



Boston University

Department of Electrical & Computer Engineering

## **Problem Definition and Requirements Review**

Amadeus: An App for Musicians

Self-Defined Senior Design (EC463) Project

Team 7

Dev Bhatia - devb@bu.edu

Brandon Desiata - desiata@bu.edu

Ryuichi Ohhata - ryu74@bu.edu

James Wasson - jwasson@bu.edu

Samantha Au - samau@bu.edu

Approved: \_\_\_\_\_

Osama Alshaykh, Ph.D.

Department of Electrical & Computer Engineering

Boston University

October 21, 2021

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Project Summary . . . . .	1
1.2	Need for Project . . . . .	1
<b>2</b>	<b>Visualizing &amp; Detailing the Project</b>	<b>3</b>
2.1	Problem Statement & Deliverables . . . . .	3
2.2	Visualization . . . . .	3
<b>3</b>	<b>Competing Technologies &amp; Requirements</b>	<b>6</b>
3.1	Competing Technologies . . . . .	6
3.1.1	MUZE . . . . .	6
3.1.2	Bandfriend . . . . .	6
3.1.3	Instagram . . . . .	6
3.1.4	Facebook . . . . .	7
3.2	Engineering Requirements . . . . .	7
3.2.1	Front-End and Back-End . . . . .	7
3.2.2	Data Management . . . . .	7
3.2.3	Messaging and Searching . . . . .	8
3.2.4	Location Data . . . . .	8
<b>4</b>	<b>Conclusion</b>	<b>9</b>
4.1	Acknowledgements . . . . .	9
	<b>Bibliography</b>	<b>10</b>

# Chapter 1

## Introduction

We begin this report by introducing Amadeus at a high level. By detailing a summary of what the product will be, as well as the general need for the product in the market, we demonstrate the effective uses for Amadeus in a variety of capacities.

### 1.1 Project Summary

Amadeus is a mobile application designed for musicians to connect and self-promote. Focused on the Boston area, the application will allow users to create a profile in order to showcase themselves playing, tether those profiles to a musical group, and network with other musicians. These individuals and groups can communicate with others and promote events to their network.

We see this application as having two prime focuses. First, we will build a comprehensive platform that allows users to search for and connect with other profiles, somewhat akin to LinkedIn, Instagram, or Facebook. Second, we will build a complementary service that will allow musicians to promote their local shows. This will likely be in the form of a hyper-localized map with user-interface tools that aid in guiding the user.

### 1.2 Need for Project

This project began with a series of potential user interviews in which we sought to test the viability of the application in the wider market. Some of the key questions that we sought to answer are as follows -

- Would potential users switch from existing social media platforms?
- Are there enough musicians to justify a platform?
- What would the core user base look like?
- Is there a preference towards one half of the implementation over the other?
- Is there even a need for this product within the music community?

By performing over 30 user interviews, we were able to ascertain the answers to many of the aforementioned questions. These interviews also served as the basis for the problem statement and visualization while giving us a series of competing technologies, both of which will be further discussed in Chapters 2 and 3 respectively. The key takeaways from the interviews are as follows -

- Younger musicians (especially those without an established network) don't have an existing solution
- Finding potential bandmates is a tricky prospect, despite having large-scale social media platforms
- A "try before you buy" approach was highly sought after
- The "business aspects" of starting a band are very prohibitive for many musicians
- Established (older) musicians preferred the "Show Promotion" feature whereas new musicians (younger) preferred the "Connect Local Musicians" feature
- A significant number of musicians preferred to be solo

These insights were imperative in uncovering the core need for Amadeus. We gathered that though existing platforms exist for connecting people, their mass appeal often leave much to be desired in the form of finding a specific "brand" of musician to play with, often preventing many from starting bands or even just jamming altogether. These musicians additionally had little in the form of promoting live shows, which further decreased their propensity towards creating a musical group. The illustrated needs are further exacerbated for certain groups of people - younger musicians, those without an established network of musicians, are particularly susceptible to imposter syndrome, in which they feel as though the entire musical community is far above their level.

We plan to position Amadeus squarely in the middle of this market, enabling musicians of a wide array of skill levels to connect more efficiently and effectively.

## Chapter 2

# Visualizing & Detailing the Project

This section aims to provide an insight as to how Amadeus will work from a more technical standpoint. Building upon the lessons learned in Section 1.2, we aim to provide a comprehensive overview of the problem at hand, the expected result of the building process, and rudimentary visualizations of the service.

### 2.1 Problem Statement & Deliverables

Through the client interviews that we have performed (1.2), we have identified three central problems that are pervasive throughout the musician community -

1. Lack of a **Person** to play with
2. Lack of a **Venue** to play in / attend
3. Lack of a **Centralized Location** to acquire such information

To solve these three problems, we ultimately propose Amadeus. The proposed solution will unify the disjointed, and quite frankly unreliable, aspects of starting a band. By aiding users through the many phases of starting a band and eventually gigging, we hope that our platform will become the de facto application for connecting with like-minded musicians.

**Final Deliverable:** A shippable mobile application with profiles, direct messaging, and a hyper-local map that registered users can add entries to.

### 2.2 Visualization

We often simplify Amadeus to having three core functionalities -

1. Broadcasting a musician's preferences and skills through the profile page and videos
2. Filtering musicians for collaboration
3. Showing musical activities on the map

Figure 2.1(a) shows a sample profile page. The user's music preferences, activities, location, affiliations, and videos will be displayed in this page. Other users can use this page to learn about the musician, helping them decide whether they want to collaborate with this musician or not. Figure 2.1(b) demonstrates the search filter functionality that allows users to find

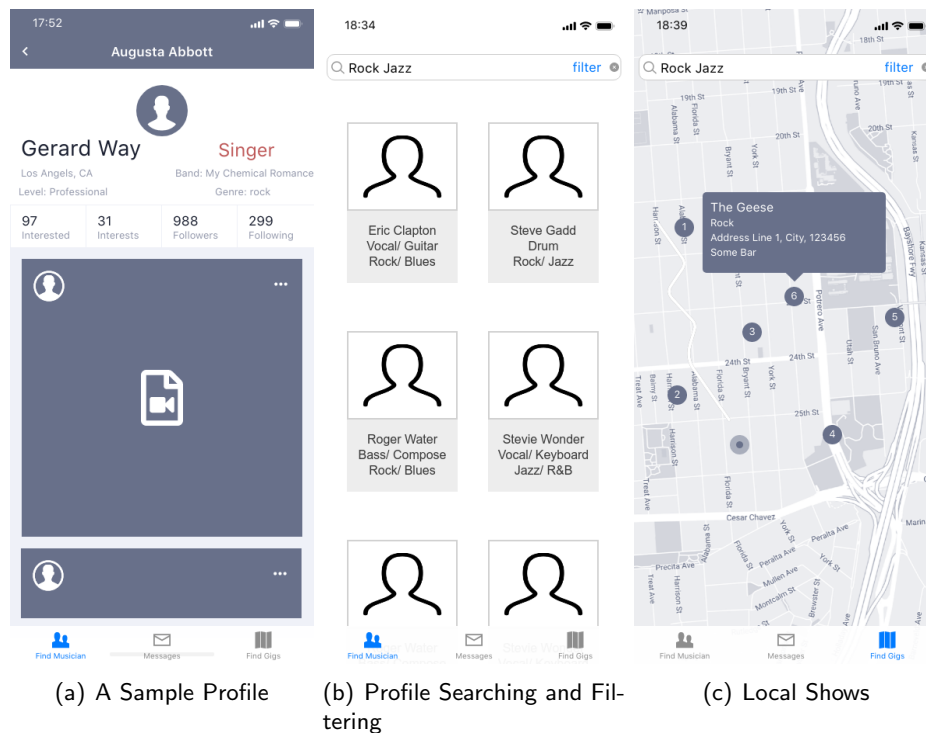


Figure 2.1: Basic UI Views

specific musicians, based on the input criteria. Machine learning on the backend will learn the user's preferences as s/he searches musicians, which will be used to display recommended musicians based on the learned user preferences.

The map function in Figure 2.1(c) can display local gigs based on the keywords specified by users. When the user taps on the gig icon, a text box will pop up with the band's information, address, and genre.

In order to build such functionalities, Amadeus will require us to develop both a front end and back end. The front end will utilize React Native [1] for iOS UI development. It will also incorporate Firebase [2] for Google login, as well as Facebook login. The back end will be responsible for the development of APIs using Python Flask [3], which supports the general dataflow of Amadeus. As Figure 2.2 shows, user information will be stored in a PostgreSQL [4] database, which will be queried for finding local gigs and musicians functions through an API. Kafka's [5] stream processing will enable notifications and enhance communication between devices. [6]

Docker and Kubernetes [7] [8] will be used to containerize the app and scale it in AWS [9]. Kubernetes will be configured to have multiple pods for scaling the app, as well as to prevent a total service shut down in case the pods fail. The PostgreSQL database will be hosted externally for the service reliability, so the server failure will not interfere with the SQL database [10]. A diagram depicting the different sides of this architecture can be found in Figure 2.2.

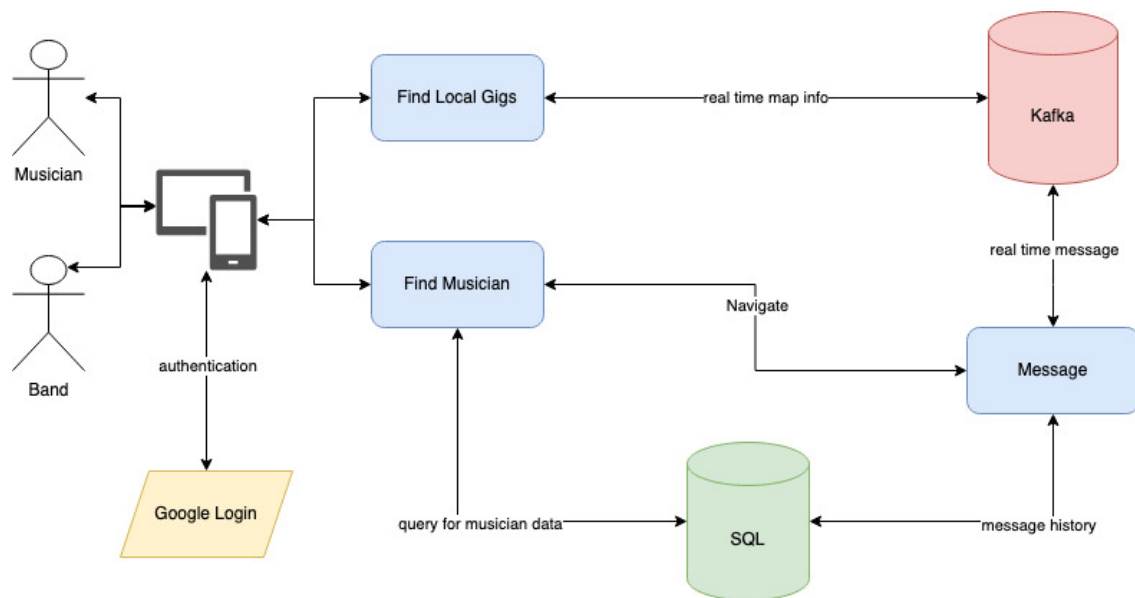


Figure 2.2: Software architecture of Amadeus

## Chapter 3

# Competing Technologies & Requirements

### 3.1 Competing Technologies

The competing technologies to our platform can be categorized into two groups, the first of which consists of platforms that connect their users with musicians. The technologies in the first group (3.1.1 and 3.1.2) contain aspects of UX design that are similar to the ones in Amadeus and target a similar demographic, but they do not fully encapsulate all the functionality of Amadeus.

The second group (3.1.3 and 3.1.4) consists of platforms that do not focus specifically on music but that allow users to connect socially. The target demographic for the technologies in the second group encapsulates Amadeus' target demographic, but the UX of these technologies does not cater to Amadeus' target demographic specifically. Further elaboration can be found below.

#### 3.1.1 MUZE

- Positioned as a platform allowing users to book musicians for events or to attend events created by musicians on MUZE
- Target demographic smaller as MUZE does not allow for interactions between musicians
- Musicians that sell their services on the platform must be chosen by MUZE [11]

#### 3.1.2 Bandfriend

- Positioned as a platform that facilitates connections between musicians
- Contains search functionality that we intend on implementing in Amadeus, including the ability to search for individual musicians and bands based on music genre, distance, interests, and other factors.
- Target demographic is smaller as Bandfriend offers no services for non-musicians [12]

#### 3.1.3 Instagram

- Positioned as a image sharing platform that allows any user to post images of their choosing along with short captions to describe each image



- Caters to a more general demographic as Instagram does not contain music-specific services
- Primary mode of content engagement is through following specific accounts. Though users with similar music interests can be found with the hashtag functionality, this system loses effectiveness when attempting to search for local musicians or concerts [13]

### **3.1.4 Facebook**

- Positioned as a platform allowing any two users to connect to share music, videos, articles, and thoughts and opinions
- Caters to a more general demographic as Facebook does not contain music-specific services
- Primary mode of engagement is through searching for specific users to connect with so that content can be shared. Facebook also contains hashtag functionality so that users with similar interests can be found, but this system also loses effectiveness when trying to connect with local musicians or find local concerts. [14]

## **3.2 Engineering Requirements**

In this section, we aim to take the problem statement as well as lessons learned from the competing technologies and compile a list of requirements that Amadeus should ultimately adhere to.

### **3.2.1 Front-End and Back-End**

1. Users will sign up/sign in through Google or Facebook accounts
2. Penetration tests will be conducted prior to the launch of the app
3. The app must be easily used by more than 80% of users, and the general workflow must be transparent - conforming to the Apple AppStore technical specifications [15]
4. The server must be able to handle at least 50 simultaneous logins and uses
5. A simple machine learning algorithm will be used to learn user preferences for filtering musicians

### **3.2.2 Data Management**

1. An SQL database will be used in order to store data with respect to its relation to other data for at least 3,000 users
2. Data - including credentials collected from the users through the app - must be stored securely on cloud databases

### **3.2.3 Messaging and Searching**

1. Notification and message delays should be under 30 seconds for users with relatively strong networks
2. Search times for retrieving and displaying should be at least 2 musicians per second

### **3.2.4 Location Data**

1. The location data must be able to pinpoint an exact location on the map with less than a 0.1 mile location error
2. The map should not take more than 5 seconds to load gig data in the area specified by the user

## Chapter 4

# Conclusion

We conclude this report by giving some final thoughts to the end goal of Amadeus. Ultimately, we see this platform as a springboard for many musicians to grow confidence in their skills, collaborate with each other, and develop incredible music together. By implementing the above requirements, this project should be easily transitioned to the open market upon completion. Though a highly competitive market, Amadeus carves a comfortable niche in the musical community - one that we plan to continue to learn more about as we develop the service.

### 4.1 Acknowledgements

First, we would like to thank Professors Osama Alshaykh and Alan Pisano for their support in this self-defined project.

Additionally, we would like to give thanks to the Department of Electrical and Computer Engineering for giving us a platform to learn, grow, and develop our ideas.

Lastly we would like to thank our families, without whom this journey would not have been possible.

# Bibliography

- [1] React, *Getting started - react documentation*, 2021. [Online]. Available: <https://reactjs.org/docs/getting-started.html>.
- [2] Google, *Firebase documentation*, 2021. [Online]. Available: <https://firebase.google.com/>.
- [3] Flask, *Welcome to flask - flask documentation (2.0.x)*, 2021. [Online]. Available: <https://flask.palletsprojects.com/en/2.0.x/>.
- [4] PostgreSQL, *Postgresql documentation*, 2021. [Online]. Available: <https://www.postgresql.org/>.
- [5] B. Diamant, *Kafka, meet bruce: How one service boosts app messaging reliability*, Feb. 2019. [Online]. Available: <https://techbeacon.com/app-dev-testing/kafka-meet-bruce-how-one-service-boosts-app-messaging-reliability>.
- [6] Apache, *Apache kafka documentation*, 2021. [Online]. Available: <https://kafka.apache.org/>.
- [7] Docker, *Empowering app development for developers*, 2021. [Online]. Available: <https://www.docker.com/>.
- [8] Kubernetes, *Production-grade container orchestration*, 2021. [Online]. Available: <https://kubernetes.io/>.
- [9] AWS, *Amazon ec2 - secure and resizable compute capacity to support virtually any workload*, 2021. [Online]. Available: <https://aws.amazon.com/ec2/>.
- [10] E. Drkusic, *Learn sql: Types of relations*, May 2021. [Online]. Available: <https://www.sqlshack.com/learn-sql-types-of-relations/>.
- [11] Muze, *Live music performances over zoom*, 2018. [Online]. Available: <https://www.muze-music.com/>.
- [12] Bandfriend. (2016). "Bandfriend · your ios and android app for musician classifieds and more," [Online]. Available: <http://www.bandfriendapp.com/>.
- [13] K. Sehl. (Jan. 2021). "Instagram demographics in 2021: Important user stats for marketers," [Online]. Available: <https://blog.hootsuite.com/instagram-demographics/>.
- [14] S. Mughal. (Oct. 2021). "63 facebook statistics you need to know in 2021," [Online]. Available: <https://www.omnicoreagency.com/facebook-statistics/>.
- [15] Apple, *Submit your ios and ipados apps to the app store*. 2021. [Online]. Available: <https://developer.apple.com/ios/submit/>.