Date: 6/23/18
To: Daniel Ek
CEO
Spotify Technology SA
From: Nikhil Shirolkar
Undergraduate Student at Virginia Polytechnic Institute and State University
Subject: **Proposal to develop collaborative music streaming application for Spotify**

Purpose

The intent and purpose of this proposal is to offer my services in developing a collaborative music streaming tool that can be easily integrated into or added to Spotify's current application. The tentative name for this product is *Spotify Share*.

Summary

Imagine the scenario where two Spotify users are listening to music together but then one of them has to leave to go to work. The collaborative music listening experience doesn't have to stop for them if Spotify has *Spotify Share. Spotify Share* is a collaborative music streaming product that can be seamlessly added as a feature to Spotify's main application. *Spotify Share* allows for users to remotely listen to their favorite songs together at the exact same time simply by connecting to the same playlist, or room. *Spotify Share* can revolutionize the music streaming industry by allowing people to connect with one another and remotely listen to music together with their friends and loved ones. This product is the perfect way to connect people using Spotify and thus to increase Spotify's user base, popularity, market cap, and of course, profits. Incorporating this product as a "Premium only" feature will incentivize more people to switch to premium in order to get the benefits of this applications which in turn will result in tremendous profits for Spotify. In fact, factoring in the cost and revenue estimated to be generated, only a 0.05% increase in Spotify premium users facilitated by this product will result in a profit of 40 million USD. A 5% increase would result in a 400 million USD profit. Utilizing Spotify's huge customer base and ability for users to interact with others through followers and following, *Spotify Share* can do extremely well due to its utility and unique concept which will in turn increase premium users, popularizing Spotify further, and cementing Spotify's place as the market leader and technological visionary of this market.

Introduction

Spotify has been very successful since its inception in 2008 due to its ability to retain utility over music piracy while still rewarding the music artists and content providers. In fact, Spotify has been so successful that it has managed to outperform all its competitors by a wide margin making it the number one music streaming provider in the world with the largest user base and market cap (Turner & Shaw, 2017). This application can capitalize on Spotify's user base. I propose to make an application that allows users to connect to a "room" where all users in that particular room stream and share the same music simultaneously. This application will successfully allow users to listen to music together when not in the same physical environment. This is a feature that currently has not been addressed by Spotify or any other major music streaming service. However, there are versions of this idea that have been implemented online, and through services such as Sonos. However, the online versions, such as the popular "watch2gether" and "rabbit," are very primitive and have poor GUIs, high levels of lag, and are targeted towards movie and video watching. "Sonos," a more sophisticated collaborative music streaming experience, on the other hand, only works on local networks, not over the internet, or the 3G/4G networks. The application I propose to create can fill this gap by creating a beautiful, flawless application that is easily maneuverable and allows for people to listen to music together anytime regardless of physical location. Given Spotify's already large user base and user network in place, this application can easily be integrated into Spotify's system, which can result in huge profits and increases in popularity/market size (Castillo, 2018).

Adding a product like this to Spotify's menu will make it even more appealing. This product is innovative as it solves a problem that people may not know exists: it allows users to listen to music together when not physically near. This is something that has been addressed by very few before, and implemented it in Spotify's framework will bring people together, increase the utility and need for Spotify, and thus is intended to increase Spotify user base and premium subscribers which results in increased profits.

Spotify Share can be implemented in a couple ways based on Spotify's preferences, but in the big picture, this application will be based on a system of sockets and server "rooms". Each socket, or user, connects to a unique 'room' which can host multiple users up to a preset limit, and then using angularJS (does not have a period) and node.js (does have a period), a dynamic system can take place where users interact with the room and one another, effectively playing music for everyone to hear.

Before discussing the specs and tasks of the application, I would like to address how this application can fit into Spotify's long term business strategy and how I can help accomplish Spotify's goals. Spotify's business strategy is most similar to that of Amazon. Spotify has identified a market niche and found a unique way to utilize that market. Spotify prioritized building its large user base and ecosystem at the cost of profit deficit. In fact as shown in Appendix A, Spotify is still currently operating at a profit deficit of around 440 million USD, although its gross margins are steadily increasing (currently it is at 21%) as shown in Figure 2. The strategy Spotify is employing is to become so dominant and powerful in the sector in which it operates in at the cost of the initial loss, before later on becoming profitable. This application fits into this vision perfectly, as it can be one of many things that takes Spotify from being in a deficit, to finally becoming profitable (Richter, 2018).

The cost for maintaining a feature such as this is high due to development and server maintenance, so there are not many feasible or competitive alternative applications currently on the market. For this reason and because Spotify, unlike most of its competitors, is big enough to implement this feature without those drawbacks, having this feature will help Spotify stand out even further than it already does in its sector. This will in turn generate tremendous revenue to cover the costs solely due to the sheer number of users which Spotify has. I will talk later about the multiple ways this product can be priced to users (Kafka, 2018).

Proposed Tasks and Application Overview

I propose breaking the development and implementation of *Spotify Share* into the following chronological categories:

Task 1: Application Development

This will be the biggest part of this project. The premise of this application is that it is a system where friends can listen to musics together over the internet. Friends should be able to log on to Spotify, and jump onto their friends "room" and listen to the same music as their friend(s). In this task, I will meet with Spotify's database and systems management teams in order to get a good feel of the current structures. I need access to the database in order to create this application. Specifically, Spotify currently has a system in place that this application should logically build upon. Spotify has a system of user profiles, with followers, and friends. You can follow your favorite artists, and friend your friends, that way you can see who is listening to what music. With this application, it makes sense for me to have access to user data, specifically to friend data, since invited friends should be the only people allowed into a user's rooms. This application should build off this system of friends so that users can add as many friends as they want to their “room” so that everyone can listen to music. This concept can be furthered and rooms can be distinguished between private to public rooms. Some celebrity figures or artists could host public rooms where all their followers or fans come to listen to the same music as them. This is another example of the social and marketing potential this product has.

Without revealing too much about application development plans, *Spotify Share* will be based on the concept of rooms as alluded to earlier. Each room is stored on a online server, and each user that connects to that room interacts with the server in a dynamic fashion. Every user connected to the room can dictate which songs are played and the order of songs plays. Users can even pause or play songs for all other users in the room. This process is *asynchronous*, or happening dynamically at the same time. A popular framework for asynchronous programming is angularjs and I will consider using this framework dependent on Spotify’s current structure and technologies (Britannica, 2015).

On the server side, there are a variety of technologies that I can use. Which technologies I use is largely dependent on the types of servers Spotify has currently. However, one option is the use of node.js. Node.js is a open source server side programming language that executes JavaScript. Additionally, node.js is very useful for dynamic and asynchronous programming, so it works well with using angularjs (Chandrayan, 2017; Rodrigues, 2010).

Beta Testing

After developing the *Spotify Share*, the next step is to initiate testing in a smaller environment with fewer users. A certain amount of users should be selected to be exposed to this new application, and any problems/errors that occur will be dealt with. Additionally, we will ask these users for their feedback, and review the feedback in order to try to find ways to make product better.

Pricing and other logistics

There are a couple more logistics that have to be addressed about this product. The biggest and most obvious is: how is this application going to make money? There are multiple ways it can be implemented.

1. This feature can be added to all users for free, and it can serve as another means of expanding the market cap, user base, and popularity of Spotify which will be helpful in generating profit in general.
2. This feature can be added as a “Premium only” feature at the same price. Spotify premium is the paid version of spotify, it is add free and has a variety of features that non-paid customers do not get. Currently, there are just over 71 million premium Spotify

users, and 160 million monthly active users. This feature can be a way of enticing more people to go premium while also reward those premium members.

3. The feature can be added as a “Premium only” feature at an increased price for all premium members. This gives a reason to drastically increase revenue generated by premium, but does come at the cost of potentially losing some members who cannot afford the updated price.
4. Offer this feature as a stand alone one time product that users pay once to be able to utilize.
5. Create another brand of users called “super premium” which is slightly more expensive than regular premium, but has access to this application and future applications that may make Spotify better but more expensive.

I recommend option number two because it will entice enough users to go premium to make itself profitable while also not having too many users clog up and waste money from the Spotify servers. Additionally, it will raise the overall value of spotify premium and Spotify. However, the final implementation decision should be made with more facts and figure that I do not have right now, as I do not know the details of Spotify’s current expenses and framework.

Finalization and release of application

After we finish the application and believe it is ready to go out to the public, we will release it accordingly.

Qualifications and Experience

I am an undergraduate at Virginia Polytechnic Institute and State University majoring in both Computer Science and Computational Modeling & Data Analytics (CMDA). Throughout my education, I have taken a variety of programming and math courses which have instilled in me the foundations of computer science. Throughout these courses, I have become very familiar with the programming languages of Java, python, JavaScript, and HTML; however, I have also been exposed to php, SQL, node.js, django, C, MATLAB, Octave, Swift, and others. Additionally, I have worked on a number of projects independently and for school. Most notably I created an application hosted on an online server using node.js for the popular games of *Ghost* and *Scrabble*. Additionally, I have created web and mobile applications and written AIs for the games of connect 4, othello, Scrabble, and tic tac toe. I have no related work experience as of yet, but I am interning at a company this summer as a computer science intern. I believe I am a qualified candidate because my education and projects I have worked on have given me a solid foundation in programming and an analytical mindset.

Budget and Expenses

There are a number of expenses that come with a project as such. The budget is as follows:

1. Development cost: For a project of this scope and magnitude, it should have a flexible budget of $50,000 to develop. This includes buying servers and other technologies needed to make this product.
2. Payment for the product: The preferred mode of payment is stock options up to $100,000 USD. This product will make Spotify more successful and raise its stock price, so compensating with options is beneficial.
3. Maintenance expenses: Some expenses of this project will stick around after the application has been completed, namely the server maintenance. Although the exact cost of these servers varies based on Spotify's current situation and how many users are expected to be processed, it is roughly estimated that maintenance will cost around around $1,568,628 USD yearly, but this will be able to handle and process the whole of spotify’s user base. (Arthur, 2009)

Expected revenue will vary based on what pricing approach is used. Given that spotify premium charges $9.99 per month, if option number two from above is used and premium membership increases by 5%, then around $425,574,000 yearly will be generated. The profit in this scenario is around 400 million including other expenses such as processing fees. This is a huge profit for Spotify. Refer to Table 1 which shows the estimated profit based on different levels of premium increase.

Appendix A

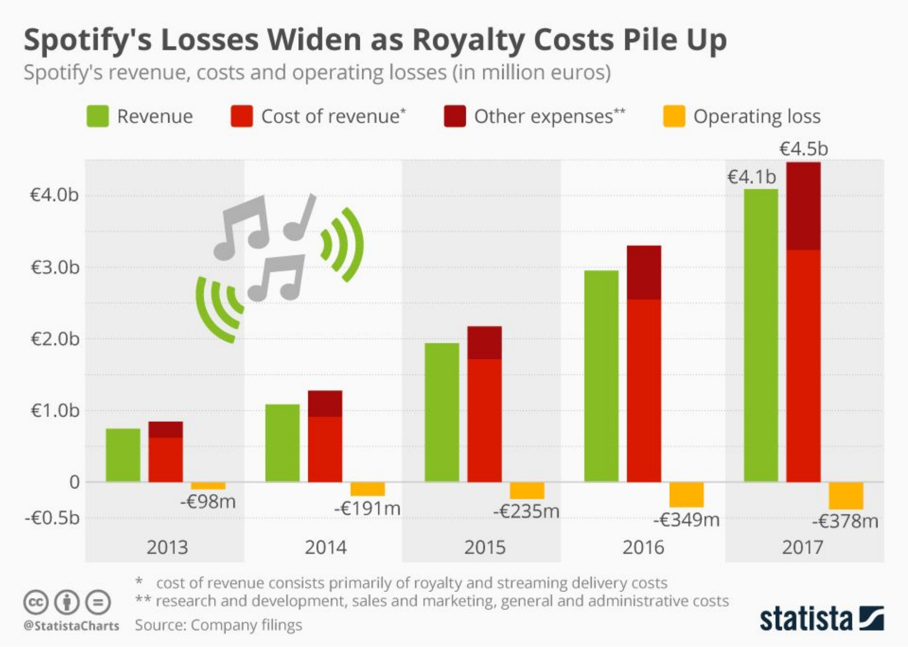

Figure 1: Spotify Year Revenue, Cost, and Loss (Source from Richter, 2018)

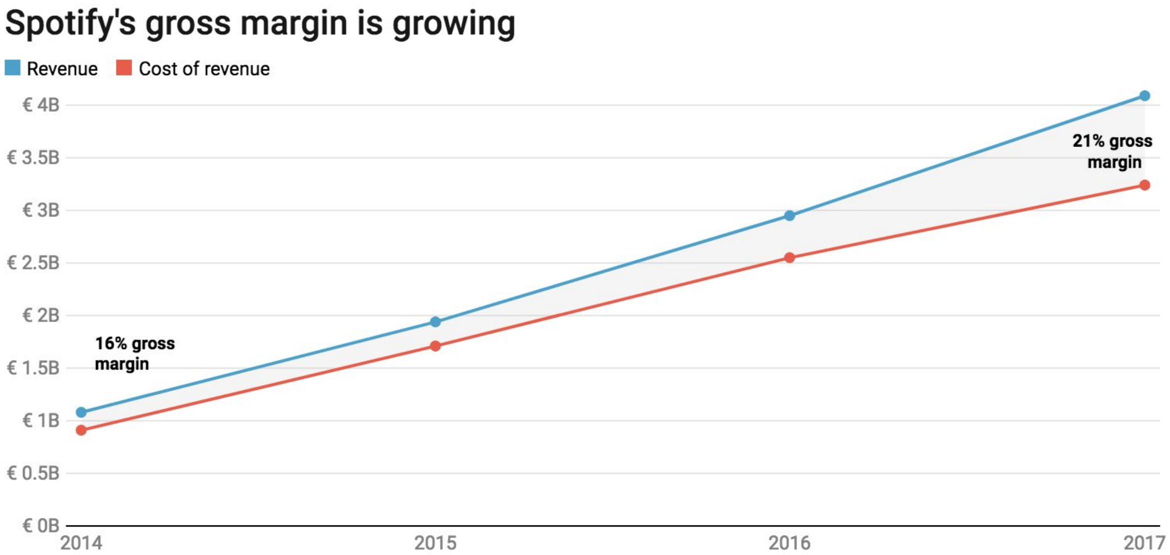

Figure 2: Spotify Gross Margins (Source from Molla, 2018)

% Increase in Premium Subscribers	Rough Profit (millions)
.5	40
1	85

5	400
10	850

Table 1: Spotify Estimated Profit based on % increase in Premium Subscribers

References

Arthur, C. (2009, October 08). How much does Spotify cost to run? We analyse the numbers. Retrieved from https://www.theguardian.com/technology/blog/2009/oct/08/spotify-internet

Beal, V. (n.d.). Client-server architecture. Retrieved from https://www.webopedia.com/TERM/C/client_server_architecture.html

Britannica, T. E. (2015, November 23). Client-server architecture. Retrieved from https://www.britannica.com/technology/client-server-architecture

Castillo, M. (2018, February 28). As Spotify prepares to go public, there's no obvious solution to its shaky business model. Retrieved from https://www.cnbc.com/2018/02/28/spotify-ipo-business-model-flaws.html

Chandrayan, P. (2017, November 05). Getting Started With Node.js : A Beginners Guide – codeburst. Retrieved from https://codeburst.io/getting-started-with-node-js-a-beginners-guide-b03e25bca71b

Kafka, P., & Molla, R. (2018, February 28). Spotify is still burning an enormous amount of cash, but the bigger it gets, the better it looks. Retrieved from https://www.recode.net/2018/2/28/17063892/spotify-ipo-margins-music-labels-streaming

Richter, F. (2018, March 01). Infographic: Spotify's Losses Widen as Royalty Costs Pile Up. Retrieved from https://www.statista.com/chart/4894/spotify-revenue-vs-costs/

Rodrigues, S. (2010, September 03). JavaScript enters the server room with Node.js. Retrieved from https://www.infoworld.com/article/2626383/paas/javascript-enters-the-server-room-with-node-js.html

Total cost of ownership of servers. (2017, November 24). Retrieved from https://www.sherweb.com/blog/total-cost-of-ownership-of-servers-iaas-vs-on-premise/

Turner, G., & Shaw, L. (2017, June 15). Spotify's Loss More Than Doubles Even as User Growth Surges. Retrieved from https://www.bloomberg.com/news/articles/2017-06-15/spotify-losses-widen-after-music-site-flags-accounting-errors