

INTERNATIONAL BURCH UNIVERSITY
FACULTY OF ENGINEERING AND NATURAL SCIENCES
DEPARTMENT OF INFORMATION TECHNOLOGIES



MyDJ Mobile App

UNDERGRADUATE PROJECT

Emerald Podbićanin

Mentor

Assist. Prof. Dr. Zerina Mašetić

SARAJEVO

June, 2021

MyDJ Mobile App

Emerald Podbićanin

Report Submitted in Fulfillment of Requirement for the
Undergraduate Project

INTERNATIONAL BURCH UNIVERSITY
2020/2021

APPROVAL PAGE

Student name and surname: Emerald Podbićanin

Faculty: Faculty of Engineering and Natural Sciences

Department: Information Technologies

Project Title: MyDJ Mobile App

Date of Defense: July 2nd, 2021

I certify that this final work satisfies all the requirements as a Undergraduate Project for the Bachelor degree in Computer Science Engineering.

.....
Assist. Prof. Dr. Zerina Mašetić
Head of Department

This is to certify that I have read this final work and that in my opinion it is fully adequate, in scope and quality, as an Undergraduate Project for the Bachelor degree in Computer Science Engineering.

.....
Assist. Prof. Dr. Zerina Mašetić
Mentor

Examining Committee Members

	Title / Name and Surname	Affiliation	Signature
1.		head	
2.		member	
3.		member	

It is approved that this final work has been written in compliance with the formatting rules laid down by the Department of Information Technologies.

.....
Head of Committee

MyDJ Mobile App

ABSTRACT

Many parts of our everyday lives have been digitized, but there is one part that has not. Almost every person found themselves in a situation where they would go out to enjoy a night with their friends but the music being played would just not fill their needs or suit their taste. Adding the digital factor, in the form of a mobile application, would solve this problem. MyDJ app brings democracy to the table and lets everyone have their say for the next song that will be played that night. Every person attending a certain event, would be able to join the event's lobby with their provided lobby code. Inside that lobby, the DJ for the night creates dynamic polls where users are able to vote for a song they prefer the most. The most voted song gets played next and so on, until the end of the event. This would refresh and raise the standard of events that are based on listening and dancing to music in public. It would bring a new dynamic to both the owner and the customer and make everything more fun in general. The technologies used for development were Flutter and Dart for the frontend of the application and Firebase for the backend and the service side of the application. The app works for both Android and iOS.

Keywords: democracy, polls, music, mobile app, song, android, ios

ACKNOWLEDGMENTS

There are many people who helped to make my years at the graduate school most valuable. First, I thank Professor Zerina Mašetić, my major professor and dissertation supervisor. Having the opportunity to work with her over the years was intellectually rewarding and fulfilling. I also thank Admir Salihagić who contributed much to the development of this project starting from the early stages of my dissertation work. He also provided valuable contributions to the development of the econometric model. I thank him for his insightful suggestions and expertise.

Many thanks to the Department computer staff, who patiently answered my questions and problems on word processing. I would also like to thank to my undergraduate student colleagues who helped me all through the years full of class work and exams. My special thanks go to Elmedin Karišik, Mehmed Tukulović and Faris Bektaš, whose friendships I deeply value.

The last words of thanks go to my family. I thank my parents Lejla and Dženan and my sister Adna for their patience and encouragement. Lastly, I want to thank all my friends from outside the university for their endless support through this long journey.

DECLARATION

I hereby declare that this Undergraduate Project titled **MyDJ Mobile App** is based on my original work except quotations and citations which have been duly acknowledged. I also declare that this work has not been previously or concurrently submitted for the award of any degree, at International Burch University, any other University or Institution.

.....
Emerald Podbićanin

June 28th, 2021

TABLE OF CONTENTS

APPROVAL PAGE	i
MyDJ Mobile App.....	ii
ABSTRACT	ii
ACKNOWLEDGMENTS	iii
DECLARATION.....	iv
TABLE OF CONTENTS.....	v
LIST OF ABBREVIATIONS.....	viii
1. INTRODUCTION.....	1
2. SYSTEM ANALYSIS	2
2.1. Functional requirements.....	2
2.1.1. Authentication.....	2
2.1.2. Administrative functions.....	2
2.1.3. Lobby functionalities	2
2.1.4. Poll functionalities	3
2.1.5. Song library.....	3
2.1.6. Voting.....	3
2.1.7. User song suggestions	4
2.2. Non-functional requirements	4
2.2.1. Security	4
2.2.2. Performance	4

2.2.3.	Usability	4
3.	APPLICATION DESIGN	5
3.1.	Use case diagram.....	5
3.2.	Activity diagrams	6
3.2.1.	Admin activity diagram.....	6
3.2.2.	User activity diagram	7
3.3.	Class diagram.....	8
4.	IMPLEMENTATION.....	9
4.1.	Technologies used for development.....	9
4.1.1.	Flutter and Dart	9
4.1.2.	Firebase	9
4.2.	Application results	10
4.2.1.	Lobby admin home page.....	10
4.2.2.	Lobby creation screen	11
4.2.3.	Lobby admin Songs tab.....	12
4.2.4.	Lobby admin Poll tab	13
4.2.5.	Lobby user home page	14
4.2.6.	Lobby user in-lobby screen.....	15
4.2.7.	Lobby user song suggestions tab.....	16
4.2.8.	Poll winners screen	17
5.	SYSTEM TESTING.....	19
5.1.	Unit testing	19
5.2.	Integration testing.....	20

5.3.	Future testing and conclusion.....	20
6.	MAINTENANCE ANALYSIS.....	21
6.1.	System maintainability.....	21
6.2.	Data integrity and security	21
7.	CONCLUSION.....	22

LIST OF ABBREVIATIONS

API	Application Programming Interface
SDK	Software Development Kit
UI	User Interface
UX	User Experience
QA	Quality Assurance

1. INTRODUCTION

I believe that almost every person finds themselves in a situation where they go out to have a fun time with their friends but the music being played simply puts the atmosphere down because they have never heard it before, or it just does not suit their taste. From personal experience, I know how underwhelming this can be. Exactly this, inspired me to seek for a solution that would hopefully solve this problem. My plan is to include democracy in the song selection process and let every single individual have their say. I feel that only in this manner, we could find the balance between what is fair and what is fun to provide the best possible experience for both the owners and the customers. In addition, there are no current studies around this problem, or any applications developed before, as long as I'm aware of. To have a successful mobile application, you have to solve a problem never solved before, and that is the case here. The importance of this application could become huge if it gets accepted as the new standard. What I mean by that is that people could develop a habit of picking up their phone and opening this application every time they go out to have a fun time at places with music in the background. The fact that there are no applications of this kind currently in the market means that there is a bigger possibility for this app to go viral faster.

2. SYSTEM ANALYSIS

2.1. Functional requirements

Functional requirements are the primary way that a customer communicates their requirements to the project team. Functional requirements help to keep the project going in the right direction. Unclear requirements might lead to a poorly defined scope and create unnecessary problems from the beginning of the project. Every functional requirement listed below is mandatory and focused on user needs.

2.1.1. Authentication

Every user must be able to register a unique profile with an email, password and username. The email must be valid, the password must be at least 8 characters long, and the username should be at least 4 characters long. After a successful registration, users are able to login to the application's home page. The application cannot be used without authentication.

2.1.2. Administrative functions

There are two types of users, one being the general users and the other one being the administrators (music club owners or music club DJ's). The administrators will have their own special functions such as creating the lobby, creating the voting polls and adding or deleting new songs. To be able to login as the administrator, you must have to provide the admin code. The admin code will be generated by the project team and handed to the respective admin users.

2.1.3. Lobby functionalities

Admins must be able to create new lobbies and the users must be able to join them with a lobby code which will be provided to them by the admins. Every lobby must

have a name (maximum 10 characters long), a capacity (how many users can enter the lobby), a code (must be unique and must contain letters and numbers), a duration (how long does a poll last) and the number of songs inside one poll. When inside the lobby, the number of active users inside the lobby must be visible to everyone. Users must be able to leave the lobby, as well.

2.1.4. Poll functionalities

A poll can only be created by an admin. One poll can have a duration of 60, 90 or 120 seconds. Also, it can have a song per poll number of 4, 5 or 6. Both of these are selected at the lobby creation moment and cannot be altered after.

2.1.5. Song library

Song library is only visible to admins. They must be able to add new songs to their library and delete songs from it, as well. One song must have a name, an artist (person performing the song) and a collection of music genres that correlate with it. Songs are used as poll fillers and without them the polls cannot be created. Every admin must be provided with a Song Starter Pack that will contain around 100 songs in total and will be available only when the admin enters the application for the first time. Lastly, there must be a search option that will ensure that songs can be found as quickly as possible.

2.1.6. Voting

The voting process lasts as long as the selected poll duration. There must be a timer functionality that will display the exact number of seconds left until voting ends. One user is able to vote only once per poll. A user can vote for one song only. The voting percentages are visible to users only after they have voted, but the admins can always see them. After voting ends, the song with the most votes is added to the poll winners section of the application. That section is always visible to everyone.

2.1.7. User song suggestions

Every user inside a lobby must be able to write a personal song suggestion. That suggestion should be taken into consideration by the admin when adding songs at the poll creation moment. The song suggestions section is visible to everyone inside the lobby and every suggestion can be marked as ‘liked’, with a like counter visible to everyone as well.

2.2. Non-functional requirements

Nonfunctional requirements define system attributes such as security, reliability, performance and maintainability. They serve as constraints or restrictions on the design of the system across the different backlogs. In this case, these requirements should be rather simple.

2.2.1. Security

Administrative functions can only be performed by the administrators, and they can never be visible to a regular user. All data must be safely stored in the database, and it cannot be accessed without permission.

2.2.2. Performance

Every user must have a stable internet connection in order to use the application at all. Also, to be able to use the application at the highest performance, it is recommended to use an iOS version higher than 10.0 or the minimum android SDK version of 23.

2.2.3. Usability

The application must have the easiest user experience possible, with a neat and simple UI that is efficient and intuitive to use. People using the application will be in the mood for fun and the UX must ensure that the fun is not spoiled, even for a little bit.

3. APPLICATION DESIGN

In the following sections, I will represent all the necessary diagrams to showcase my application workflow in a visual and intuitive way.

3.1. Use case diagram

As you can see in the diagram below, there are three types of actors in my application. The primary actor is the lobby admin as he initiates everything else. The lobby user can vote, see the voting results and optionally suggest a song, and the system calculates the winners and ensures the lobby timer works properly. Once an admin creates a lobby, he creates a poll as well and after the poll finishes he checks the poll results. The lobby user joins the lobby first, then votes for his favorite song and checks the poll results. He can optionally suggest a song of his own preference if he wants to.

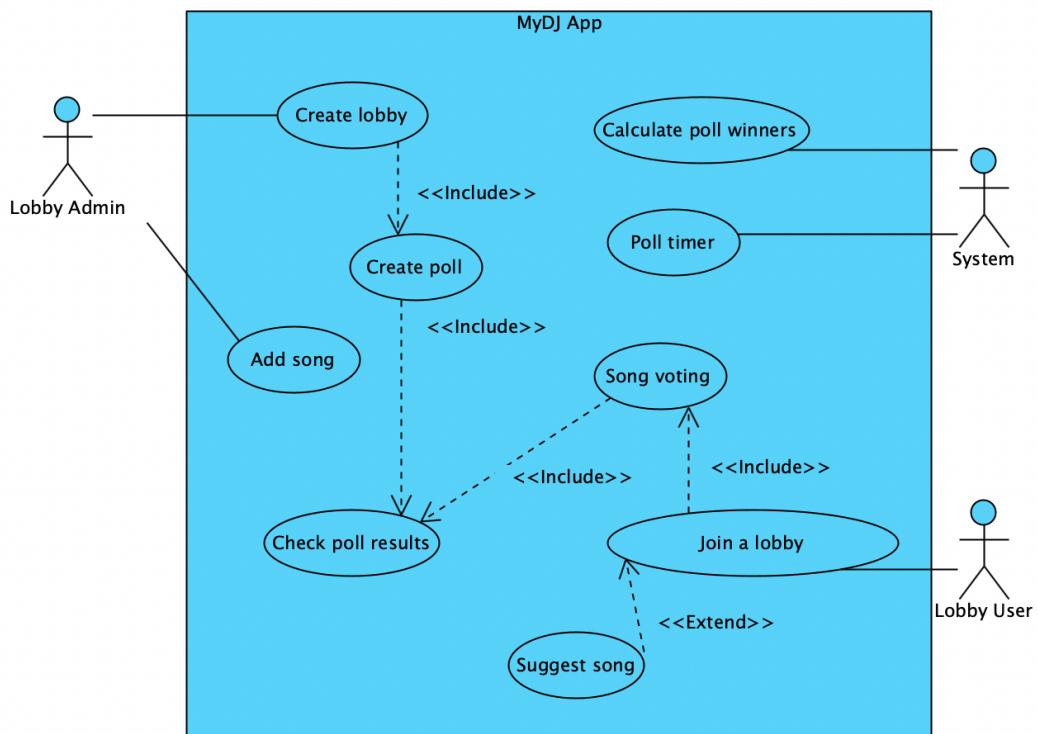


Figure 1.1. Use case diagram

3.2. Activity diagrams

There are two types of activity diagrams in my application and those are Admin and User diagrams. The flow is similar but there are some key differences that can be spotted in the diagrams below.

3.2.1. Admin activity diagram

The admin firstly logs in, then he can create a lobby. After that he can create a poll, add a song or check the current song suggestions. In the end, he can log out of the app.

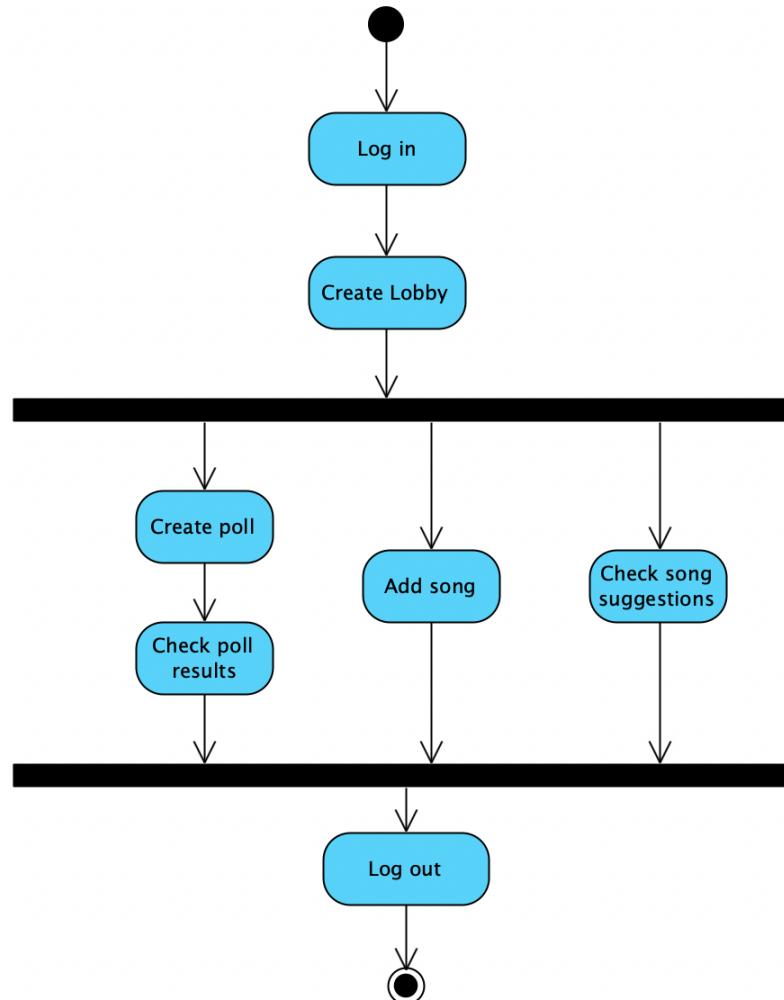


Figure 1.2. Admin activities

3.2.2. User activity diagram

The lobby user can firstly log in, then he can join a lobby with the provided lobby code. After that he has two options, he can vote and check the poll results or he can suggest a song of his own liking. In the end he can simply log out of the application.

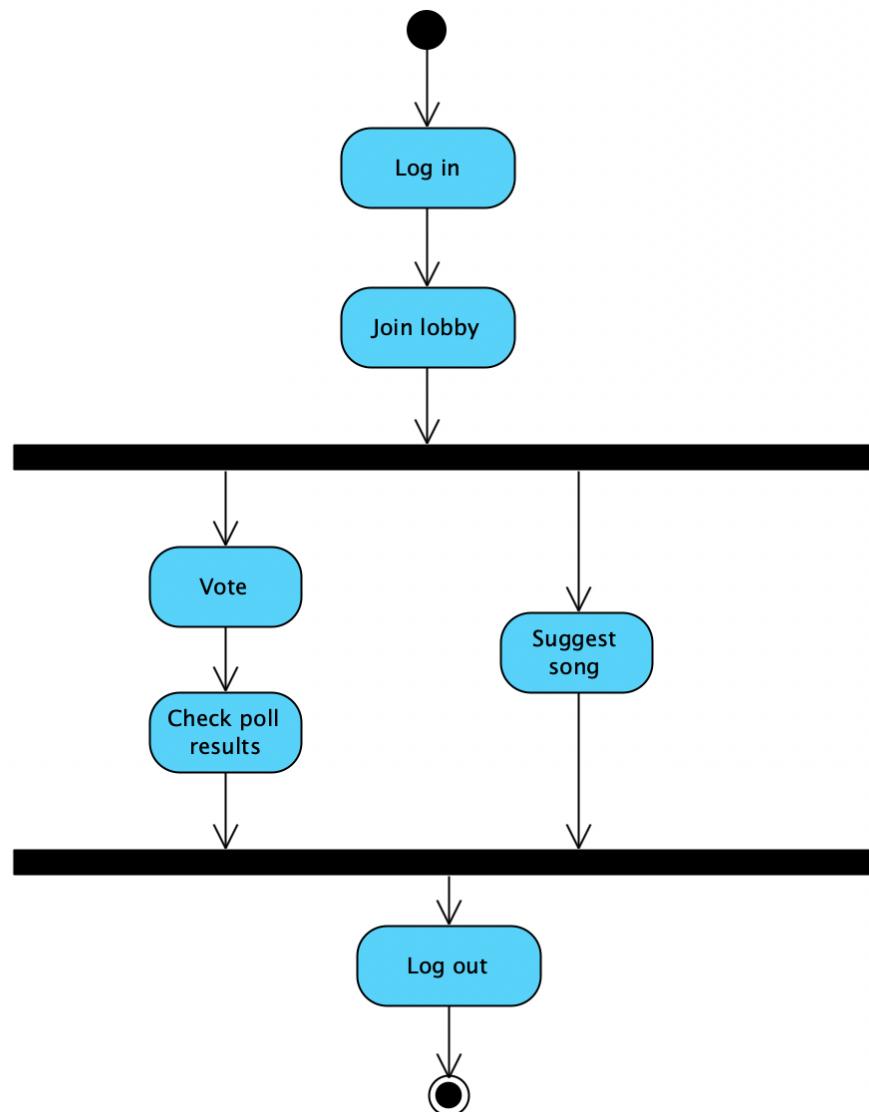


Figure 1.3. User activities

3.3. Class diagram

The class diagram below, indicates certain usage dependencies. The Lobby class cannot exist without a Lobby Admin, and the Poll class cannot exist without an existing lobby first. There are two types of users and that can be generalized with an abstract User class. The Lobby Admin and Lobby User classes inherit the User class and then add their own methods and attributes. Poll winners are stored inside the Poll class. All active users are stored inside the Lobby class as a list of Lobby Users. Lastly, the Song class is only directly related to the Poll class and the Lobby Admin class, when it is being added to the song poll by the Admin.

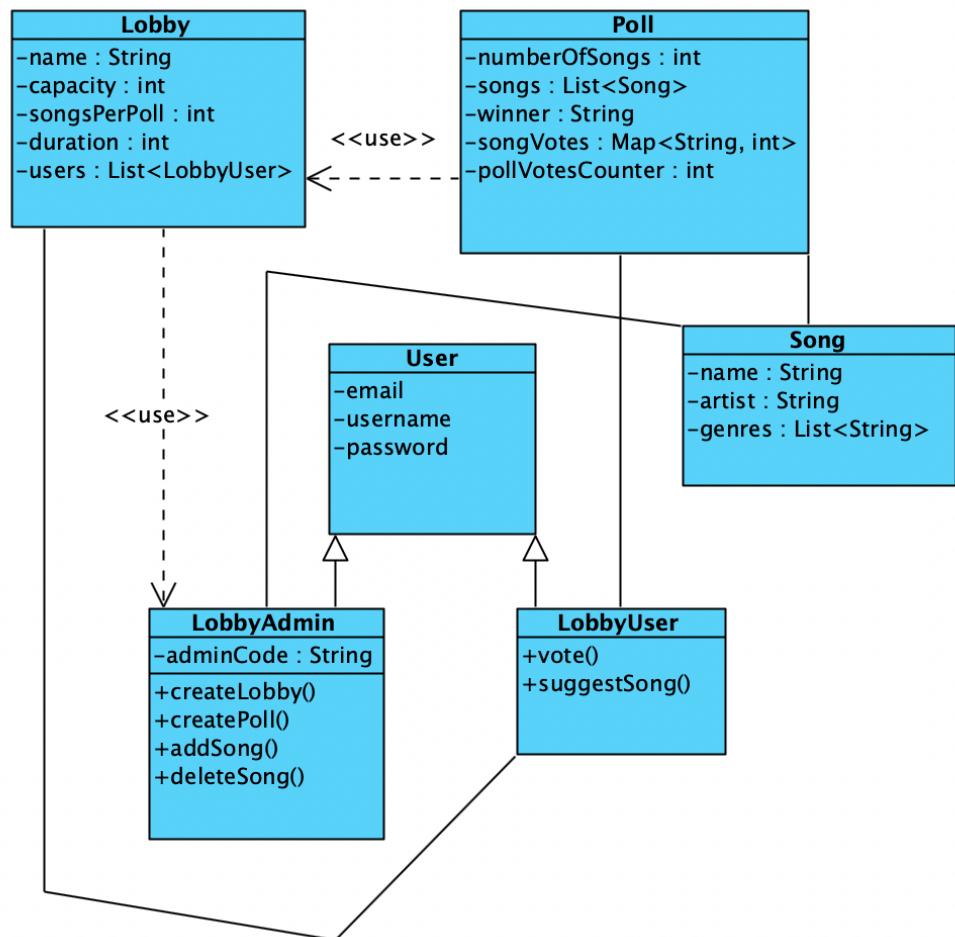


Figure 1.4. Class diagram

4. IMPLEMENTATION

4.1. Technologies used for development

In the following sections, I will describe the tech stack used for implementing my ideas into an actual software product.

4.1.1. Flutter and Dart

Flutter is a user interface SDK developed and released by Google. The programming language behind Flutter is Dart. Dart was created in C++ with an intention of simplifying mobile development. It was January of 2021 when I first heard about Flutter and its features. I have to say, that was one of the biggest revelations in my software engineering career as I immediately felt that Flutter was the perfect fit for my personality and what I seek in this industry. It gives developers a lot of comfort and ease when developing cutting-edge mobile apps. The biggest advantage of Flutter is that it can be used for cross-platform development. That convinced me to choose Flutter for my Senior Design Project, because it would give me an opportunity to have the exact same application design and workflow for both Android and iOS. Flutter compiles the Dart code into native code with great efficiency, and then renders widgets directly on the device screen. I believe that Flutter will be the future of mobile applications, but it will take some time before that happens because a lot of current companies have active projects in other technologies and do not feel the need to make a switch. Lastly, I must mention that Flutter not only gives a great user experience but also a great developer experience, which is what makes it really special for me.

4.1.2. Firebase

Firebase is Google's service platform with many features and capabilities. It is a document-based database, where I stored all my application data. Firebase works perfectly with Flutter because it can be easily integrated inside a Flutter project. It is

fast, reliable and secure. It helped to create my own API and it filled all the needs I had before starting to implement my application. The greatest thing with Firebase and my particular project is the Stream Builder feature. This feature follows and streams any change of data in a certain document or collection. It allowed me to create lobbies and polls for multiple users and devices in a live streaming fashion. So when a Lobby admin creates a poll, the users active on the Lobby screen can immediately see the newly created poll. It also allowed me to stream the Lobby timer, a feature that is so valuable to my whole idea. When I first started thinking about my app needs, I felt that it would be too hard for me to make it exactly as it is in my mind. But using Firebase with Flutter helped me achieve everything I wanted with my application. I would recommend this tech stack to anyone who wants a beautiful and fast mobile application for cross-platform development.

4.2. Application results

4.2.1. Lobby admin home page

Here you can see how the lobby admin home page looks like. It has 3 bottom navigation tabs: Lobby, Poll and Songs. On the lobby tab, the lobby admin can create a new lobby or restore the previously created one, if he accidentally closed the entire application and restarted it again.

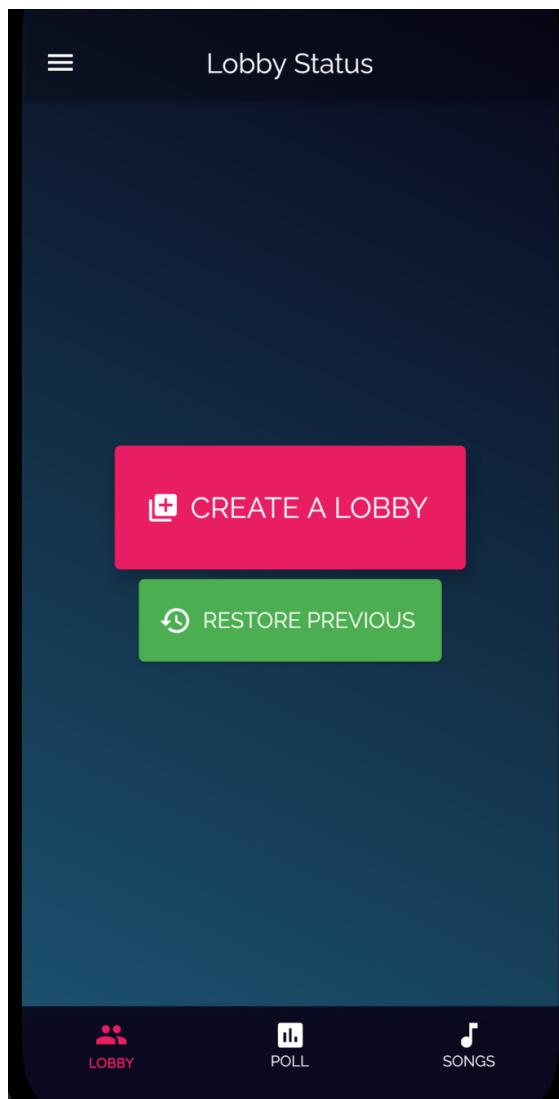


Figure 2.1. Lobby admin home page

4.2.2. Lobby creation screen

On this screen, the lobby admin is able to create a new lobby. There is validation for all 5 inputs. I must mention that in this screen design I really showed my talent and what I am capable of when it comes to perfect UI/UX design.

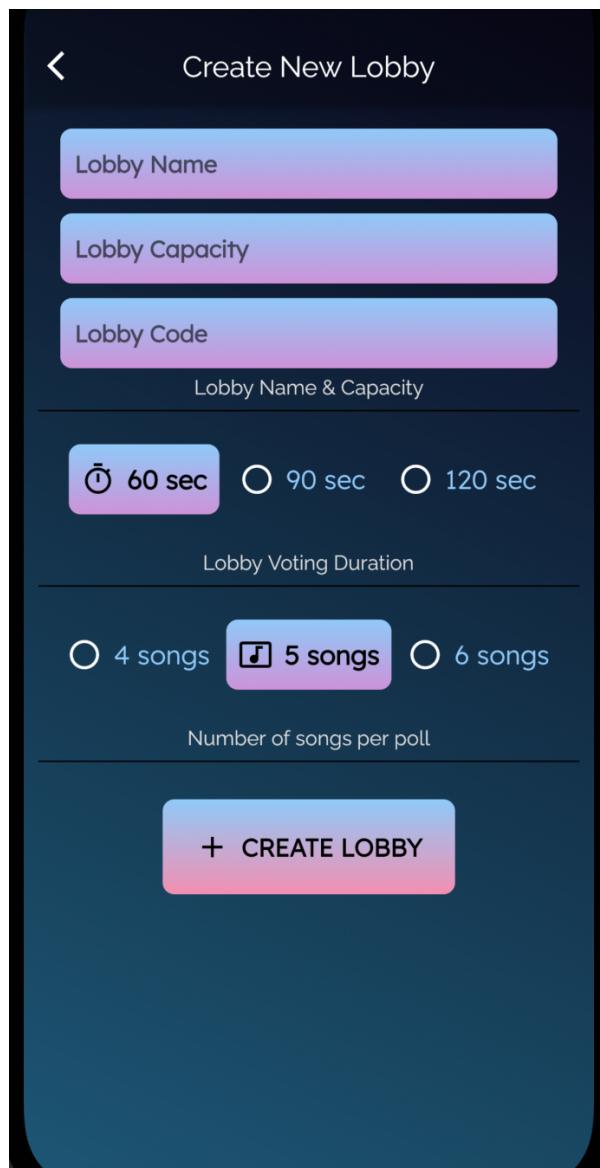


Figure 2.2. Lobby creation screen

4.2.3. Lobby admin Songs tab

On this tab, the lobby admin can quickly find songs that he wants to add to the next poll. He can either search for a song or try to find songs through their genre.

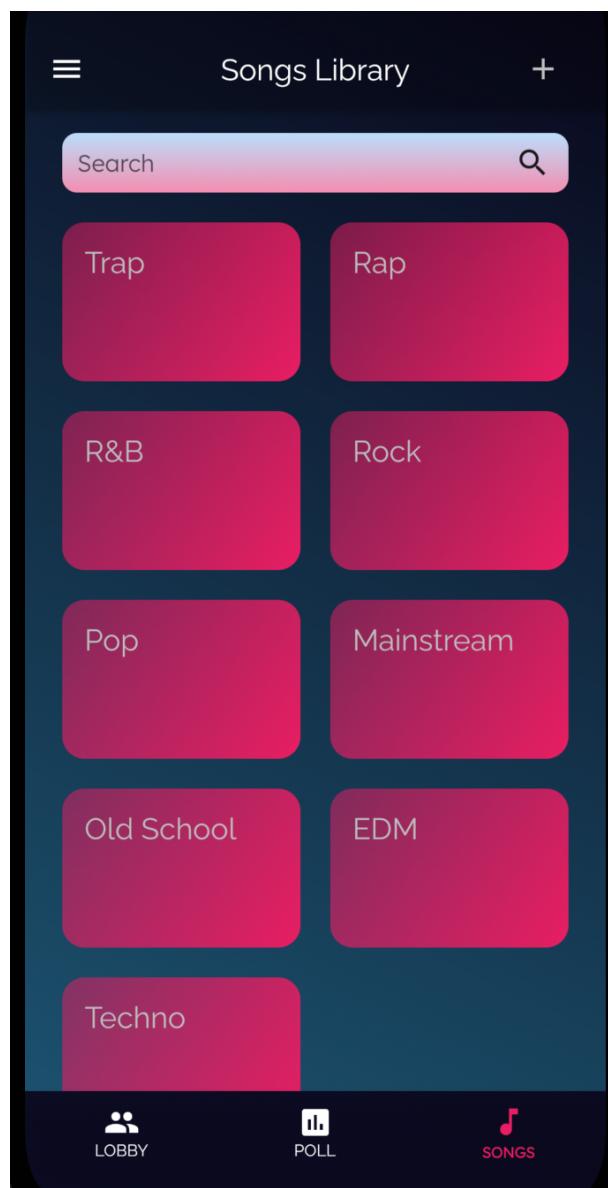


Figure 2.3. Lobby admin Songs tab

4.2.4. Lobby admin Poll tab

After adding songs to the poll, the lobby admin can finally create it in the Poll tab. After he clicks “Create Poll” the lobby timer will start and the voting for the next song will be opened.

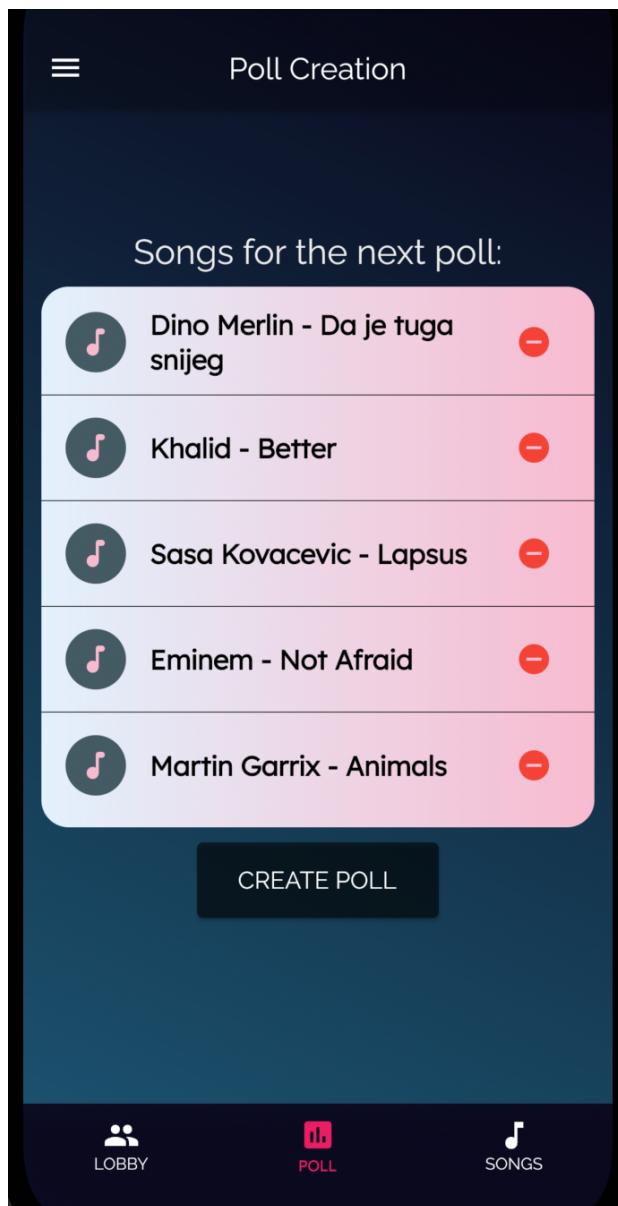


Figure 2.4. Lobby admin Poll tab

4.2.5. Lobby user home page

This is the screen that the lobby user sees first after successfully logging in. It is rather a simple page with only one input where the user must type the correct lobby code in order to be redirected further inside the lobby.

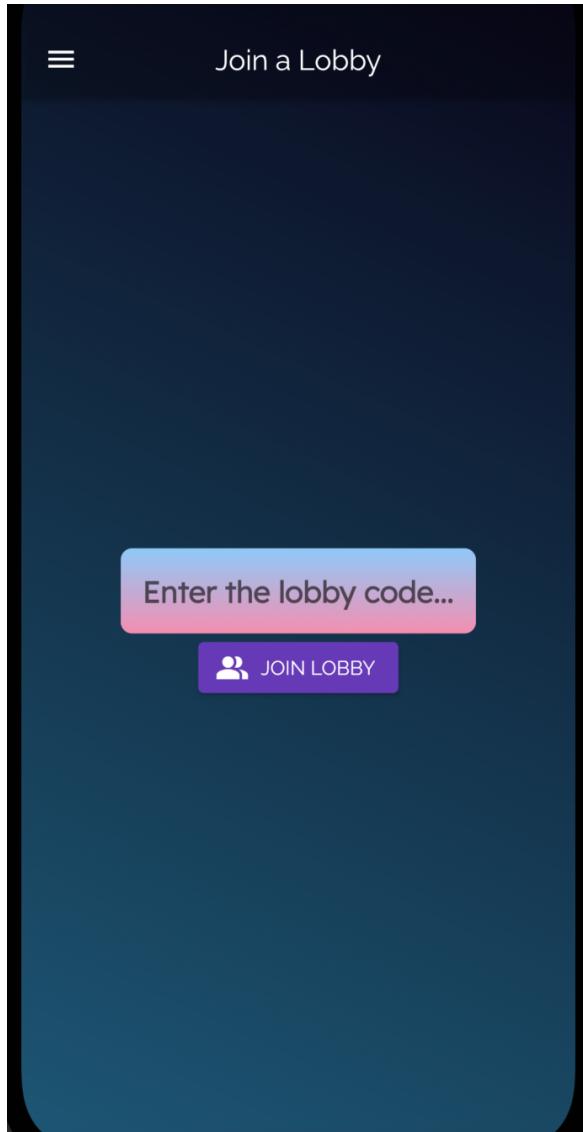


Figure 2.5. Lobby user home page

4.2.6. Lobby user in-lobby screen

When a user successfully joins a lobby, he is redirected to this screen that consists of three tabs: Lobby, Suggestions and Winners. On the screenshot below, you can see the Lobby screen, where the user is able to pick one song and vote for it before the timer expires. Only after the vote, the user is able to see the results of voting.



Figure 2.6. Lobby user in-lobby screen

4.2.7. Lobby user song suggestions tab

On this tab, all the lobby users inside the lobby can suggest a song to be considered in the future poll creations. Also, both the admin and the user can like suggestions and the like counter is streamed live to all users active on this screen.

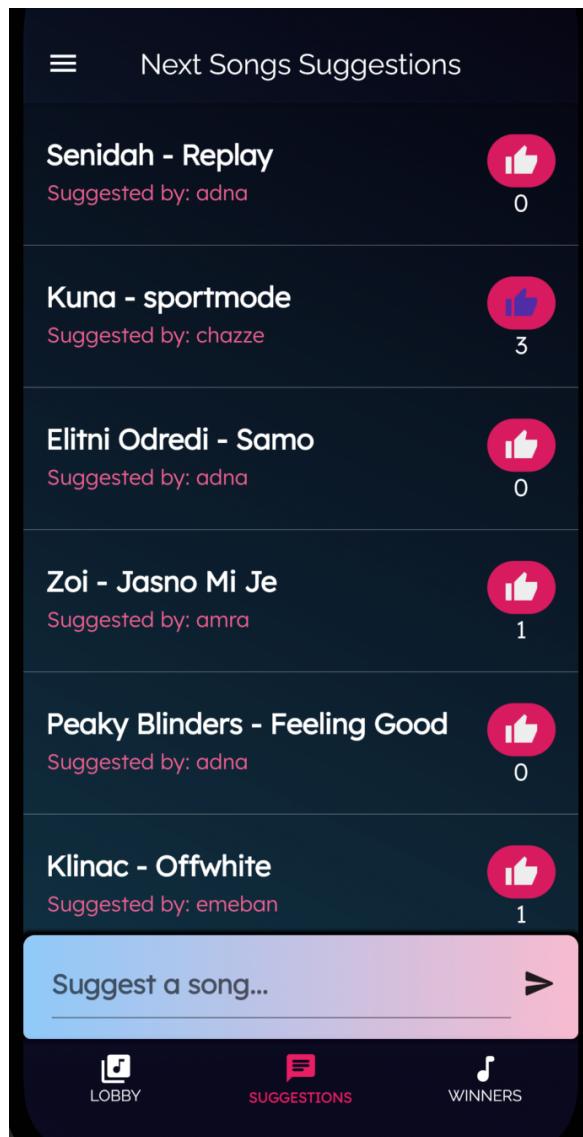


Figure 2.7. Song suggestions screen

4.2.8. Poll winners screen

The last but not least screen is the Poll winners screen. This screen is available for everyone inside the lobby and this screen updates live whenever a lobby timer reaches an end and there is a song that won the most recent poll.



Figure 2.8. Poll winners screen

5. SYSTEM TESTING

System testing is an important part of a software product lifecycle. Unfortunately, I never tested a mobile app before, so this is my first time and because of that I kept the testing simple. Of course, in the future if this application gets published to the stores, I will need proper QA testing done by a proper set of QA engineers. Right now, I am more focused on the frontend and backend development, but I still managed to test the application in an acceptable way. There were two types of testing: Unit and Integration testing.

5.1. Unit testing

For unit testing of the application, I had the following unit tests:

- Add song test (is the song added to the songs list)
- Delete song test (is the song deleted from the song list)
- Create lobby test (is the lobby created as expected)
- Lobby timer test (does the lobby timer run and end properly)
- Create poll test (is the poll created as expected)
- Add user to lobby test (is the user added to the lobby)
- User leaving the lobby test (did the user leave the lobby properly)
- User voting test (can the user vote for a song as expected)
- Poll winner test (does the most voted song win the poll as expected)

5.2. Integration testing

Integration testing is used for testing the application's overall performance. This was my first serious end-to-end application and that is why this test did not have the best possible result, but it was still acceptable. I did not have the best performance practices when I started developing this application but nearing the end of development, I learned a lot and I know that in my next big application I will indeed start in the best possible way. To be more specific, on some widgets I used the widget state in a cumbersome way but as I progressed further, I started noticing better state management patterns. That is why I am thankful for this project, as I learned how to deliver a full software product to the users.

5.3. Future testing and conclusion

In the future, when this application comes close to a public release, I would hire QA engineers to test my application and provide me the results. As I mentioned earlier, I am not that interested in QA like I am in full stack, so I will require new people in order to deliver the best version of my software product.

6. MAINTENANCE ANALYSIS

6.1. System maintainability

The application's system requires strong maintenance right after publishing the app on the stores. The app will need weekly checks to maintain a flawless version of itself. User feedback will contain valuable information for finalizing the system and after the first few version updates, the system will require less maintenance.

6.2. Data integrity and security

Firebase has a simple way of integrating application data and keeping everything secure. There are certain security rules you can set in the Firebase Cloud Firestore so nobody without a successful authorization, can access any data. Hence Firebase is an excellent service to have as your backend, but it also comes with a price. It is free in the beginning and if your app reaches great popularity, you must pay as much as the application usage and number of users increases. Of course, I find the Firebase services really worth it, because they do a lot of heavy lifting behind the scenes to ensure that the application flows in the most efficient fashion.

7. CONCLUSION

At the end, the application has implemented a whole new experience for both the owners and the customers. The main features of the app are Lobby creation, Poll creation, Lobby timer, Lobby poll winners and the Lobby song suggestions. All these features work together to ensure the best possible user experience and to provide a new level of entertainment in public places where there is music playing in the background. This was the biggest application of my current engineering career, and I could not be prouder of what I personally achieved. At the start of this idea, to be honest, I did not believe that I could accomplish everything that I imagined about this app, but in the end, when I look back, there were some memorable moments. The most memorable achievement was the live voting and live lobby updates. That part scared me the most in the beginning, but with dedication and the ability to search the world wide web properly, you can achieve everything you can imagine. After this project, I really feel comfortable with mobile development and I am ready and eager to start my mobile developer career soon, hopefully. I learned a lot on this journey, but there is one limitation in my app. That is that everyone with a lobby code can join the lobby, even if they are not in the music club at that moment of time. That could be solved by using geolocation to only allow users to join the lobby if they are in the radius of 500m. That would bring the application on a level of perfection, and it would then definitely be ready for publication and an official release. Lastly, I hope that one day, my application will be available worldwide on the app stores, and that it will reach and help a lot of people to actually enjoy their music, when they go out with their friends.

