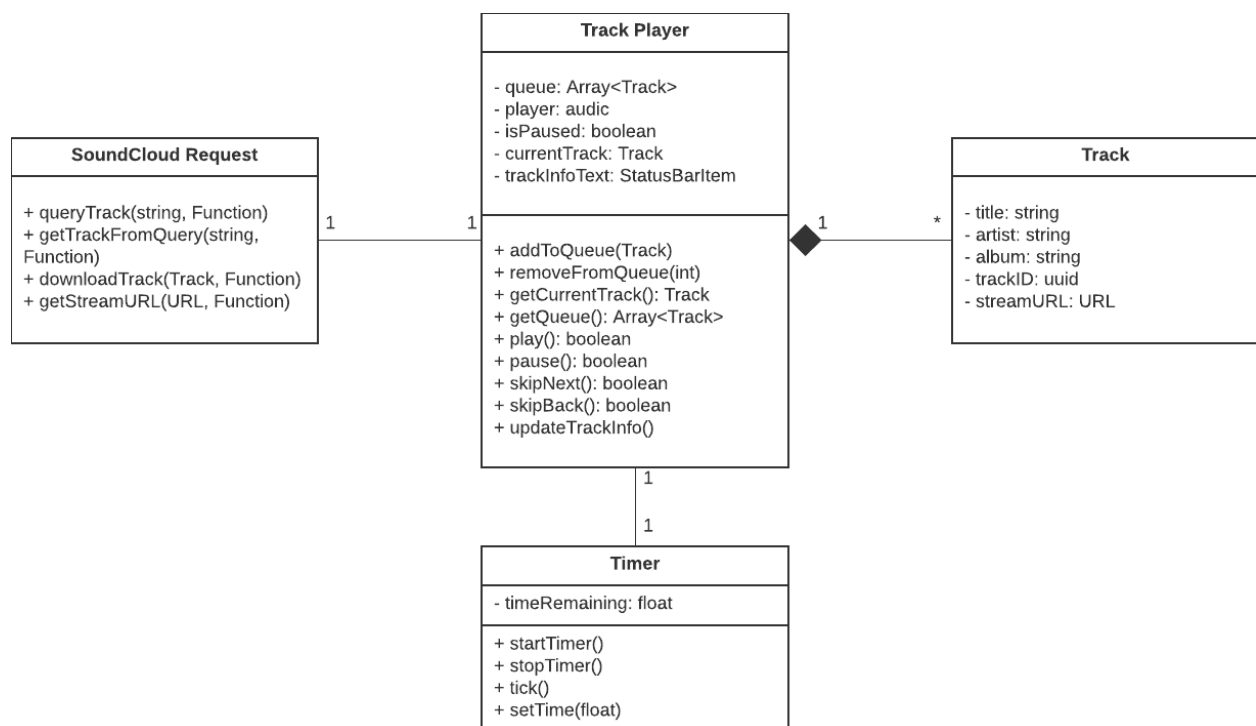

Figure 6: Class Diagram

SoundCloud for VS Code provides users the ability to play songs and control playback while coding to improve productivity. Below is the use case diagram that shows the major components of the extension and more specifically how a VS Code user and SoundCloud interact with it.
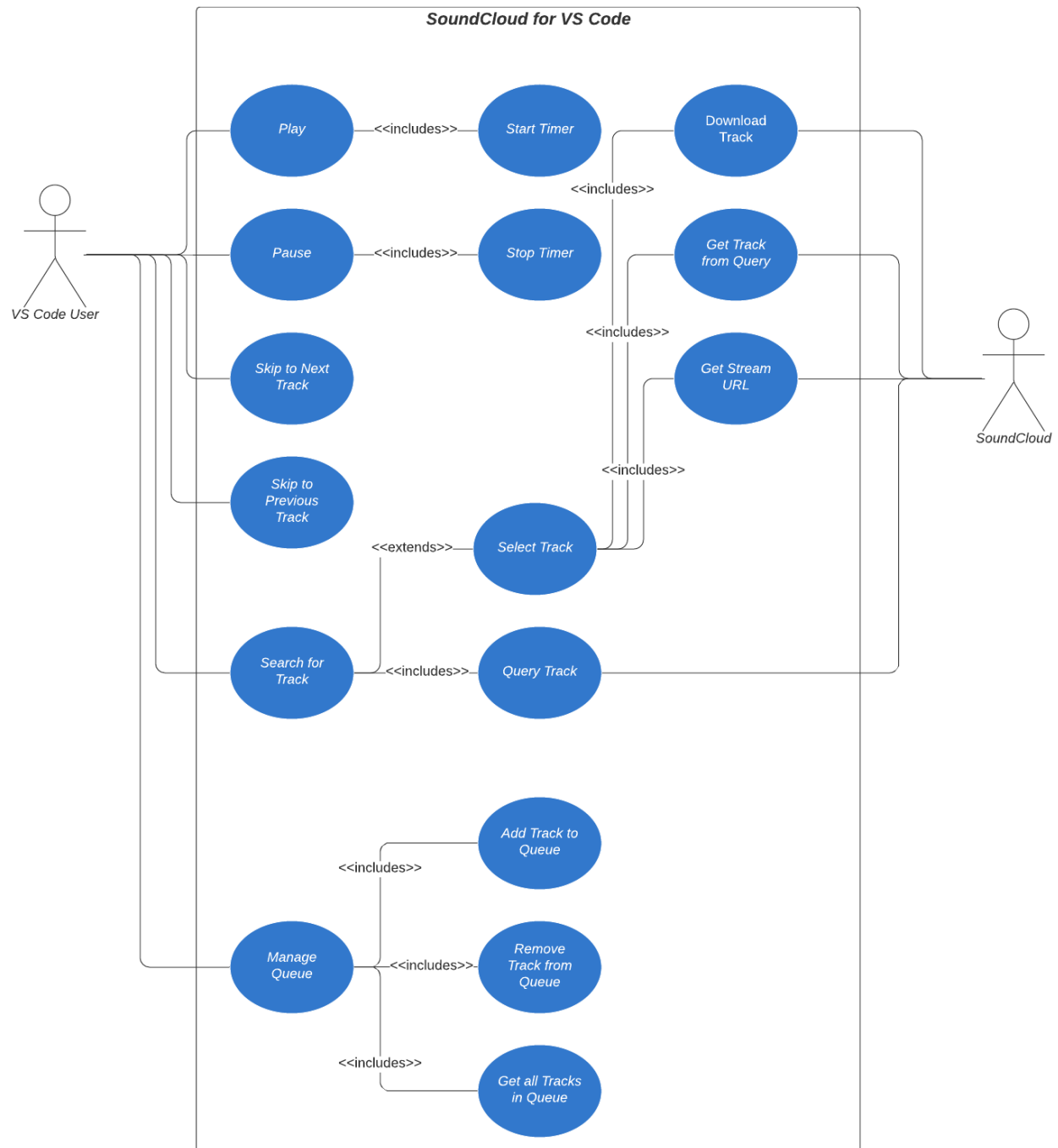

Figure 7: Use Case Diagram

7 Results

To determine the success of SoundCloud for VS Code we designed and ran several tests to indicate the completion and proper functionality of the different components. The tests can be found in our [GitHub repository](#) under the src/test/suite path. With the help of Travis CI, a continuous integration service that automatically builds and runs tests for projects on GitHub, after every commit all tests are used to build Figure 8. It shows all the tests that were run and their results from the most recent build on Travis CI and can easily be generated by following the instructions in the README of our GitHub. Figure 8 is generated by downloading the raw text file of the latest Travis CI build, extracting the test results, and organizing the data into a table. To generate the table, a link to the raw text file is needed so the user can use a recent build's results or follow the instructions displayed when running the script to get a link to the most recent build. This is useful as it shows the progress of SoundCloud for VS Code and highlights any tested features that may not be working properly.

Since SoundCloud for VS Code strongly relies on fetching tracks through an API that is no longer accepting application requests, it is important that we regularly test our code and have an automated and clear way of displaying the working functionality. If something were to break in the code, then it should be caught by one of our tests and could be easily reported in an updated version of Figure 8. We have tests for all of SoundCloud for VS Code's core functionality from the time a track is queried to when it starts or stops playing. During further development and maintenance of SoundCloud for VS Code, more tests will be added to our test file and will automatically populate in the table once committed and tested through Travis CI.

Test Name	Passed or Failed
timer_tick	Pass
timer_starting_with_0_seconds_remaining	Pass
track_creation	Pass
track_queue	Pass
query_track	Pass
get_track_from_query	Pass
play_with_sample_track	Pass
play_pause_with_no_track	Pass
new_status_bar_item	Pass
skip_to_next_track	Pass
skip_with_no_next_track	Pass

Figure 8: Automated test results

8 Challenges

The development process of SoundCloud for VS Code was a difficult one as a result one that required the team to coordinate and push through roadblocks. The most concerning challenge of the project was ensuring the viability of the workaround for obtaining a SoundCloud API key. Since SoundCloud is no longer accepting requests for developer API keys, the team

searched for different methods of connecting VS Code to SoundCloud. The most viable option allowed for basic song querying and playing, and the alternatives may have enabled account-specific functionality at the cost of an easy installation for the user. The focus was to implement the core features of the music player then attempt to incorporate the account-specific features if the alternative was reasonably straightforward for the user to install. The entire functionality is based on using the SoundCloud API and if the workarounds were to fail then the team would have been required to rethink the approach.
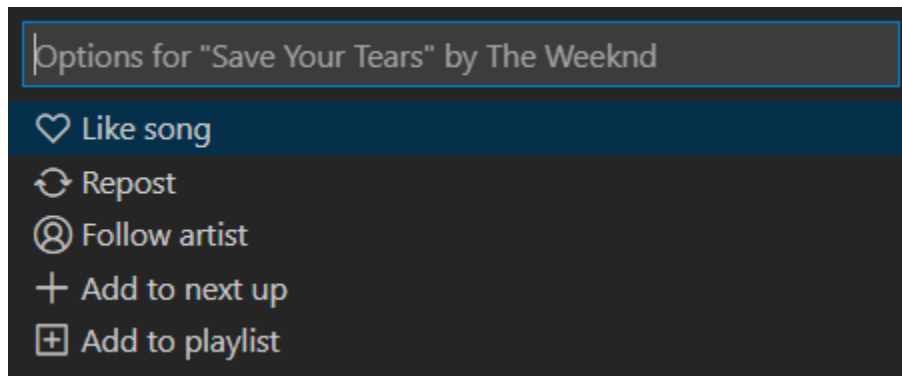

Figure 9: Potential account-specific features

Other than accessing the SoundCloud API, all team members were new to TypeScript, and developing the extension was slowed by the lack of experience. This required the team to coordinate in the design of the extension and the responsibilities that followed. Tests were regularly added that helped identify issues that were not initially apparent during the team's coordination.

Another major point of contention was the name SoundCloud for VS Code and its possibility to arise trademark infringement issues. None of the team has any background in trademark or copyright law, so the risk of using this name may be greater than expected. Using both the SoundCloud and VS Code trademark should be considered normative fair use as they only use the names to illustrate the extension's purpose, not to suggest sponsorship or create confusion of the source of the extension [7]. Through the VS Code Marketplace, many extensions use VS Code in their name, and similarly, through the Chrome Web Store, there are extensions that use SoundCloud in their name. SoundCloud for VS Code shouldn't be any different, as its goal is to provide additional functionality for SoundCloud and VS Code users through the APIs presented by the respective association.

The extension is a project for the team's Software Engineering II class, meaning the developers worked to gain experience and required no payment. The project required no other costs as it is free to use the VS Code and SoundCloud API. The final deadline of the project is on April 26th, 2021, and development will have taken the entire Spring 2021 semester starting from February 12th, 2021. The exact number of hours the project took to complete is difficult to estimate, but somewhere around six to seven hours per week per team member for the approximate ten weeks of development. If team members are to continue development after the

Spring 2021 semester, additional features will continue to follow the goal of SoundCloud for VS Code by enabling productivity and making SoundCloud more user-friendly. Additional functionality may entail access to playlists created in SoundCloud and minimal social media features offered by SoundCloud such as reposting and following artists.

9   Success

The project should be considered a success when at the very least, a VS Code extension with SoundCloud access through the SoundCloud API gives the user the ability to query and play songs without the user having to leave the VS Code. Additionally, the core playback features such as play, pause, skip, and back buttons should be functional. If the alternative approach to the OAuth system were viable, we would attempt to implement a few minimal social media features that differentiate SoundCloud from other music streaming services such as reposting and following artists. Our midterm exam to check for success was completed and due the week of March 22nd where we intended to have implemented the previously listed core playback features and have completed the first round of testing. The final exam to check for success is our final report. By this point, all the features have been implemented which include the track and connection between the SoundCloud account and extension, playlist search and selection, and autoplay to the next track in the queue. Additionally, final testing has been completed, and our presentation will soon be ready to showcase.

10   Schedule

Week	Milestones	Due Date
15-Feb	**Project Proposal due**	Group Deadline: Thurs Feb 18th **Hard deadline: Fri Feb 19th, 11:59 PM**
22-Feb	-Project approach is chosen (extension or plugin) -List of functionality we will enable is chosen -Tasks delegated for the next deadline	Group Deadline: Sunday, Feb 21st, 5:00 PM
1-Mar	**Architecture and Implementation Plan** -Proposal revised and complete -Schedule updated -Tasks delegated for the next deadline	Group Deadline: Tuesday, March 2nd, 11:59 PM **Hard deadline: Wed March 3rd, 11:59 PM**
8-Mar	**Repository Begin Implementation** --Connects to SoundCloud using the team	Group Deadline: Tuesday, March 9th, 11:59 PM

	SoundCloud account's client ID	**Hard deadline: Wed March 10th, 11:59 PM**
15-Mar	**Build and Test** -First round of testing is complete and recorded -The first draft of dev preview is recorded --Track play and pause buttons functional --Tracks can be chosen by providing a track ID --Track name and artist is displayed **Midterm Presentations**	Group Deadline: Tuesday, March 16th, 11:59 PM **Hard deadline: Wed March 17th, 11:59 PM**
22-Mar	-Schedule updated --Track queue that remembers previously played tracks and newly searched songs implemented --Skip to next track and skip to previous track buttons functional using the track queue	Group Deadline: Sunday, March 28th, 5:00 PM
29-Mar	**Initial Results due** -Add methodology section to doc -Add results figure to represent data analysis pipeline	Group Deadline: Tuesday, March 30th, 11:59 PM **Hard deadline: Wednesday, March 31st, 11:59 PM**
5-Apr	-Tasks delegated for drafting Final Report / Repo Review -Group discusses if we want to implement minimal social media features	Group Deadline: Sunday, April 4th, 5:00 PM
12-Apr	**Draft Final Report / Repository Review due** --Time remaining in the current track displayed --Track search can search based on the track name, artist, or album --Track search returns a list of relevant tracks rather than one track	Group Deadline: Tuesday, April 13th, 11:59 PM **Hard deadline: Wednesday, April 14th, 11:59 PM**
19-Apr	-Final round of testing is complete and recorded --Track queue auto-plays the next track in the queue --Reorder queue of tracks --Search for playlists	Group Deadline: Sunday, April 25th, 5:00 PM
26-Apr	**Final Report due(Final)**	Group Deadline: Tuesday, April 27th, 11:59 PM **Hard deadline: Wednesday, April 28th, 2:00 PM**

Note: The extension's features are distinguished by -- instead of - in the above schedule.

11 Feedback

We have addressed all feedback.

12 Bibliography

[1] "SoundCloud Music." *SoundCloud*, www.soundcloud.com/.

[2] "Listening Is Everything." *Spotify*, www.spotify.com/.

[3] "Apple Music." *Apple*, www.apple.com/apple-music/.

[4] "Play Music with Spotify in Visual Studio Code." *Software.com*, www.software.com/src/play-music-with-spotify-in-visual-studio-code.

[5] Balzer, Cass. "Music Can Boost Your Productivity While Working from Home – Here's How." *The Guardian*, 15 Apr. 2020, www.theguardian.com/us-news/2020/apr/15/music-productivity-working-from-home.

[6] Moncada, Sebastian. "Spotify vs Soundcloud (2020 Comparison)." *Bunny Studio*, 29 Oct. 2020, www.bunnystudio.com/blog/library/audio-ads/spotify-vs-soundcloud-comparison/.

[7] *Fair Use of Trademarks (Intended for a Non-Legal Audience)*, International Trademark Association, 16 Dec. 2020, www.inta.org/fact-sheets/fair-use-of-trademarks-intended-for-a-non-legal-audience/.

[8] Heins, Scott. "Party's Over? Soundcloud Announces Daily Streaming Limits For Embedded Tracks." *Okayplayer*, 18 June 2015, www.okayplayer.com/news/soundcloud-streaming-limits.html.

[9] Roniel Abreu, Owen Hunter, Michael McGilvray, Alina Ryan. "SoundCloud for VS Code." Apr. 2021, https://github.com/owen-hunter1/soundcloud-for-vs-code