SIP311
SUMMER SEMESTER
2025

SIP

Pitch

PROJECT PROPOSAL

Foobar Controller Mobile

(Name Pending)

Preston Chapman
Advancing Computer Science & Network Engineering

FOOBAR CONTROLLER MOBILE

PRBLEM STATEMENT

When listening to music locally, with either a server or just your computer, it can be difficult to control the playback remotely without the need for a complicated setup.

This project will solve these issues by creating a mobile app that will allow you to control an existing application called “Foobar2000” running on your computer or server, by using your mobile phone.

Around 90% of the population listen to music regularly according to Susic, resulting in 7.11 billion people, and around 10% of people listen to music locally with services such as Foobar2000, DeaDBeeF, and physical media such as CD and MP3 Players. Henceforth around 711 million people would be viable to use this mobile app.

PROJECT DESCRIPTION

This is a mobile app for the existing software Foobar2000, that will allow you to control your desktop music player, remotely using your phone, tablet, or other smart device. The goal is to provide a great user interface (UI) and user experience (UX), while still maintaining the abundant number of features that foobar2000 offers. The features would include Playlist, Library, Playback Queue, Album Art, Local Library, Lightweight, Customizable, and more. Although foobar2000 is a great freeware application, it is on the older side, but it still has updates to this date, it is also highly customizable with skins and plugins. The goal is to use an open-sourced plugin called ‘Beefweb” which allows for communication over HTTP to control Foobar2000 remotely. I will not be creating the music player (Foobar2000) or the API (Beefweb), I will be creating a mobile app that uses Beefweb to control Foobar, while having an easy-to-use UI.

INNOVATION CLAIM

When listening to music locally, with either a server or just your computer, it can be difficult to control the playback remotely without the need for a complicated setup.

USAGE SCENARIO

When the user has Foobar2000 installed on their computer, they will add the 'Beefweb' plugin to the application, afterwords they will install Foobar Controller Mobile onto their phone or mobile device. Once it is installed, they can search for the server, and they connect with it, this allows them to see the active playback, pause/play, queue songs, change volume, skip, and view their library, as well as additional features. This mobile app will be treated as a smart remote, that works directly with the music player, it will not output the audio itself, just controlling the computer that is playing the music remotely.

STUDENT ROLE

Preston Chapman – Main Role: Programmer
Preston is currently the only student that will be working on this project, their key role is the programmer, however they will also be working on the following roles:

- User Interface (UI)
- User Experience (UX)
- Frontend
- Backend [i]
- Assets [ii]

[i] – The API used is not created by Preston, however they are creating all the logic on the backend of the app itself, handing the responses of the API calls are creating classes to manage this information

[ii] – Currently all assets are from third parties, and are not created by Preston, however either another student, or Preston themselves will be creating assets closer to the end of the project. Any assets not created by Preston or a student will have proper documentation regarding the license of the assets/images. Documentation is provided on the GitHub README

EVALUATION CRITERIA

☐ – Can the user control playback of Foobar2000? (i.e., Pause, Play, Skip, Seek, Volume)
☐ – Is the application Open Sourced?
☐ – Does the application follow the licenses of all dependencies and third-party assets?
☐ – Can the user view their music library?
☐ – Can the user view their playlists?
☐ – Can the user change the playing song from their Library?
☐ – Is the app free to use?
☐ – Does the app **NOT** have ads?
☐ – Does the app work fully without an internet connection?
☐ – Can the user use the app without crashing or bugs?
☐ – Does the app have a good User Interface? (UI)
☐ – Does the app have a good User Experience? (UX)
☐ – Is the app customizable?

☐ – Does the app follow the methodologies of Foobar2000? (Freeware and advanced)
☐ – Is the app secure and follow best practices for network traffic?

INTENDED METHODOLOGY, DATA & MATERIAL

This application will be created using the TypeScript programing language, a superset to the JavaScript language. The framework I will be using will be React Native, with the Expo Go Toolchain that is built on top of React Native.

This application will be using the API Beefweb created by Hyperblast and contributors https://github.com/hyperblast/beefweb. This API is a plug-in that will be added to the users Foobar2000 instance (https://www.foobar2000.org)

In order to built the prototype you must have Node JS installed, and download either the latest commit, or latest release's source code (recommended) from the GitHub: https://github.com/pchapman-uat/Foobar-Controler-Mobile after doing so, run the command 'npm install' in the directory and it will install all the dependencies. Currently there is only support for the Android Operating System, which means it cannot be built to an Apple (iOS) device. Make sure you follow the instructions from Android for using Android Debug Bridge (ADB) or an emulator to use the prototype of the application.
https://developer.android.com/tools/adb

To run the prebuilt release, you must navigate to the GitHub Releases https://github.com/pchapman-uat/Foobar-Controler-Mobile/releases and then download the provided APK. This will allow you to install the application on your android device, as mentioned above iOS is not supported and cannot be provided.

There are a multitude of dependencies used on this project, most of which are minor, but a full list is provided here:

- @react-native-async-storage/async-storage: ^2.1.2,
    - Storage of settings
- @react-native-community/slider: ^4.5.6,
    - UI Element
- @react-native-picker/picker: ^2.11.0,
    - UI Element
- @react-navigation/native: ^7.1.9,
    - Navigation
- @react-navigation/native-stack: ^7.3.13,
    - Navigation
- axios: ^1.9.0,
    - Web Request to the Beefweb API
- expo: ~53.0.9,
    - Framework

- expo-dev-client: ^5.1.8
    - Dev Backend
- expo-screen-orientation: ^8.1.6,
    - Horizontal and Vertical views
- lottie-react-native: ^7.2.2,
    - Animated Icons
- react: 19.0.0,
    - Framework
- react-native: 0.79.2,
    - Framework
- react-native-audio-pro: ^9.9.1,
    - Audio Controller in notification bar/lock screen
- react-native-elements: ^3.4.3,
    - UI Elements
- react-native-gesture-handler: ^2.27.1,
    - Required for color picker
- react-native-network-info: ^5.2.1,
    - Finding available servers
- react-native-safe-area-context: ^5.4.0,
    - UI Elements
- react-native-svg: ^15.12.0,
    - SVG Support
- react-native-svg-transformer: ^1.5.1,
    - SVG Support
- reanimated-color-picker: ^4.1.0
    - Color Picker

Please view the package.json or the README.md for up to date information.

This project will be using Trello to keep track of progress and tasks that need to be completed.
https://trello.com/b/WqcaJ2Lh/foobar-controller-mobile

PROJECT OBJECTIVES

1. **Create the base form of the mobile app** (Marked as ‘core’ on Trello)
    - This step is creating the base features of the app, this would allow the app to be functional, but it will not look pretty or be in a fully usable state. This step may happen in conjunction with the following two steps, depending on what feature is being worked on.
2. **Add optional features and improve existing features**
    - Once all the core features have been created, this step will improve the existing ones, as well as adding quality of life and optional features, some of these features may include on-hold actions, themes, settings, preferences, and others. Existing features will be improved and expanded upon during this step
3. **Improve UI/UX** (Marked ‘Frontend’ on Trello)

- During this step the User Interface and User experience will be worked on, drastically improving how the app will look and be used, the backend code will not be changed much, and the functionality will remain mostly the same. This step can happen in conjunction to the following step.

4. **Add 'Personality'**
    - The app although will have a good UI/UX at this point, this step will entail changing the UI to be something unique and more stylized compared to other applications. Features such as themes will be worked on during this step, as well as work on assets, logos, and art for the application.
5. **Create proper documentation**
    - Although there will be documentation on the progress of the app, this step will have the usage of the app, such as set up, troubleshooting, and developer documentation. This would be provided in a Wiki like style using GitHub Wiki option, or a similar service.
6. **Polish for release**
    - This is the final stage of the application, where the application will be adjusted and cleaned up to be ready for release. Depending on the time and state of the project upon beginning this step, there is a possibility that the application will be published on the Google Play store, however this is still uncertainty. If it is not ready for the store, it will still be completed in such a way an Android Package Kit (APK) will be available on GitHub.

Detailed information about each task is provided on the Trello Page:
https://trello.com/b/WqcaJ2Lh/foobar-controller-mobile

EVALUATION PLAN

When the project has been seen as complete, Preston will provide the application to other people and have them assess it, Preston will also use the app daily for around a week for personal use. This will allow him to understand if there are any issues for active use. All tasks within the Trello, as well as the tasks within this document, should be completed.

PRIOR ART

BUBBLE UPNP

This service uses UPnP to control Foobar2000. A benefit is that UPnP allows for audio mirroring. However, UPnP is often not secure. BubbleUPnP is missing important features like a playback queue, reliable album art (results varied), and a customizable UI.
https://bubblesoftapps.com/bubbleupnpserver2

MONKEY MOTE

MonkeyMote is a paid application for $0.99. It has most of the features that BubbleUPnP is missing, however it requires a proprietary API that is closed source and not available for other applications.

https://www.monkeymote.com/home

APPENDICES

Note: This section must completed prior to PRO483.

Include as appendices any supporting material for this project, including charts, graphs, and other data; images associated with the project; or other documentation (e.g., a game design document or read-me file). Include any prior art that was used such as U.S. Patent Documents, Foreign Patent Documents, or other sources. Remember that this section should only be a list of additional files, not the actual data of the files!

Appendix A: GitHub Repository: https://github.com/pchapman-uat/Foobar-Controler-Mobile
Appendix B: Trello Board: https://trello.com/b/WqcaJ2Lh/foobar-controller-mobile
Appendix C: Foobar2000: https://www.foobar2000.org
Appendix D: Beefweb Repository: https://github.com/hyperblast/beefweb
Appendix E: Wiki: https://github.com/pchapman-uat/Foobar-Controler-Mobile/wiki