View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DigitalCommons@CalPoly

DJukebox

A Mobile Application Senior Project

Student: Alexander Mitchell
Computer Science Department
California Polytechnic State University
June 14, 2017

Abstract

I'm going to discuss the process used to research, design, and develop a mobile application to handle song requests from patrons to disc jockeys. The research phase was completed in the first half of the project, during CSC-491, along with much of the design. The rest of the design and all of the development was completed during CSC-492. Once development began there were times when reverting back to the design phase was needed, which became apparent as more was learned about the mobile platform chosen for development, Android, and the back-end server utilized, Google Firebase. Ultimately the project was purely academic, as there was not a real market or desire for the application as a product, so it was not published to the Google Play Store, and no company or corporation was formed around the application's creation to take it to market.

Project Intro / Thesis

Idea

The idea for this project came to me because it was something that I wanted to exist. I wanted to be able to use this product, and due to that fact that it didn't already exist, I thought it would be a good project to explore and create for my senior project. I found myself at various events – weddings, karaoke, bars, parties, mixers, etc. – wishing that there was a better way for me to request a song from the person in charge of the music. All of these events involved music, and at many of them the music was quite loud, making requesting a song verbally somewhat difficult. Then even if a request was made, how did I know if the request was heard correctly, or if the disc jockey was receptive of the request or if they even had the ability to play the request? Did they have the song? Were they allowed to stray from a certain set list? These were questions I wanted answered. I wanted a better way to go about requesting songs from disc jockeys, so I set out to create one.

Problems solved

If there was a mobile application to handle these requests, then it would solve a lot of the problems and issues surrounding the request. The problem of whether or not the disc jockey properly heard the request is solved because the auditory issue is removed; now they just read the request. The requester could know if their request was received well and would be played, or if it won't be played, if a simple feature was added to the application to relay back to the requester the status of their request. This was one of the most crucial components of the project. If a patron was at a venue where a disc jockey was playing, and requests a song, they may only want to stay at that particular venue if their song is going to be played. In addition, disc jockeys may not want to reject requests right to patrons' faces, and if there was a less direct way for them to let the patron know they will not, or cannot, play the request, this could be a great selling point for disc jockeys. All these problems could be solved by the creation of this product I was imagining. In addition, from my research I found that, disc jockeys on average regard listening to patrons' requests as a hassle and this project removes that annoyance from the request because it requires far less of their time and concentration.

Potential Customers

As I thought of this application and how it could be used, I thought of who the potential customers of it could be. Disc jockeys would be the main customers, along with their patrons, but this product could be useful to other people as well. Karaoke jockeys could find it useful, in that requests could be made simultaneously, and could be made while the jockey is busy queueing up subsequent songs. Masters of ceremonies (MCs) could find this product useful at a wide range of events they may be in charge of, as long as there was music to be requested, or even – with some minor adjustments – anything to be requested; activities, the next person to be chosen for an activity, movies, etc. Even bartenders who are in charge of the bar's music while a disc jockey is not playing would find this application useful because then they could separately handle the music on their device, while talking to patrons to fill their food and beverage orders.

Research Methods

Although, to a certain degree, research is not required of a senior project for computer science, I wanted to do some. I could have just thought up a project, designed and developed it, but I wanted to go about it a little differently. I wanted to do research so that I could find out what potential customers wanted, instead of developing something and then altering it to be what people wanted; I believe it is a better idea to create what is desired from the beginning. This is along the same line of reason that argues why requirements-gathering is so important in the software development lifecycle. When I set out to begin the research phase, I knew I wanted to talk to the various customers of the application, namely the disc jockeys and general patrons. Before I began to gather information from both of these sets of customers though, I wanted to do a few interviews with disc jockeys to see if this project was realistic and would fit into the procedures they use, as I am not a disc jockey and am unfamiliar with how they operate. After completing these interviews I moved on to less intense, smaller information-gathering means. I decided to separately gather information from the two main groups of customers, so I created a document for each. The questionnaires, or surveys, had open-ended, free-answer questions where the participant would write their answer in the space provided, and questions with intensity levels from one to ten for the participant to choose from, and some that included both a scale and a space for elaboration. These questionnaires can be found in the appendix. An informed consent form was signed by each of these subjects and this form can also be found in the appendix.

Research Results

Difficulties

From the interviews with disc jockeys, I learned quite a lot about the culture that surrounds them and how best to approach them. I also learned that they are not the most receptive group of individuals when it comes to new ideas, projects, academia, etc. I was only able to get three different disc jockeys to take my survey after I completed my interviews. I approached countless others, but to no avail. This was one of the most frustrating parts of my project and certainly the most frustrating part of the research. I was seeking out these potential customers of my project and asking them what they would like me to develop; what would they

like to see in a free product I was proposing to make. All they had to do was listen and give input and they could have had a great product to make their job easier, free of charge. Either they didn't care or merely didn't realize this, because my requests for an audience with them fell mostly on deaf ears.

DJ Questionnaire Results

Of the three disc jockeys I was able to survey, all three suggested the project be made for iOS and as a web application to be run in a browser, and two of them indicated a desire for an Android version. When rating the satisfaction gained from fulfilling requests while working as a disc jockey, from one to ten (ten being the most), one indicated a five out of ten, one indicated an eight out of ten and one indicated a ten out of ten. When asked about how annoying it was to take requests when working as a disc jockey, one subject found it most annoying and the other two found it to be neutral. When asked to rate the difficulty of taking requests when working as a disc jockey from one to ten (ten being the most difficult), I received a four, a seven, and an eight on the scale from my subjects. While this was a limited sample size, I was still able to see some patterns in it. These patterns showed that disc jockeys viewed taking requests as difficult, annoying or neutral, and satisfying. The elaborations from the surveys from my subjects can be found typed up in the appendix section.

Patron Questionnaire Results

I was able to get a larger sample size from the possible patrons of the application. Their survey was much simpler, and for the most part, applies to anyone who would ever request a song from a disc jockey in any situation at any venue. These subjects were mostly friends. I still did not have a large sample size, but it was enough to get an idea of what people thought of this project before I started designing. I was able to get nine subjects for my survey this time. All nine indicated that they would use a mobile application to communicate a request to a disc jockey at a venue or party. All nine also reported that they had never heard of any such application that already exists. I did my own research to add to my certainty of this as well. When asked to rate the importance of knowing their request was seen by a disc jockey from one to five (five being the most), two subjects indicated a level of three, four subjects indicated a level of four and three subjects indicated the highest level of five. When asked to rate the importance of knowing if their request had been rejected on a scale of one to five (five being the highest), two subjects reported a level of two, one subject reported a level of three, four subjects reported a level of four, and two subjects reported the highest level of five. This was enough data to give me some slight trends to base my design off of. These trends showed that most patrons would use the application if it were available, and that it was important to patrons to know that their request was received by the disc jockey. The data also showed a trend that patrons feel that knowing if their request was rejected is at least somewhat important. From here I could move on to the design stage of my project. The elaborations of these surveys can also be found in the appendix section.

Design Methods

Initial Graphical User Interface Prototype

My original graphical user interface design was based off of what I wanted to patron to see when they loaded a specific disc jockey's session. I wanted them to be able to see the dis jockey's name, where they were performing, and what time they end, along with a list of requests and the status of those requests (whether they had been accepted, rejected or not answered yet). I also considered the flow of control through the application and how it would be split up between the disc jockey functionality and the patron functionality. I roughly sketched out a control flow of the application as I imagined it would go. I then took this control flow sketch and drew up a series of rough sketches to show how I wanted the application to navigate from screen to screen and which buttons were on which screens and what those buttons did. Images of these original rough sketches are included in the appendix to demonstrate my design process and its progress.

Click-Through Graphical User Interface Prototype Design

Once I had a series of screens laid out with their components and how those components interacted to allow the user to navigate, I set out to develop the front-end of the application; the graphical user interface or GUI. This was a simple set of nine screens, or "Activities" as they are called in the android platform. I added all the components to each screen from my sketches; various buttons, and text labels, and list components to eventually be populated by my back-end and further functionality. Once I had my GUI created I could really feel out the application and investigate whether there were holes in my design or missing functionality that made it feel unnatural or awkward.

Additional Thoughts

In my original idea for the application, every screen had a "back" button on it to go back to the previous screen. This idea came from my fondness for the back button I enjoy on my Windows Phone, and how I find it quite frustrating that iPhone's don't have explicit hardware back buttons. Once I became more familiar with the android platform, I found that every device actually has a back button on it already, so in favor of this hardware button, the virtual back buttons were removed from my design.

When I approached the drawing board to design my GUI, I was not sure how exactly I wanted the users to view and be able to adjust the status of the song requests. I knew there would be three possible statuses: "accepted" for when a disc jockey decides they will play the song requested, "rejected" for when the disc jockey decides they will not play the requested song, and "created" for a request that has not yet been decided on by a disc jockey. I originally imagined a three-way slider type of button to display this. But this created an issue; if there was a button displaying the status, then the patrons *as well as* the disc jockeys could alter the status, which was not a desired functionality. I instead switched to a simple text read out of the status, combined with a button for accepting and a button for rejecting on the disc jockey side of the application.

Furthermore, there needed to be a device in place to prevent patrons from going into the disc jockey side of the application and messing with all the settings and requests. I decided an authentication process for disc jockeys would be implemented, but not for patrons. I looked into a few different options for implementing this, but ultimately decided to implement my own authentication system instead of using a preexisting one. This allowed me to explore the process of authentication and gave me complete control over the system. My eventual, simple implementation of this authentication was sufficient, in terms of security, because this is not a critical piece of software; there are no lives, economies, public policies or anything of any importance relying on it, so I was free to explore my own solution to authentication. This was done with a simple storing of usernames and passwords in my real-time database and checking for correctness of each, in succession, on login.

Google Play Services

I discussed the use of a back-end server with a peer a few years ago and remember him telling me that when he developed one he used a Google server and used Google Cloud Messaging within Google Play Services to communicate with it. I looked into this as one of my options for designing my back-end server. I found there to be a large amount of somewhat complicated functionality and this being an individual study project, I got a little lost in all of it as I tried to learn it on my own. There were far too many options and routes to accomplish similar tasks it seemed. I looked into a few other options as well, in attempt to find a simpler solution, because my database needs after all were very basic; all it had to do was store a tree structure with strings.

Firebase

I explored a few other options for server hosted databases before I found exactly what I was looking for in Google Firebase. Google Firebase offered a real-time database that would store my data as a tree and hold strings as the values in the tree. There wasn't much more to it than that. I'm sure there is more functionality available, but I didn't get buried in it the way I was when researching other services. The Firebase was simple, easy to use and implement, and had the exciting feature of being a real-time database. This means I did not have to keep refreshing my viewing of database information; when the information was changed, all connected devices would reflect that change in real-time as long as they have proper listeners set up. I was also able to use this database to house my implementation of an authentication system.

Software Design Document

As I was researching the service to use for my application's back-end, I was reminded, by another course I was taking at Cal Poly, that I had mostly skipped a somewhat crucial step in the software development process. I never created a software design document. At this point in time much of the application had been designed, but I still thought it was a necessary step in my process for this project to develop this important document. In an effort to reduce adding duplicate information to this report, the software design document was created in mostly an outline format, as it was mostly for my academic benefit and to aid my design and development.

But I believe this document was too important to be tucked away in the appendix of this report so I have embedded it in the report below.

[Embedded Document Start: Software Design Document]

Introduction

- Purpose

  The purpose of this project is to investigate a mobile application to handle disc jockey requests by patrons in various settings. The mobile application will solve problems relating to this and serve as a testament to my knowledge and skills in the field of computer science.

- Scope

  This project will consist of the research, design, and development of a mobile application on the Android platform. This mobile application will be split into two sides or versions; one for the Disc Jockeys (often called the server) and one for the patrons (often called the client). These two versions will interact with each other offer the application's functionality. The mobile application will not be published to the Google Play store. The project has a single researcher, designer, developer, and tester; that is, the team consists of only Alexander Mitchell.

- Definitions, Acronyms, Abbreviations
  - Definitions
    - Client side/application/version: Refers to the portion of the application that is used by patrons and the general public, all non-disc jockey users. This is the part of the application that makes requests.
    - Server side/application/version: Refers to the portion of the application that is used by disc jockeys, karaoke jockeys, party and event hosts, etc. This is the part of the application that takes requests.
  - Abbreviations
    - DJ: Disc Jockey
    - App: application, as in mobile application

Design Overview

- Description of Problems
  - DJ's have a hard time hearing requests.
  - DJ's generally find taking requests annoying.
  - Patrons do not know if a disc jockey heard their request.
    - Even if the request was heard, patrons don't know if the request will be accepted or rejected.
- Technologies Used
  - Platform

    This project will be created for the Android Mobile platform only.
  - Development Environment

The Android Studio development environment will be used to do all the research as well as the development for this project.

- Testing

  Testing will be done on the Android device emulators available on Android studio as well as physical Android devices. The actual Android devices used will be listed here when they are acquired.

  - Testing Devices Used
    - Various different devices on the Android Studio Device Emulator as previously stated
    - A borrowed Motorola Droid running API 16
      - Note: Project had to be altered to target this older API in order to test on this device; as such it was much less aesthetically pleasing on it.

- System Architecture
  - Back-end
    - Real-time database system hosted on Google servers called Google Firebase.
  - Structure is a tree as follows: '…' indicates repetition of nodes of this depth.
  - ROOT: DJukebox
    - Users
      - <Username1>
        - Bio
          - Name: <DJ Name>
          - Venue: <Venue Name>
          - EndTime: <End Time of Current Session>
          - Paragraph: <Paragraph of biography of DJ>
          - Password: <Password of this user>
        - RequestList
          - Request0
            - Artist: <Request's Artist>
            - Song: <Song Requested>
            - VersionRemix: <Version/Remix of Song>
            - Status: <Accepted | Rejected | Created>
          - Request1
            - …
          - …
      - <Username2>
        - …
      - …
- System Operation
  - Deploys on any Android device running Android version 17 or newer.

User Interface Design

- Client
    - See sketches and images in appendix.
    - Client side referred to as “Patron” in graphical user interface.
- Server
    - See sketches and images in appendix.
    - Client side referred to as “DJ” in graphical user interface.

Data Model and Storage

- Data Objects
    - DJ
        - Object to represent a user, specifically a user, and their profile and associated session
        - Fields:
            - `name`: A string of the name the DJ would like displayed on their session
            - `venue`: A string of the venue the DJ is performing at during the session
            - `endTime`: A string of the time the DJ’s current session ends at
            - `paragraph`: A string of a paragraph to describe the DJ; a biography paragraph
            - `key`: A string of the key of this DJ’s node in the database; its username
            - password: A string of the password for this DJ’s login
        - Defined Values:
            - `NOT_IN_SESSION_VENUE`: A code string to indicate that a DJ is not currently in session to be stored as the venue when a DJ is not in session, used in combination with `NOT_IN_SESSION_ENDTIME`.
            - `NOT_IN_SESSION_ENDTIME`. A code string to indicate that a DJ is not currently in session to be stored as the end time when a DJ is not in session, used in combination with `NOT_IN_SESSION_VENUE`
            - `BIO`: A string to represent the key of the Bio nodes in the database.
            - `END_TIME`: A string to represent the key of the endTime nodes in the database.
            - `NAME`: A string to represent the key of the name nodes in the database.
            - `PASSWORD`: A string to represent the key of the Password nodes in the database.
            - `VENUE`: A string to represent the key of the Venue nodes in the database.

        - `USERS`: A string to represent the key of the Users nodes in the database.
        - `PARAGRAPH`: A string to represent the key of the Paragraph nodes in the database.
    - Song Request:
        - Object to represent a song request and its associated status
        - Fields:
            - artist: A string of the name of the requested song's artist.
            - song: A string of the name of the song requested.
            - versionRemix: A string of the specific version or remix of the song that is being requested.
            - status: An enumerator of the current status of this request; accepted, rejected, or created
            - key: A string of the key of this song request nodes in the database; "request<position in list of requests>".
        - Defined Values
            - `REQUEST_LIST`: A string to represent the key of the RequestList node in the database.
            - `REQUEST`: A string to represent the key, minus the position, of the node of each request in the database.
            - `ARTIST`: A string to represent the key of the Artist nodes in the database.
            - `SONG`: A string to represent the key of the Song nodes in the database.
            - `STATUS`: A string to represent the key of the Status nodes.
            - `VERSION_REMIX`: A string to represent the key of the VersionRemix nodes in the database.
            - `accepted`: A string to represent that accepted status value in the database.
            - `rejected`: A string to represent that rejected status value in the database.
            - `created`: A string to represent that created status value in the database.

References

- *Sams Teach Yourself Android Application Development Fourth Edition*
    - Use this book to learn how to develop Android applications. Use as a reference during development as well.

[Embedded Document End: Software Design Document]

Development Methods

Eclipse

Originally, I began to explore the android platform and develop my project in an integrated development environment (IDE) called Eclipse using a Google plugin for android development. I was constantly running into problems with the way my files were stored and the dependencies of the environment and various updates and configuration issues. This became very frustrating as I spent more time trying to fix my development environment than actually benefiting from using it. Eventually I decided to look for a better development method.

Android Studio

I switched to Android Studio after my experience with Eclipse and my development environment ran smoothly from then on. The book I purchased to help me learn the android platform to give me the necessary skills to complete this project was written to be used alongside the Android Studio IDE, so the switch was doubly beneficial.

The book, *Sams Teach Yourself Android Application Development in 24 Hours*, was split into 24 chapters, or hours. Each one described new features of android and then taught the reader how to implement them in a single android mobile application. I began going through this book, chapter by chapter, creating the applications it described as I learned how to navigate Android Studio and develop mobile applications. After about a fifth of the book I felt I was ready to implement the GUI I had designed. So I took my design mock-up sketches and created an android activity for each one. Then I implemented the navigation of the application by connecting the buttons on the GUI to switch between the activities in the way I had described on my sketches. Some of the buttons I could not implement yet because they relied on the data that would be stored in my server-based database, which was yet to be developed at this time.

After finishing the GUI, this is when I revisited my design phase to find how I wanted to design and implement the back-end, as I described during my previous discussion of my design methods. After I found the particular service I wanted to use for my back-end server-hosted database I went back to my reference book. This time instead of going through chapter by chapter, I skipped ahead and found the chapters that showed how to implement things I need to know how to implement for my project. I then later used these small applications as references when implementing the features contained in them in my actual project application.

During my development, because I was so new to the platform, I did not want to develop in large chunks and then test and debug for long periods of time. Instead I split my development into small tasks that could be accomplished rather quickly once the process to implement them was understood, and specifically how the task could be accomplish on the android platform was understood. After each small task was completed, such as “make sure a new user cannot choose a username that is already in the database”, I would build the project, deploy it to an emulator running a virtual android device and test this new feature or task as well as do regression testing on all the previously implemented tasks to make sure nothing previously implemented was broken by the new development.

Some of these tasks had interesting solutions that I did not foresee needing during my design, and are worth noting. One of the situations I thought was an issue was that patrons don’t need to load up a disc jockey’s page if that disc jockey is not in session. I added a conditional to

that click listener to check if the disc jockey was in session or not, using the two string codes described in the software design document that indicate a disc jockey is not in session when stored in the disc jockey's Venue and EndTime node in the database. If the disc jockey is not in session, then the application just displays a little message saying so. This message is called a toast notification in android. The same process was implemented on the disc jockey side of the application when a disc jockey attempts to resume a session when they are not currently in session. A similar process was also implemented to indicate to the new user that the username they have selected is not available because it is already stored in the database and being used by another user. Toast notifications were also utilized to communicate to the user that they have entered the wrong username when attempting to login and if they have entered the wrong password. These notifications were part of the authorization system I implemented.

Project Conclusion

Application Functionality

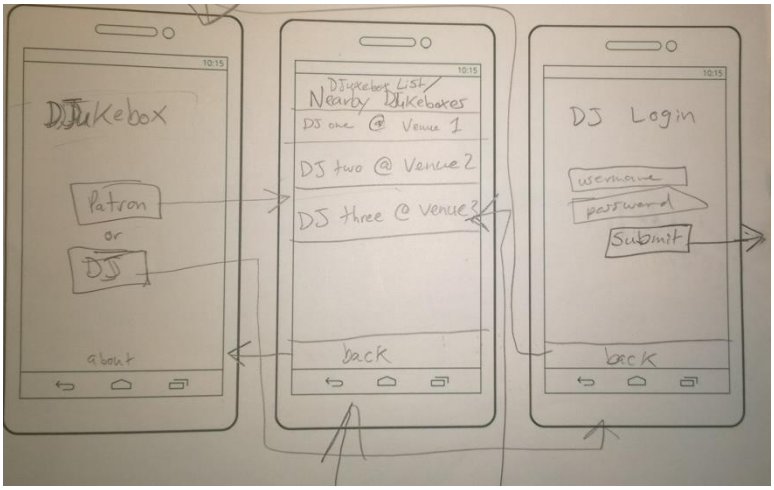
The purpose of the project was to explore mobile application development and specifically to create an android application that handles requests made to disc jockeys and solves problems related to this. The purpose was also to meet the requirements of a senior project and to learn about new software platforms and tools and then demonstrate that learning. I can confidently say that these tasks were accomplished. The application I developed works properly. Multiple instances of the application can run at the same time, interacting with the database in real-time just as they were designed, imagined and intended to.

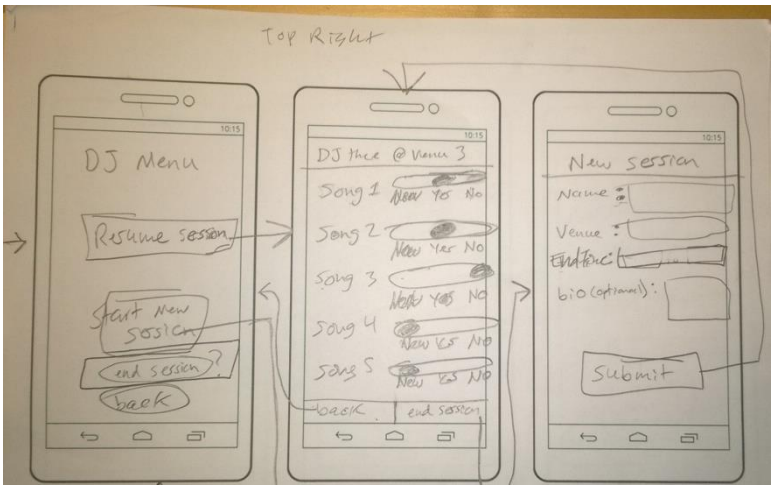
Application Desire and Academic Achievement

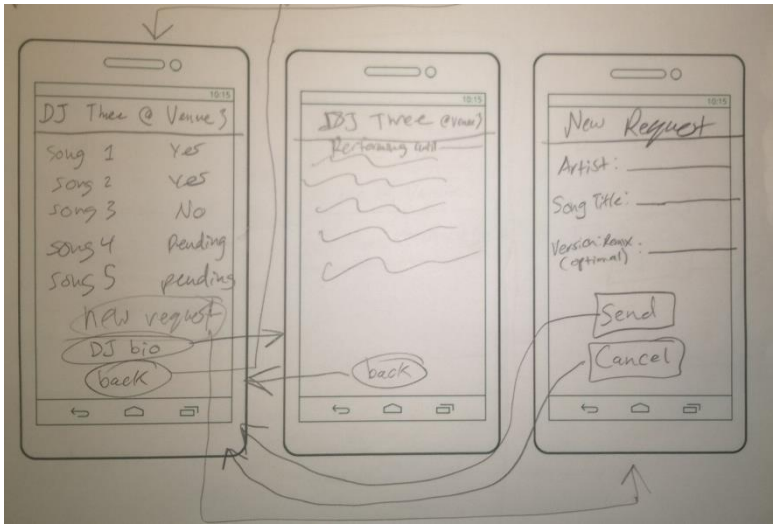
Unfortunately, my research strongly suggests that there is no market for this application. However interesting I found it to research, design, and develop, my research suggests there are few users beyond me that would use this application or download it. The application perfectly serves its purpose in academia, but academia is where it stops. As such, minimal time was put into the aesthetics of the application, and it is mostly functional; I did not feel it was necessary to paint a ship that would never sail. Furthermore, if the application was not going to have many users, then a wish-list feature of geo-locating users to only show them sessions that are close by was not implemented. The application was not published to the Google Play Store, and the original intention to enter it into the Innovation Quest competition was abandoned. Nevertheless, I feel this was a great experience learning new applications programming interfaces (APIs) and connecting different technologies and to take a project from idea to product through the different stages of the software lifecycle.

Appendix

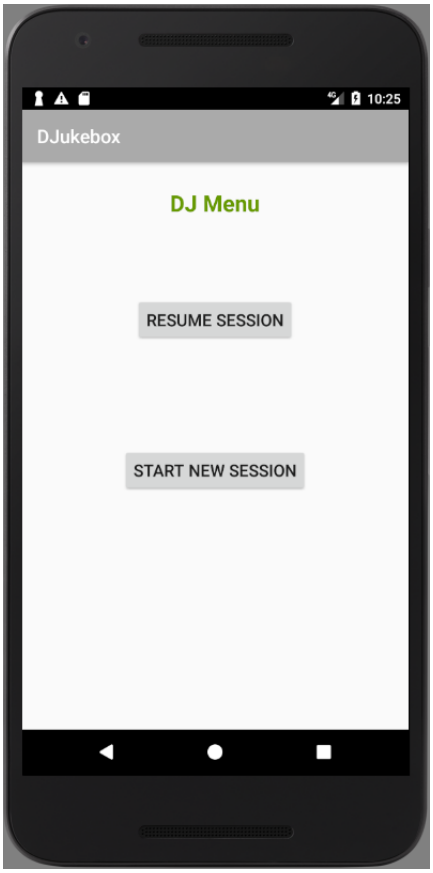
Rough Sketches of the Graphical User Interface

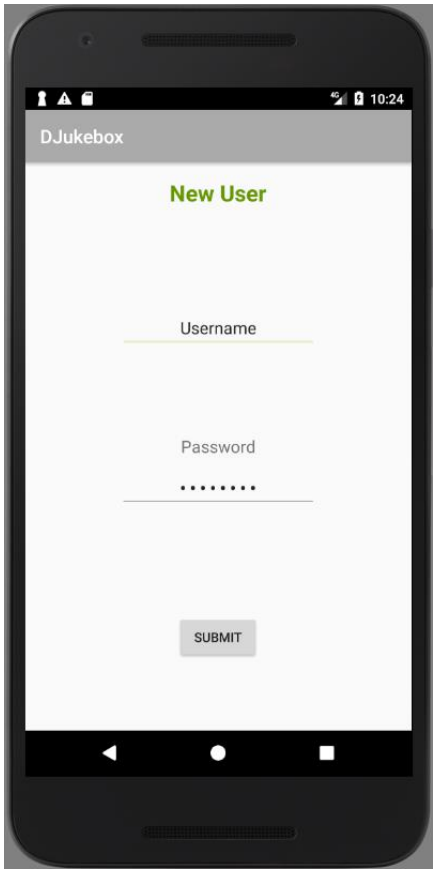


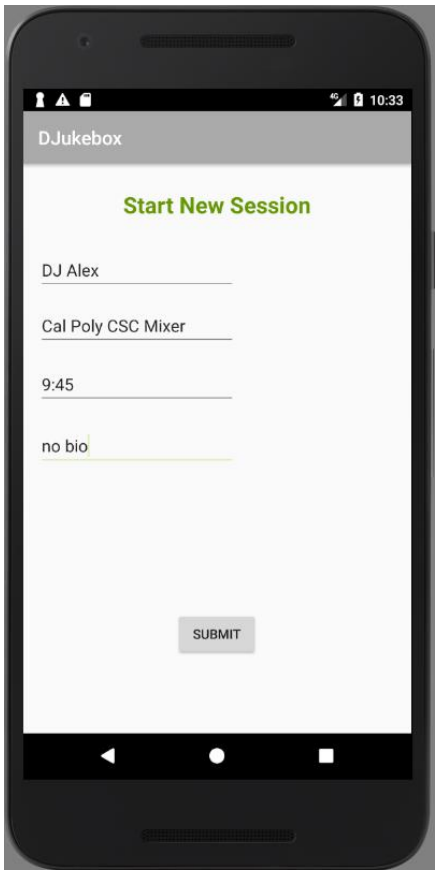


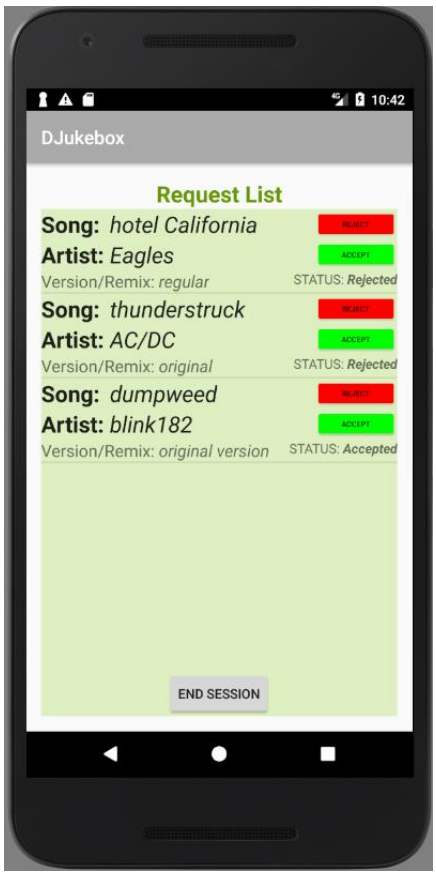


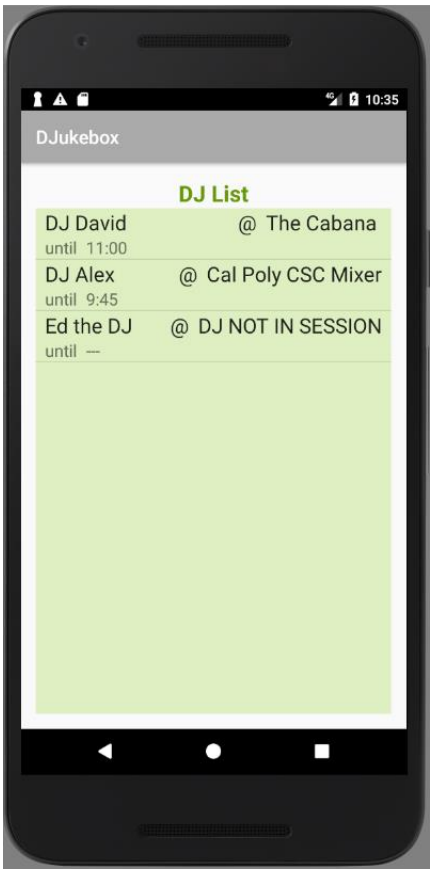
Images of the Graphical User Interface

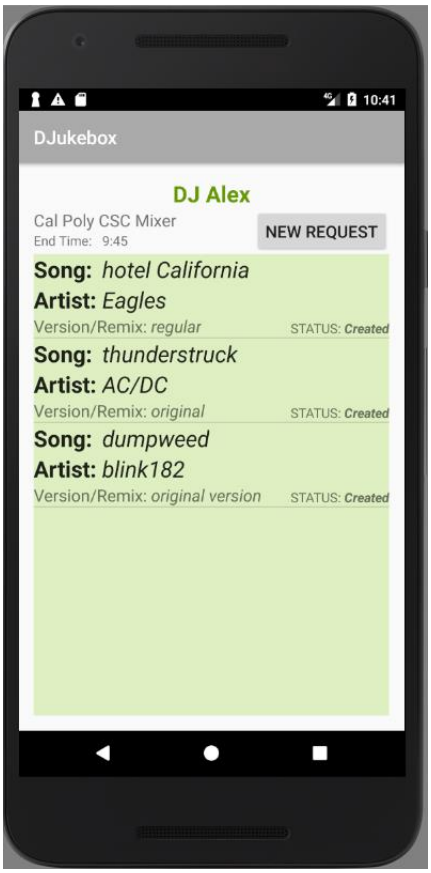


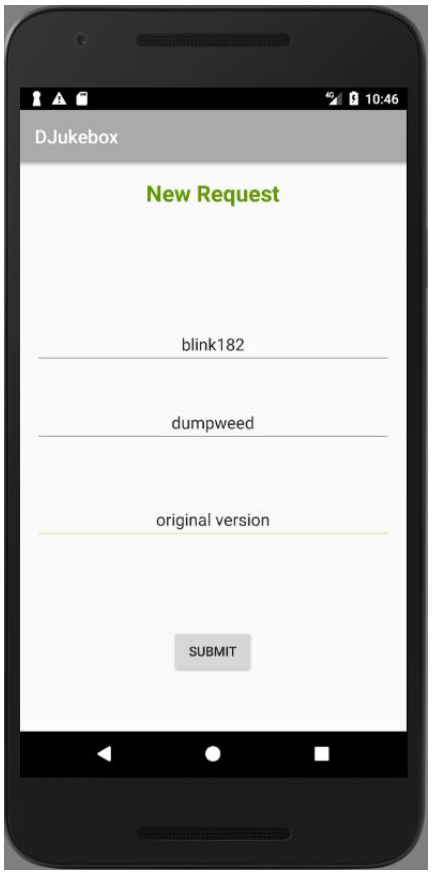



10:33
DJukebox
Start New Session
DJ Alex
Cal Poly CSC Mixer
9:45
no bio
SUBMIT


10:42
DJukebox
Request List
Song: hotel California
Artist: Eagles
Version/Remix: regular
STATUS: Rejected
Song: thunderstruck
Artist: AC/DC
Version/Remix: original
STATUS: Rejected
Song: dumpweed
Artist: blink182
Version/Remix: original version
STATUS: Accepted
END SESSION


10:35
DJukebox
DJ List
DJ David @ The Cabana
until 11:00
DJ Alex @ Cal Poly CSC Mixer
until 9:45
Ed the DJ @ DJ NOT IN SESSION
until ---


10:41
DJukebox
DJ Alex
Cal Poly CSC Mixer
End Time: 9:45
NEW REQUEST
Song: hotel California
Artist: Eagles
Version/Remix: regular
STATUS: Created
Song: thunderstruck
Artist: AC/DC
Version/Remix: original
STATUS: Created
Song: dumpweed
Artist: blink182
Version/Remix: original version
STATUS: Created


10:46
DJukebox
New Request
blink182
dumpweed
original version
SUBMIT

DJ Questionnaire

1. Please rate taking requests from patrons when you are working as a DJ, on a scale from 1-10, on easy (0) vs. difficult (10); with 5 being neutral. Please elaborate if you will.

   0 1 2 3 4 5 6 7 8 9 10

   _____
   _____
   _____

2. Please rate taking requests from patrons when you are working as a DJ, on a scale from 1-10, on annoying (0) vs. pleasant (10); with 5 being neutral. Please elaborate if you will.

   0 1 2 3 4 5 6 7 8 9 10

   _____
   _____
   _____

3. How satisfying is it when you fulfill a request made by a patron, on a scale from 1-10; 10=most

   1 2 3 4 5 6 7 8 9 10

4. What features would you like to see on a mobile app that is based on making requests simpler?
   a. For the DJ?

      _____
      _____
      _____

   b. For the requesters?

      _____
      _____
      _____

5. What platform(s) would you like to see an app of this sort be available? (check all that apply)
   a. iOS (iPhone/iPad)
   b. Android
   c. Windows Phone
   d. Blackberry
   e. Other, please specify _____
   f. Web (internet browser) application
6. What input do you have on the interface on your side of the application? How would you like it to navigate from screen to screen and function to function? Please elaborate as much as you are willing to, keeping in mind that if you ask for it, I may implement it.

   _____
   _____
   _____
   _____
   _____
   _____
   _____

   You may draw picture on the back if you would like, if it will help explain your thoughts.

Patron Questionnaire

1. Would you use a mobile application to communicate a request to a DJ at a venue or party? Please elaborate.

   Yes No

_____
_____
_____

2. Do you know of any mobile application that exists already that helps patrons to make requests to DJs?

   Yes (if so, please identify it _____) No

3. How important is it to you as a patron to know if your request to a DJ has been heard/received on a scale from 1-5, 5 being very important?

   1 2 3 4 5

4. How important is it to you as a patron to know if your request to a DJ has been rejected or accepted, on a scale from 1-5, 5 being very important.

   1 2 3 4 5

5. What features would you like to have a patron viewing DJs playing at different venues, other than making requests? What specific information about the DJs would you like to see? Would you like to see a queue (ask if you do not know what this is) of the requests that have been made and approved/rejected? Please elaborate as much as you can.

   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____
   _____

6. Please think about what this application could possibly provide you as a user, free of charge, and provide any other requests or input that you would like to, keeping in mind that if you ask for it, it may be implemented and considered very important from a business standpoint.

   _____
   _____
   _____
   _____
   _____
   _____

Informed Consent Form

Informed Consent Form

INFORMED CONSENT TO PARTICIPATE IN A RESEARCH PROJECT, *Alexander Mitchell's Senior Project*

A research project on *DJ-related mobile applications* is being conducted by *Alexander Mitchell* in the Department of computer science at Cal Poly, San Luis Obispo. The purpose of the study is to gather input from potential users of the application that is being developed.

You are being asked to take part in this study by completing the attached/enclosed questionnaire(s). Your participation will take approximately 10 minutes. Please be aware that you are not required to participate in this research and you may discontinue your participation at any time without penalty. You may also omit any items on the questionnaire(s) you prefer not to answer.

The possible risks associated with participation in this study include minor social irritation. If you should experience any negative emotions, please be aware that you may the Cal Poly Health Center at *805-756-5298* for assistance.

Your responses will be provided anonymously to protect your privacy. Potential benefits associated with the study include influencing the potential features and functionality of my mobile application, which will eventually be free to the public to use.

If you have questions regarding this study or would like to be informed of the results when the study is completed, please feel free to contact Alexander Mitchell at *206-947-8202*. If you have questions or concerns regarding the manner in which the study is conducted, you may contact Dr. Steve Davis, Chair of the Cal Poly Human Subjects Committee, at (805) 756-2754, sdavis@calpoly.edu, or Dr. Dean Wendt, Interim Dean of Research, at (805) 756-1508, dwendt@calpoly.edu.

If you agree to voluntarily participate in this research project as described, please indicate your agreement by completing and returning the attached questionnaire. Please retain this consent cover form for your reference, and thank you for your participation in this research.

_____ _____

Signature of Participant Date

DJ Questionnaire Elaborations

Question 1. Elaborate on the difficulty of taking requests from patrons when you are working as a DJ:

"I tell them to have a name of artist and Song. When they don't know the name, then I ask them for the lyrics or to sing the beat."

"Loud music and no order to people's requests makes it extremely difficult especially in larger club format"

Question 2: Elaborate on whether taking requests when working as a DJ is annoying or pleasant:

"You get many types of people wanting a song and many times you just have to say no especially when the patron is intoxicated"

"Most times people are nasty and rude if you say no as if they are the only one at the party"

Question 4: Features for a DJ:

"To know what version, song name, and artist name"

"incorporate into DJ software, already-played button, way to clear fulfilled requests, profile, DJ profile for booking"

"tips, genres, tempos"

Question 4: Features for a Requester:

"To send their song that they have on their phone and add it to [the] playlist. Also to see if DJ has song when they request it"

"Artist, song, remix. ETA on request fulfill

Question 6: Interface input:

"Most open format DJs play very quickly, 1:45-2:30 per song, something easier to use instead of features."

"[pop-up window on DJ software with artist, song name, and version]"

Patron Questionnaire Elaborations

Question 1: Would you use a mobile app to request a song from a DJ?

"This would eliminate the need to shout over the music or fight through a crowd"
"Yes, it would be a fast way to give your requests without trying to yell to the DJ"
"Sure if I wanted to request [a song]"
"This would be a great way to communicate directly with a DJ without much effort. People are lazy and this is a great idea"
"I think it would be beneficial at a party. I know countless times I have been screaming at the DJ"
"Music can make or break a party. I would love to put in requests"
"Because it would help the DJ to better organize song requests"
"Yes, there are many times we want to give requests but the DJ can never hear at functions"
"Hard to yell at them, don't want to bug them"

Question 5: What features, what do you want to see?

"A queue, what has been approved/rejected, and possibly what genre(s) are being played that night"
"I [want to] see who is spinning where and I [want to] see booking/contact info. Also if my request will be played, I want to know that. Sometimes I [want to] know why they won't play it, or when they will."
"I would like to see bar deals, I'd like the request to have the song choice be interactive (like when you type in songs to Pandora) and you can select them. I would also like to see the queue and even able to see how late DJ goes. I'd also kind of like the possibility of rating the DJ. If there are photo booths/etc. I would like to see hours of operation/a map of the event."
"I would like to choose any song from ITUNES and youtube. So both official and unofficial songs. Also I would like to see maybe a music suggestion category or recommended songs category."
"Type of music he usually plays. His accept rate of songs that people put in. Yes, a queue would be awesome. Comments to the song?? Where all the parties are located, who's DJing, theme. wait time to get into venue."
"The app could suggest hot songs, or songs that are popular at this club. Videos of the DJ's other performances. Bios about the DJ and what music he specializes in"
"A queue would be a great idea and would give people a direct idea as to whether people's requests are actually viewed. It would be cool to have a portion of the app where you can thumbs up and thumbs down songs in order to influence the DJ's choice. If there's a way to make Androids and Apple products capable of performing the same functions in regards to the app."
"A good DJ wouldn't want to take requests. Kinda a pride thing. The app hooked up with his music so as a listener I could find out the song he is playing on the app if I had no idea what it was. Shows what general types of songs he plays"
"Queue is a must. It's a good way to see what's coming up next and also possibly upvoting/downvoting songs. Have different tabs with different genres of music to queue. Autofill songs/artists to search. Have a tab with most played or top songs"

Question 6: Other requests:

"An ability to see what songs have been played or are currently playing and to "like" or somehow express interest, for both the DJ's sake as well as the ability for you to download or revisit the song later"

"It just provides an ease of way to send requests without the hassle of trying to inform the DJ up at the booth. Could be a fun way to play music at a party too."

"Great idea for application in this world of upcoming technology."

"If the app showed the album cover and other songs by that artist"

"introducing me to new music. Price of app??"

"Allow the user to save/see the song the DJ just played so they can fin it/listen to it again".

"Like airdrop on the iphone, if there is no service you can send it over wifi/Bluetooth"

"Definitely free! This could be cool for karaoke"