Music Info Retrieval and Pain Management

MQP W1W - 181

Student Authors:

Jesse Gaulin

Andrew Nemeth

Liam Yoo

Faculty Advisors:

Professor Wilson Wong

Professor Vincent Manzo

Major Qualifying Project Report:

Submitted to Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

Computer Science

Worcester Polytechnic Institute

United States

2 March 2018

CONTENTS

Abstract— **Overuse of opioids for the treatment of chronic pain has led to a national epidemic of overdose and addiction. Though music is already known to cause pain relief, the proposed Music-induced Analgesia Genome Study (IMAGS) is interested in discovering which aspects of music reduce chronic pain the most. For this study, the team developed a software system with two components: a phone application that tracks participants' Spotify song selection while they rate their pain, and a web application to allow researchers to view and aggregate gathered data. This software system enables researchers to identify what types of music could be used to supplement or replace opioid pain relief.**

I. INTRODUCTION

Death from opioid overdose is an increasingly large problem in America. Sixty-eight percent of the 72,000 cases of drug overdose in 2017 can be attributed to some form of opioid. The number of overdoses from synthetic opioids has drastically increased from fewer than 5,000 cases in 2013 to 30,000 cases in 2017. Reducing the prescription of opioids by identifying alternatives to pain reduction could help to reduce the scale of this epidemic (National Institute on Drug Abuse, 2018). Music has already been found to alleviate pain, but researchers are interested in identifying what aspects of music reduce pain most strongly. The Music-induced Analgesia Genome Study (IMAGS) is a proposed research plan with the goal of testing the correlation between self-selected musical listening and a decreased perception of pain. To this end, the study plans to record the self reported pain ratings of 100 participants while they listen to self selected music (Manzo). Our team created a software system to enable both the participants to report pain levels and the researchers to view participant data. To allow participants to report pain levels, our team developed a phone application which allows participant to report their pain via a digital slider while they listen to music played through Spotify. The slider allows participants to move a dot along a horizontal line marked with faces indicating pain or an absence of pain. Participants can use this application when experiencing a sudden increase in pain, called breakthrough pain, and considering using medication. Researchers can use a web application to access and visualize data gathered by the phone application. Design and development of the two applications took place over the course of 8 months as part of Worcester Polytechnic Institute's Major Qualifying Project. The team decided to employ the React Native stack to develop the patient pain-reporting phone application. This technology stack allows for easy deployment to Android, IOS, and Web: the three target platforms for this application. Furthermore, this stack employs Firebase, a database which not only stores data but also handles authentication and real-time data updating using Google APIs. Participants will need to create a Spotify account or provide a preexisting one to the application. The application uses Spotify's API to track and store participant music selection during a session. The team also developed a React web application for researchers to organize groups of participants and visualize research data. The web application synchronizes with the Firebase database in real-time and provides detailed graphs of participant's changes in pain over time. The graph not only marks what music participants were listening to at certain times but also allows researchers to view the following attributes of the music by using Spotify's song attribute API:

- Danceability
- Energy
- Key
- Loudness
- Mode
- Speechiness
- Acousticness
- Instrumentalness

- Liveness
- Valence
- Tempo
- Duration(ms)

II. Research

A. IMAGS

The Music-induced Analgesia Genome Study (IMAGS) is a proposed research plan with the goal of testing the correlation between self-selected musical listening and a decreased perception of pain. The study has 2 stated aims. First, it aims to develop a mobile application that participants will use to both listen to self-selected music and continuously report pain levels. Second, the study aims to determine if listening to self selected music can reduce pain on a self reported scale of zero to ten. The study hypothesizes that there will be a significant decrease in self reported pain after ten sessions of listening and that there will be a significant decrease in the use of breakthrough pain medication following listening sessions. This study plans to record the pain ratings of 100 participants while they listen to self-selected music. Each participant will spend two weeks actively participating in the study: launching the application and listening to music whenever they experience breakthrough pain. In addition to listening to music and reporting pain during a session, the participant will report whether they intend to use breakthrough medication after the listening session. Potential participants will be identified through the NeuroMusculoskeletal Institute (NMI) located on the Rowan University School of Osteopathic Medicine Campus (Manzo). In addition to developing a mobile application to record participant pain levels, a second application was planned to allow researchers to view participants' reported pain levels and the aspects of the music they were listening to. Music will be automatically analyzed by the application in 20 second segments for various quantitative and qualitative attributes including genre, tempo, dynamic, harmony, timbre, meter, and lyrical.

B. Opioid Analgesics and Overdose

Chronic pain is pain which lasts longer than three months, and leads to decreased well-being and functional capacity. It is considered one of the most prevalent and debilitating medical conditions; more than 30% of Americans experience some form of acute or chronic pain. The urgency of treating patients experiencing chronic pain combined with the effectiveness of opioid analgesics (pain relief) has led to an overreliance on opioid medications in the United States (Nora, 2016). The widespread use of opioids as a treatment to chronic pain has resulted in a national epidemic of opioid overdose deaths and addictions. According to the Center for Disease Control and Prevention, the drug overdose death rate has more than doubled, increasing from 6.2 per 100,000 persons in 2000 to 14.7 per 100,000 in 2014. The age-adjusted rate of overdose deaths then saw another significant increase in 2016 to 19.8 per 100,000. In 2014, 61% of drug overdoses involved some form of opioid. The number of drug overdose deaths involving opioids increased by 14% from 2000 to 2014 (CDC, 2016).

C. Music Analgesia Research

Music therapy is one potential alternative to opioids for pain reduction. Research has shown that listening to music can reduce the perception of pain. One study on the effect of music therapy on postoperative pain management in gynecological patients analyzed the results of seven postoperative music analgesia studies. Of the seven studies analyzed, five of them found that patients experienced significantly less postoperative pain when listening to music (CDC, 2017). Despite five decades of extensive research on the subject, the exact mechanics of music analgesia are still not known. Many researchers suggest music reduces pain through a number of separate operating mechanisms, such as attention, distraction, and changes in affect and arousal levels. Studying music analgesia is difficult in part because pain is a complex, private, and subjective experience (Garcia, 2016). Furthermore, the methods by which researchers study music analgesia vary significantly as well. Music could be preselected by researchers, self-selected by patients, or a set of music could be selected by researchers and presented to patients to select from. Though self-selected stimulus is preferential (and is the chosen method for IMAGS), it makes the study less controlled. Researchers are seeking methods of understanding what aspects of music provoke an analgesic response to create more consistent sets of selected music. However, certain songs may also have additional, unique effects on participants who have strong memories associated with the song independent of the traits of the music itself. These patient-specific associations often cause the most profound psychological, emotional, and physiological response. Music can have differing effects if listened to either actively (as the primary activity of the participant) or passively (as background music while the participant does other activities)(Garcia, 2016). A 2006 review of music analgesia research claims that the variety of music administering methods used by researchers suggests that they have not yet found a standardized, consistently successful method of administering music to study its pain reduction effects (Cepeda, 2006). Researchers employ a variety of techniques to query and quantify participants for pain levels. Blood pressure, pulse rates, and scores on POMs Bi and VAS-U and VAS-I are considered quantitative methods of tracking the effects of pain. In addition to quantitative measures, interviews can be conducted with participants to attempt to qualitatively define their pain. One study simply asked participants how much pain they were in and their opinion on whether music was able to reduce it (Garcia, 2016). IMAGS is a qualitative study of pain because it allows participants to self report pain levels using a subjective zero to ten slider (Manzo).

D. *Aspects of Music*

IMAGS aims to find out the pain reducing effects of different aspects of music, such as volume, tempo, and pitch. The volume and pitch of a song can be derived in the physical properties of amplitude and frequency, respectively. Tempo is the temporal organization of a song and is measured in beats per minute. Instrumentation is an aspect of music that deals with the tools or voices used to create a piece. Finally, melody and harmony are the sequential and plural aspects of music. They refer to the sequential organization of notes in a song.

E. *Phone Applications for Querying Pain*

Phone applications are frequently used in medical studies to gather patient information over an extended period of time. A survey inquiring about receptiveness to using phones to report medical information was sent to 100 patients as part of a study to determine how technology could be used to increase communication between patients and researchers (Shah, 2014). Eighty-seven percent of patients reported a comfort level of five or more on a zero to ten Likert scale where zero was not comfortable and ten was extremely comfortable. Patients were also comfortable using texting (81%) and emailing (77%) to communicate medical information but not social media (40%) (Shah, 2014). One pain study created a two-day consensus conference where various medical experts discussed best practices for creating a phone application to query patient's pain levels and suggest pain-reduction methods(Jibb, 2014). The members of the conference considered the application design from both a medical and technical perspective by discussing ideal pain gathering techniques and system design requirements. The conference concluded that the application should not only ask the participant for daily, scheduled, long-form assessments of pain but also allow the participant to complete a short-form pain assessment whenever they chose. After reporting pain a user will be prompted with several potential remedies to their reported pain and select one they play to try. An hour after the user has selected a pain relief method, the application will ask the user how they believe the suggested pain relief method affected their pain levels. This study suggests that asking participants for their pain levels only before and after pain-reduction treatment, not during, may be the best method of gathering self-reported pain levels and pain relief data (Jibb, 2014). However, the IMAGS study is interested in identifying the pain reduction effects aspects of specific songs. It would likely be difficult for participants to listen to many songs and recall their specific pain during each song, so the IMAGS application queries participants for pain continuously as they listen to music.

F. Music Provider

The team decided to use Spotify as the application's music provider. Spotify is the second most widely used music provider with 47.7 million users as of March 2018. Though Apple Music is marginally more popular with 49.5 million users, it does not offer an API which provides musical attributes of its music (Most popular music, 2018). The Spotify music attribute API contains the most information about the attributes of each song, including:

- Danceability
- Energy
- Key
- Loudness
- Mode
- Speechiness
- Acousticness
- Instrumentalness
- Liveness
- Valence
- Tempo
- Duration(ms)

Characteristics such as danceability, speechiness, and acousticness could be especially useful for analyzing their effects on pain. Loudness corresponds to the volume of a song, one of the important physical characteristics of music described earlier in this proposal. Key refers to the musical key the song is in, based on the pitch of the notes in the song. Tempo corresponds to the beats per minute (BPM) of the song. Valence refers to the overall mood of the song. Low valence refers to songs that sound more negative, or sad, while high valence refers to songs that sound more positive, or happy (Spotify for Developers, 2018). Though many music services other than Spotify have an API, most of them offer little to no analysis of song attributes. For example, Soundcloud only offers beats per minute and song genre. Though Spotify does not capture all of the attributes of music the IMAGS study wishes to have such as genre, dynamic, harmony, timbre, meter, it is the current best fit.

III. Development Environment

A. Agile & Scrum

The software development methodology the team picked was Scrum, following the Agile principles of development. Agile methodology promotes rapid and flexible software development by prioritizing action over planning. Before and during development, the team gathers user stories from the customer to create a list of items to develop and deliver. The

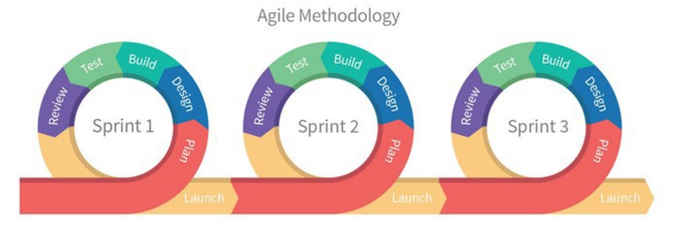


Fig. 1. Agile Methodologies

development of a large piece of software is broken into one to four week sprints. In each sprint the software development team picks several user stories to develop and integrate into the application. Rather than spending extended time (one or more months) working to develop the entire product at once, the product is incrementally expanded every sprint (Mahalakshmi, 2013). When using Scrum methodology, teams hold daily, short, stand-up meetings called daily scrums where team members report their progress, plans, and roadblocks. Teams have two roles in addition to software developers: a product owner and a scrum master. The product owner is responsible for representing the customer's interests and ensures that the development team is working efficiently to achieve the customer's desired deliverables. The scrum master facilitates the daily scrums and works to remove remove distractions and roadblocks from developers.

The Waterfall model is another method for software development, but it has several drawbacks in the context of the team's project. Waterfall methodology requires that every step of the project must be fully completed in sequential order, as can be seen in the figure above (Mahalakshmi, 2013).

This method of software development was not used for several reasons. First, waterfall methodology does not easily allow new features to be added to the product in the middle of development: to add a new feature, the team must go back to the requirements gathering and design phase. Second, waterfall projects have a predefined size and scope: once a set of features are decided on in the requirements phase, they must be designed, developed, tested, verified, and maintained. If the project runs out of resources or time before all phases are complete, there is no final product. If our team used waterfall methodology and agreed to implement too many requirements for the time allotted for development (about 3 months), then the team would have no product to deliver by the end of the MQP. The team chose to employ an Agile methodology because it ensures there will be a product by the end of development. Furthermore, Agile allows for new features to be incorporated into the project as the needs of the MQP change.

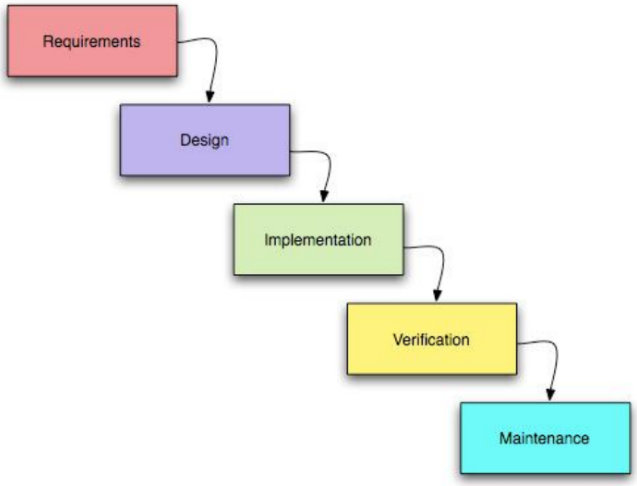


Fig. 2. Waterfall Methodologies

B. Project Management

The team decided to use Trello for project management software. This is because it is one of the simplest of competing project management software. It provides an interface that is easy to read and navigate because it allows color coding of tickets on the board. Its minimalist design also makes it easier to add or remove tickets from the board than other software. While alternatives software such as Github projects allowed for more advanced features like linking directly to issues, the team decided it would be more helpful to have a simple solution to track project management.

C. IDE

The team had to use a few different IDEs because we were supporting multiple platforms, which each have their own IDEs. For Android, the only IDE available is Android Studio so the team decided to use that. Similarly, for iOS the only IDE provided is Xcode. For the web application and writing the JavaScript for the React Native application the team decided to use Webstorm, as we had already used one of JetBrains' other IDEs, Android Studio, and switching between them was easiest.

D. Communication

The team planned on using Slack for our communication software, however decided not to as the team was co-located - everyone lived in the same residence - and preferred face-to-face communication. The team was able to meet and work on the project at home which made for much more efficient communication than having to wait for replies and schedule meeting times over Slack.

E. Technology Stack

The main function of the team's participant pain reporting application is to report pain levels and listen to music on the same device. This allows participants to easily switch between selecting music in Spotify and rating their pain in our application. Because many participants use Spotify on their smartphones, the team initially decided to use a mobile app for participant pain reporting. However, in order to allow as many participants as possible to use our application, the system needed to support participants on both Android and iOS. For our researchers' application the team decided to use a web application because it is cross platform and can easily display and visualize data sets, such as session data. The software system needs to target multiple platforms and synchronize data across them. To this end, the team decided to use a client-server model; clients (such as participants' phone applications and researcher web applications) all connect to a centralized server to get information. In addition to serving and synchronizing data across all clients, the server is able to transform and aggregate data and resend it back to be viewed by researchers. By pulling all of this logic into the server, our client's applications became thinner and easier to develop. In order to minimize needless development, the team decided to use Firebase: a real-time back end platform created by Google. Firebase removes the need to develop a server and database separately. Instead, Firebase can both run cloud functions as a server and store JSON data as a database. Deploying functionality to Firebase is low-maintenance and more secure than client-side code. Furthermore, Firebase offers a cloud-hosted NoSQL database. The database is document-oriented and uses key-value storing in the form of JSON objects. The data the team is storing is non-relational so there would be little benefit from implementing a relational database such as SQL. As a real-time database, Firebase synchronizes JSON data across all clients the moment it is changed, allowing researchers to view session data the moment it is created. Firebase is also optimized for offline use; all client devices (such as a researcher's desktop) will store a cache of research data, allowing researchers to view session information even when offline. Finally, all Firebase functionality is well secured, allowing the team to focus effort on app functionality (Cloud Functions for Firebase, 2018). For our participant's mobile applications, the team decided to use React Native as it would allow us to keep the same programming language as our backend and build the application once with the ability to deploy it to Android and iOS. This is much better than the alternative where the team would have to program the entire mobile application and build the user interfaces in Android, then redo all of the work again with a new programming language in order to make the application on iOS. Additionally, because the code for the iOS build and the Android build will be the same, it reduces the required testing effort in half. Yet another benefit in using React Native is that it removes the operating system dependency of having a Mac, allowing all of the team members to work on all aspects of the software system regardless of their development computer. Another feature of React Native is the ability to use native operating system calls, which gives the team access to special phone GUI features, such as sliders as a notification drop down or Android floating widget, seen below (https://www.spaceotechnologies.com/Android-floating-widget-tutorial/).

For the researchers' client, the team decided to use React, as it is one of the most popular web application frameworks and would allow us to reuse some of our modules from our mobile React Native application. Additionally, by keeping both our client applications in React the team would be able to seamlessly switch between developing in each language without even changing IDEs. React's modular and state-based system allows researchers to easily navigate the website and view the data in the desired way.

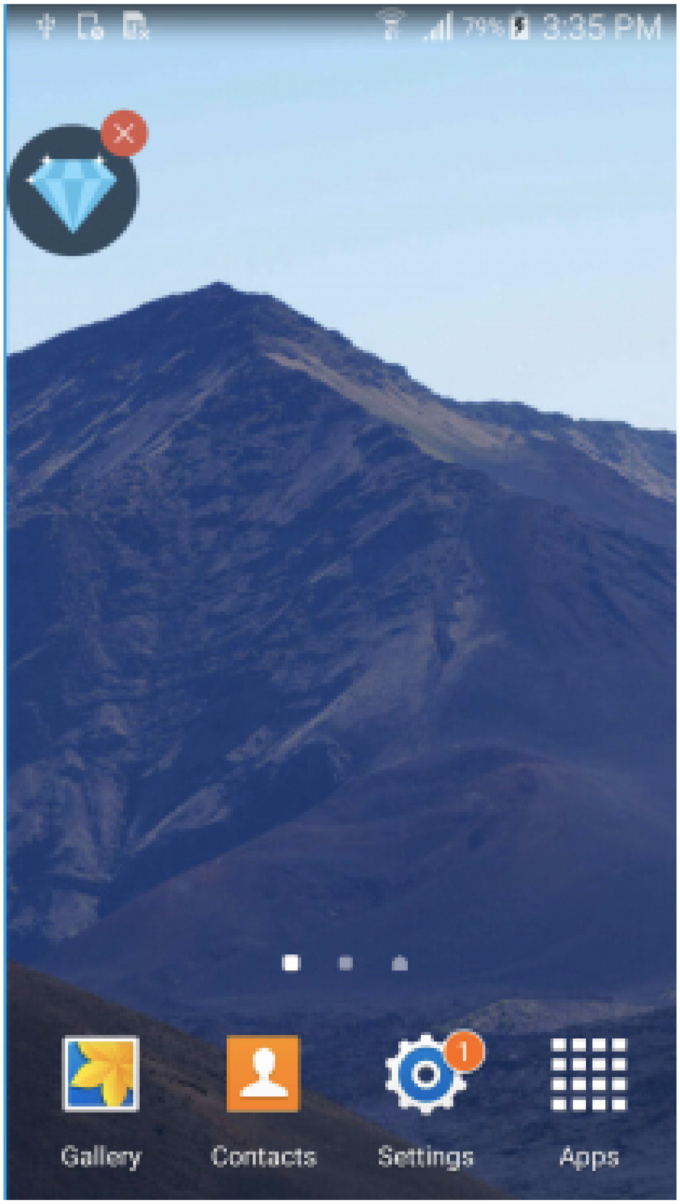


Fig. 3. Floating Widget Displaying a Diamond

IV. Requirements

A. Functional Requirements

The team gathered requirements from the proposal of the previous student research group to approach the problem and from discussions with the advisors of this project.

- System must include a button to start a listening session.
- System must be able to connect and login to Spotify.
- System must continuously prompt for user's pain with a slider.
- System must record all changes in pain level during a listening session
- System must include a prompt asking whether the user will take their medication after the session.
- System must include a button to end a listening session.
- System must include a web application that displays data for researchers.

B. *Non-Functional Requirements*

- Intuitive user interface needs to be usable by non-proficient phone users.
- No significant latency (<1s) in frequent user actions that impede usability.
- Slider is always visually responsive during a session.
- The application can be used by end users without training or help.
- Only researchers can view participant's session data; participants can not view each others' session data

C. *Customer Success Factors*

The team identified which requirements were critical to the success of the project and compiled a list of Customer Success Factors (CSFs). CSFs are high-level goals that are critical for a business to meet in order to satisfy the customer's requirements.

- The ability to search for songs from the application during a session.
- Full media control during a session (pause, play, next/previous song)
- The ability for participants to continuously report pain during a session via a slider.
- Persist session song data and pain rating data in a database.
- The ability for researchers to access single session data in a visually intuitive way.
- The ability for researchers to access aggregated session data in a visually intuitive way.
- The ability for researchers to access raw data files of all information collected by the app.

D. *Epics*

- Create Participants mobile application front-end
- Create Researchers web application front-end
- Create Firebase back-end
- Connect mobile application to back-end
- Connect web application to back-end

E. *User Stories*

- As a participant, I can enter my Spotify credentials and my identification, so that researchers can record the music I listen to.
- As a participant, I have a profile with static information such as name, and dynamic information such as blood pressure, so that researchers know the information about me.
- As a participant, I can start a music session, so that I can try to substitute my breakthrough medication.
- As a participant, I can continuously rate my pain, so that I can signify if the music helps.
- As a participant, I have to rate my initial pain first, so that the researchers how much pain I felt before listening to music.
- As a participant, I can stop a music session, so that I can signify if I took my breakthrough medication.
- As a researcher, I can create participant profiles and enter their information, so that the participant does not have to.
- As a researcher, I can invite a participant to the application using the created profile, so that the participant's information is already entered when they log in.
- As a researcher, I can assign existing patients to groups and edit existing groups, so that I can aggregate data.
- As a researcher, I can search for and view a single session's reported data, so that I can analyze the effects of the music.
- As a researcher, I can search for and view a group's aggregated session data, so that I can analyze the effects of the music.

- As a researcher, I can view the characteristics of a song, so that I can correlate the characteristics to the effects of the song.
- As a researcher, I can create and edit groups of participants, so that I can view the aggregated data of many sessions.
- As a researcher, I can access files of raw data collected by the app, so that I can analyze the effects of the music on a large scale.

V. Design

A. Use Cases

Use Case: Initial Sign In to Application

Actors: Participant, Researcher

Flow of events:

- Participant recieves an invitation email from Researcher containing 4-digit PIN
- Participant downloads application to phone.
- Participant opens application and enters PIN
- Participant fills out form and enters spotify credentials

Entry Condition: Participant received an invitation and has a phone and an internet connection.

Exit Condition: Participant was able to download and open app, then provide their information and spotify credentials.

Quality requirements: No delays greater than 1 second when entering data into form.

Use Case: Complete A Session

Actors: Participant

Flow of events:

- Participant opens application
- Participant rates initial pain
- Participant presses the Start Session button
- Participant searches for and plays music while continuously rating pain by moving the slider

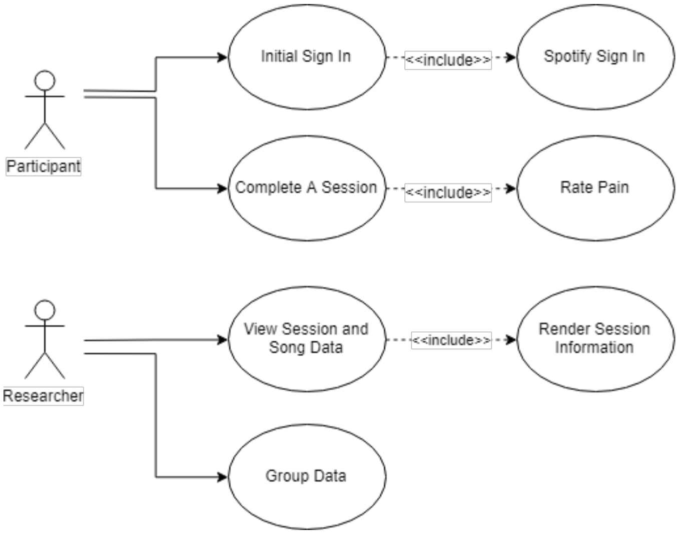


Fig. 4. Use Cases

- Participant presses ”End Session” button
- Participant reports whether or not they plan to use medication after the session

Entry Condition: Participant opens the application.

Exit Condition: Participant reports weather or not they plan on using medication at the end of the session.

Quality requirements: Participant can listen to all songs available through the Dpotify API.

Use Case: View Session and Song Data

Actors: Researcher.

Flow of events:

- Researcher navigates to IMAGS webpage
- Researcher selects a session
- Researcher views graph of session showing pain levels over the duration of the session and song information during the session

Entry Condition: Researcher can access the internet.

Exit Condition: Researcher viewed session data.

Quality requirements: No delays greater than 10 seconds when searching for or viewing session data.

B. Activity Diagrams

The team chose to use activity diagrams to model the flow of the system for both the patient's perspective and the researcher's perspective in order to gain a better understanding of the system's functionalities. From the patient's perspective, the first time user would go through a registration screen which when completed would request authorization from Spotify, send the user's personal information to Firebase, and generate a 4-digit ID number for the user. A user who has already registered would instead be directed to a login screen. Next, the user is directed to a screen with a button that starts a listening session. Once this button is pressed, the user then will be asked to rate their pain using a slider on a scale of 1 to 10. Once they have rated their pain, they then have the option to choose their song using the Spotify app, continuously change the pain rating as they listen, and end the current listening session. When the user selects a song, the app asks our backend what the currently playing song is. If the song being selected is a new song, we request the song characteristics from Spotify and keep track of the song on the backend. The same procedure occurs if the user records a change in their pain. The end session button tells the backend that the session has ended and to stop tracking songs played on Spotify.

From the researcher's perspective, the user will first be directed to a login screen. After logging in, they will have the option to either select a session, select a group, or manage a group. If the user selects a session, the app will pull session data from the backend and display it on a graph. The user then has the option to either select a song or select a particular time in a session, which will then provide a popup screen displaying the characteristics for the song or pain rating for the time selected. If the user selects a group upon logging in, the app will aggregate session data of the group selected from the backend and then go through the same display graph procedure. Finally, if the user decides to manage a group after logging in, the app will pull the users in the group from the backend. They will then be able to manage the users in a group, and then save the group and display its data.

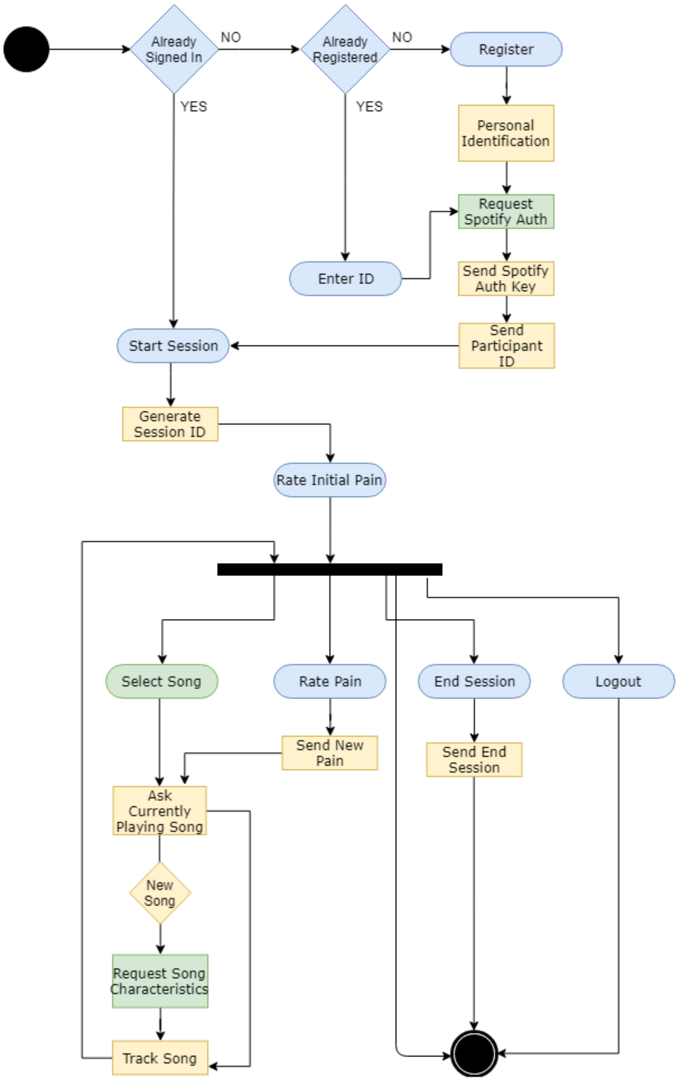


Fig. 5. Participant Activity Diagram

C. ERD

The team chose to use an Entity Relationship Diagram (ERD) to model and design the database for the app. The rectangular boxes represent entities that have data stored about them. The circles represent the attributes of each box, and the double circles represent multivalued characteristics.

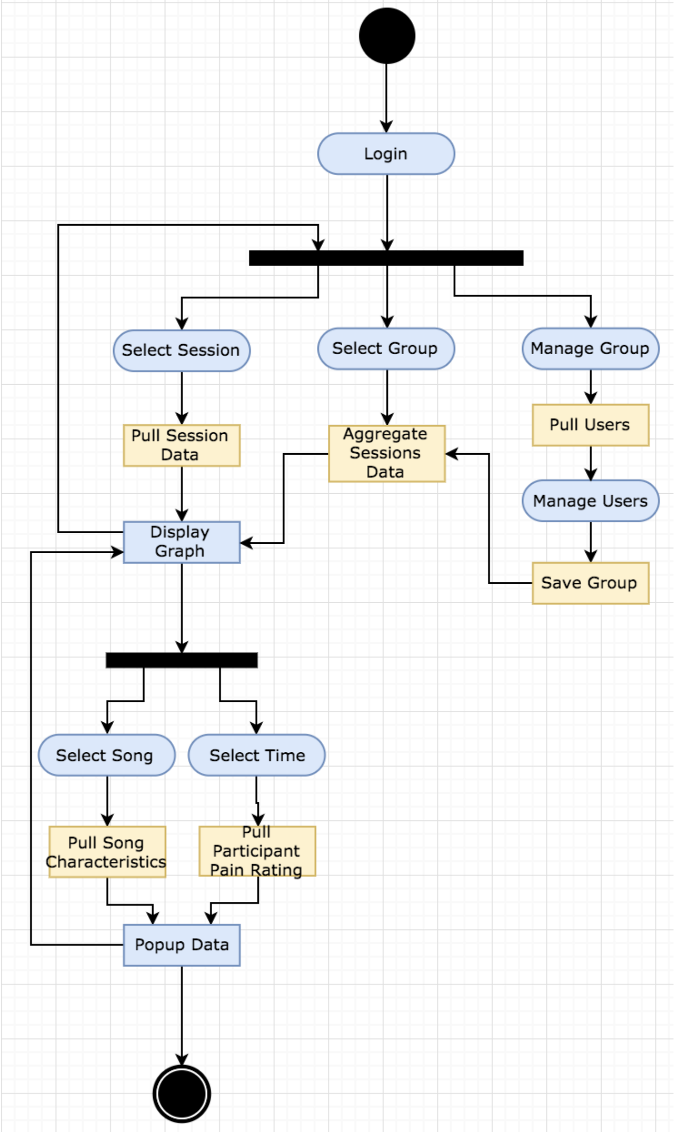


Fig. 6. Researcher Activity Diagram

The diamonds represent relationships between entities.

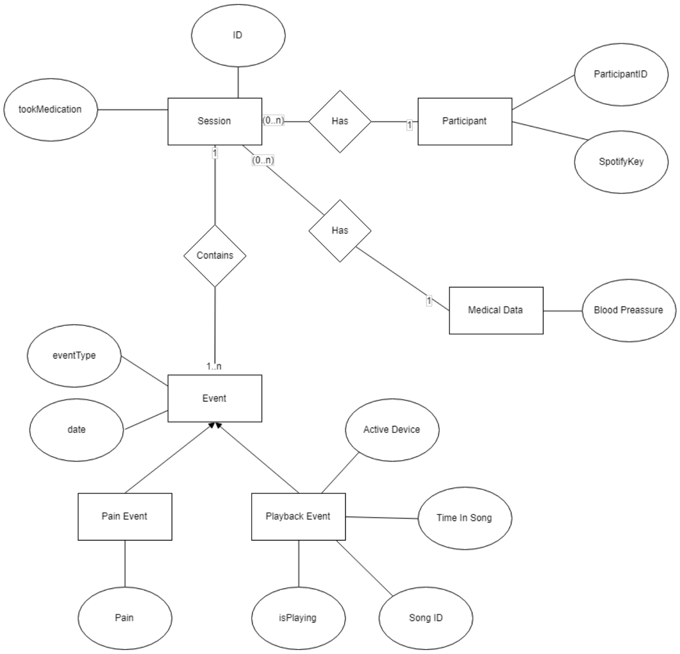


Fig. 7. ERD

D. *GUI Mock-Ups*

The team chose to use mock-up GUIs because they allow for fast prototyping of functionality for each screen in the app. They also allow for intuitive reading and easy revisions in the early stages of design. Though these mock-ups do not reflect some changes made to the first working prototype of the app, they are meant to reflect the core functionality required for each screen. The first screen is where the user would input their username and password. This would lead to the second screen, where the user starts their listening session. The third screen contains the pain rating slider and an end session button. The end session button brings the user to the survey asking if they still need to take their medication after the session.

The researcher's UI mock-ups consist of two screens, the select session screen and the data display screen. The select session screen allows the user to choose one of the sessions. Once a session is selected, the user is brought to the data display screen shown above. If the user wants to go back to the session select screen, they can press the IMAGS button on the top left of the data display screen.

VI. Implementation

A. *Iteration 0*

In this iteration, the team focused on creating an application prototype which created the interfaces for the participant user stories:

- As a participant, I can enter my spotify credentials and information such as name, so that researchers can record

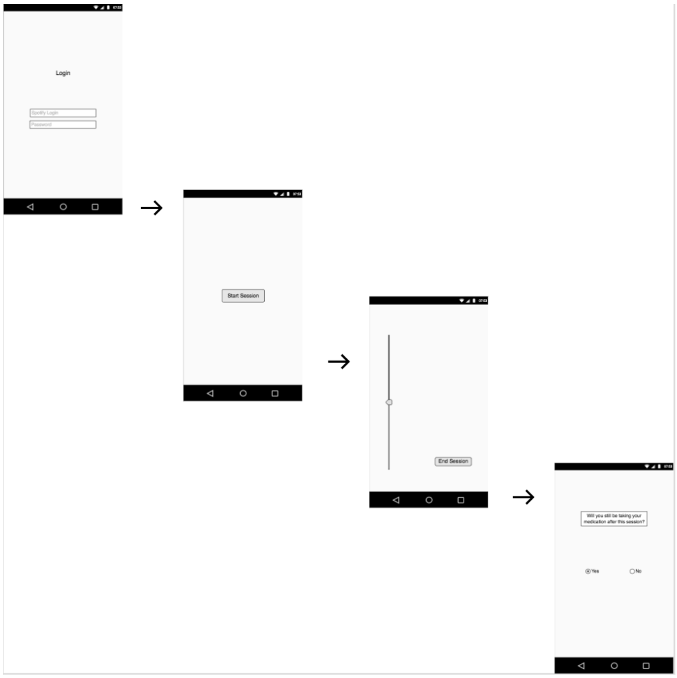

Login
Spotify Login
Password
Start Session
End Session
Will you still be taking your medication after this session?
● Yes
○ No

Fig. 8. Participant GUI Mockup

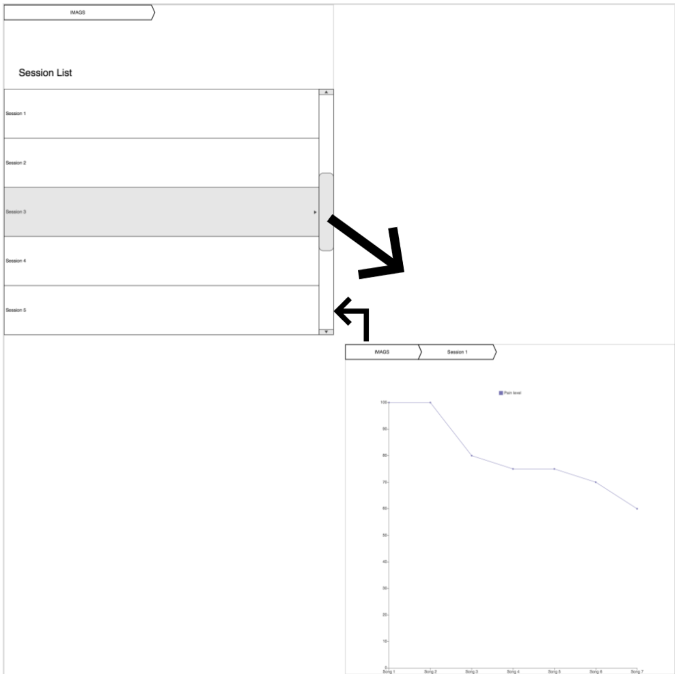


Fig. 9. Researcher GUI Mockup

the music I listen to.

- As a participant, I can start a music session, so that I can try to substitute my breakthrough medication.
- As a participant, I can continuously rate my pain, so that I can signify if the music helps.
- As a participant, I can stop a music session, so that I can signify if I took my breakthrough medication.

The prototype did not save or persist data entered into this interface. This iteration simply had the goal of creating a working interface that captured the participant's user stories.

In the first week of the sprint, the team successfully developed the initial prototype according to the design. A participant could use the prototype to enter information about themselves, provide their Spotify credentials, report pain through our application, and begin and end music sessions. To report pain during a session, the participant would use a pain slider on the main screen and switch to Spotify to play a new song, then return back to IMAGS to change the pain rating or end the session.

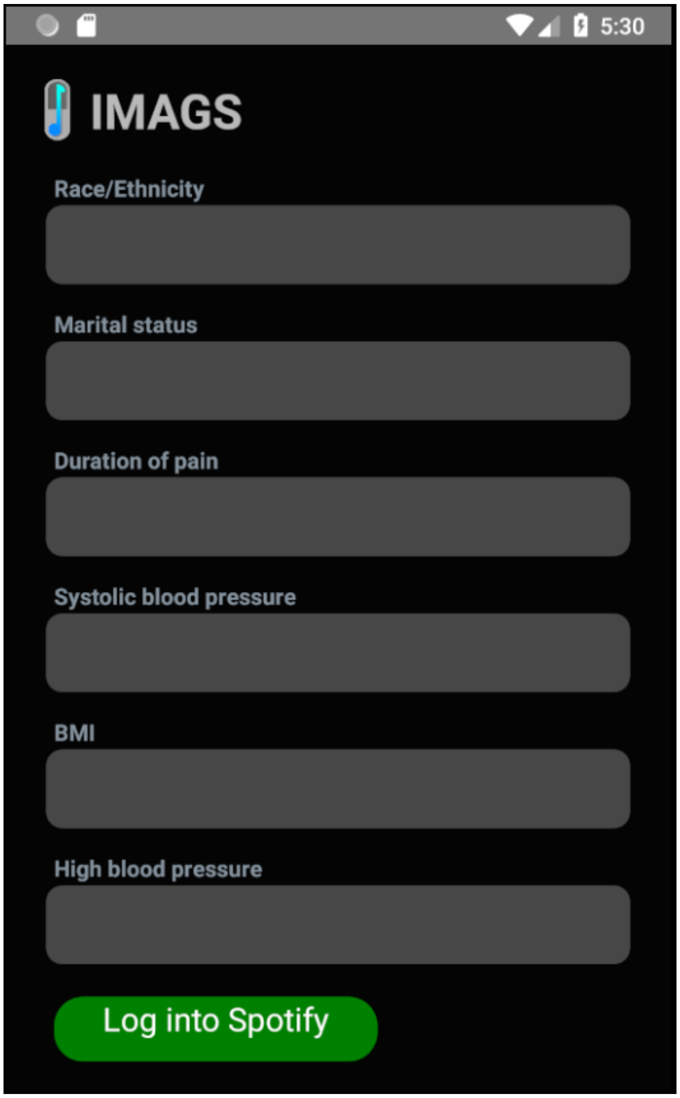
After demonstrating the initial prototype, the team received feedback on the design which created a new functional requirement. Requiring participants to switch between IMAGS and Spotify was found to be too complicated and could potentially make it difficult for participants to quickly change their pain rating after managing music in Spotify. To mitigate user confusion and increase ease of pain rating, a new functional requirement was created: the IMAGS application must allow users to select, play, and pause music from within the application. It is significant to note that if the team had used the waterfall approach, they would have finished the full stack application with the app-switching design only to find out that it would not be usable.

To accommodate these requirement changes, the team redesigned the app to be an all-in-one Spotify player application with a slider in the media controls to record pain. In the new design, the participant was able to enter personal information about themselves, provide their Spotify credentials, search Spotify for a song, play a song with full media controls, rate their pain, and begin and end listening sessions all within the same application. This new design has three main screens: initial patient data entry, Spotify login, and session management. Because the first two screens are only necessary the first time a participant uses IMAGS, they do not need to use those screen again after they login. Our prototype skips those screens and brings users directly to the session screen if they have already successfully provided their participant information and logged into Spotify.

The first screen queries the participant for their basic information required by the study. After filling in all of the fields and scrolling to the bottom, they may push a button to proceed to the Spotify login page.

The next screen requires participants to log in to Spotify in order to listen to Spotify songs. In order to create a simple and intuitive login experience the team used Spotify's API to make the standard Spotify login form appear, shown in the figure below. This form allows users to log in with their Spotify or Facebook account, or create a new account.

After providing Spotify credentials, the participant is brought to the main and final screen of the IMAGS application, where users may search for, play, and pause music while rating their pain on the slider near the bottom. The team decided to use simple icons representing happiness and sadness to indicate which end of the slider indicated low pain and which indicated high pain. Though the team only set out to create a interface prototype, this prototype successfully allows users to log in to their Spotify Premium account and play music from the IMAGS application.


5:30

IMAGS

Race/Ethnicity

Marital status

Duration of pain

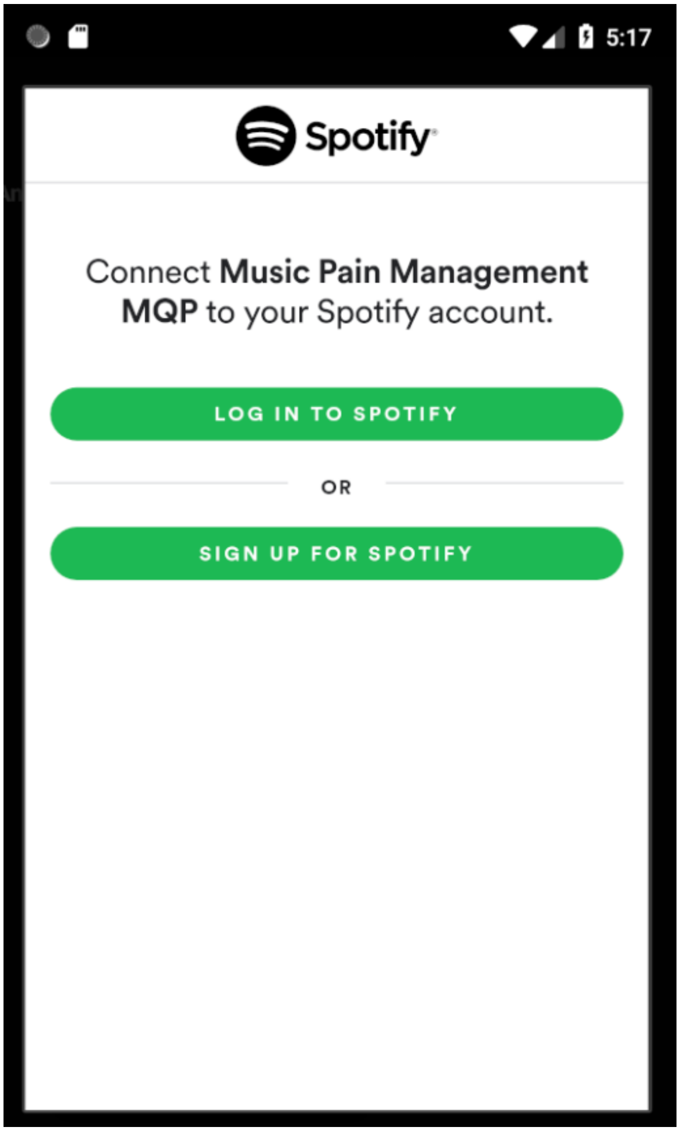
Systolic blood pressure

BMI

High blood pressure

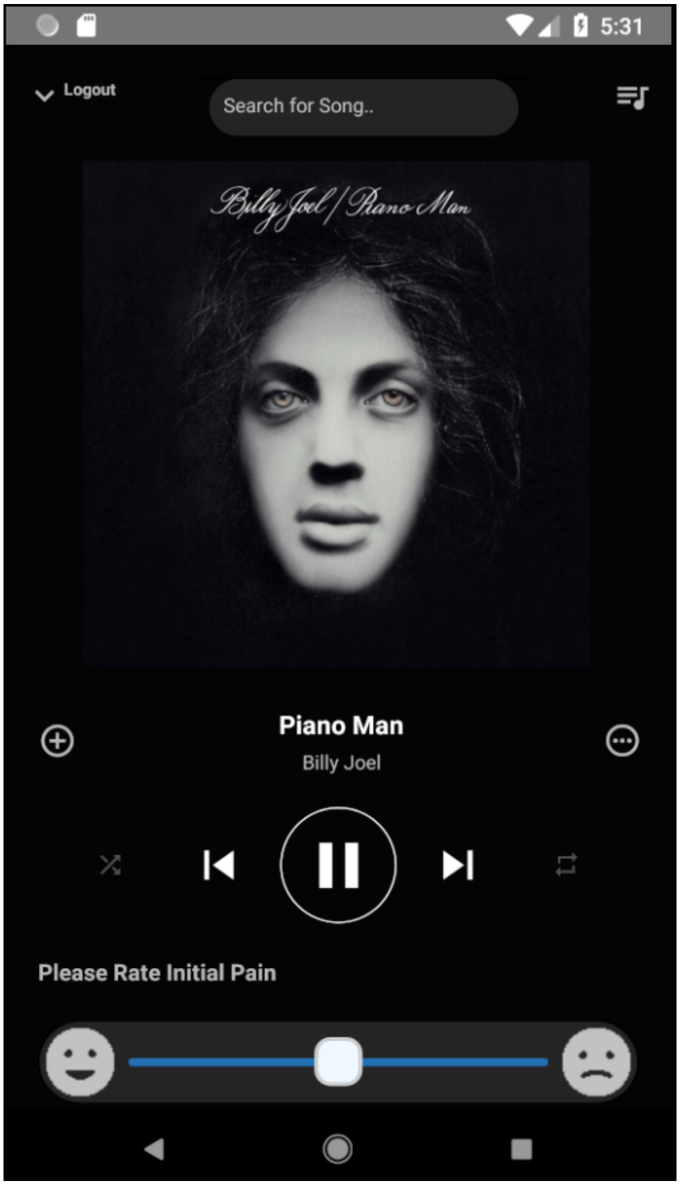
Fig. 10. Iteration 0 Setup Screen

Upon reflection of our first iteration the team found two big takeaways. The first is that our design for the first screen could be confusing as it is one large scrolling screen. The second is that while the first iteration keeps the user logged into Spotify when they leave the app, but it would log the user out after an hour and require them to re-enter their credentials. For next iteration, the team plans on breaking the initial screen up into multiple smaller ones to make it simpler, and automatically refresh the user's authentication tokens so that they do not have to re-login.



Connect **Music Pain Management MQP** to your Spotify account.

Fig. 11. Iteration 0 Spotify Login



Please Rate Initial Pain

Fig. 12. Iteration 0 Main Screen

B. *Iteration 1*

In this iteration, the team worked to create basic data capturing and visualization for researchers. The team developed the following user stories:

- As a participant, I can enter my Spotify credentials and my identification, so that researchers can record the music I listen to.
- As a participant, I have a profile with static information such as name, and dynamic information such as blood pressure, so that researchers know the information about me.
- As a researcher, I can search for and view a single session's reported data, so that I can analyze the effects of the music.

In the first week, the team worked on capturing participants actions, pain level, and song selection during a session as a series of events. When the session ends, those events are sent to the Firebase cloud database. Previously the IMAGS application required Spotify be uninstalled because IMAGS did not provide the correct developer key, which caused Spotify to block the API calls. In this iteration this blocker

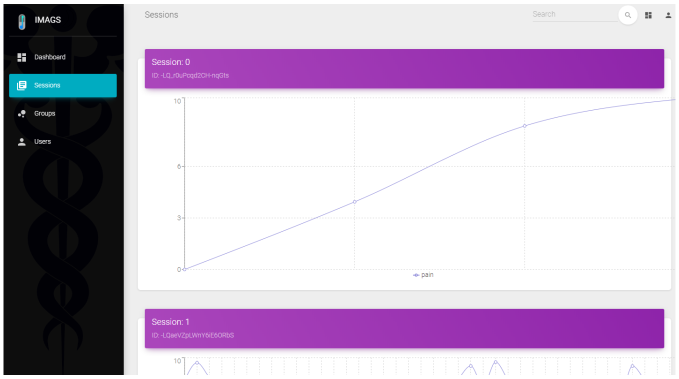


Fig. 13. Iteration 1 Researcher Data View

was fixed by providing a SHA-1 key to the Spotify developer console. This allows participants who are already logged into Spotify to begin using our application without logging in again.

In the next week, the team created a static website hosted by Firebase which allows researchers to view session data. In this iteration, session data is displayed as a simple chart with events on the x-axis and pain level on the y-axis. Though more data about the session is saved, this graph demonstrates the ability for the webpage to display data from Firebase. The team decided to use Google's material UI components for the website as it allowed for a responsive experience. The website was broken up into four different pages, a dashboard that displayed general information about the sessions reported, a session view that displayed the currently reported sessions, a group view, and a participant view.

In addition to creating the researcher's web page, the participant initial form was significantly improved. All of the data entered into the fields are sent to Firebase when the user logs in. Furthermore, the fields of this form were divided up into multiple pages to remove the need to scroll.

After reflecting on this iteration, the team identified clear goals for the next sprint. Though the current implementation of the pain rating chart was the first step to displaying data from Firebase, the chart must display significantly more information, such as which songs are playing at what time throughout the events. Upon further investigation of several graphing libraries available, the team found that the currently used library was capable of making simple graphs easily, but did not have the robustness to create complex graphs. The other option, D3.js, has the robustness necessary to create complex graphs, but is very low-level and would require significant time to implement. In addition to expanding on the graph functionality, the team plans to expand upon the participant information entry, editing, and display.

C. *Iteration 2*

In this iteration, the team developed the researcher's ability to view session information and create and edit participant data. The team worked on the following user stories:

- As a researcher, I can search for and view a group's aggregated session data, so that I can analyze the effects of the music.
- As a researcher, I can create participant profiles and enter their information, so that the participant does not have to.
- As a researcher, I can invite a participant to the application using the created profile, so that the participant's information is already entered when they log in.
- As a participant, I have a profile with static information such as name, and dynamic information such as blood pressure, so that researchers know the information about me.
- As a researcher, I can search for and view a single session's reported data, so that I can analyze the effects of the music.

In the first week of the iteration, the team focused on developing the Researcher's view of session data. After considering several graphing options, the team decided to

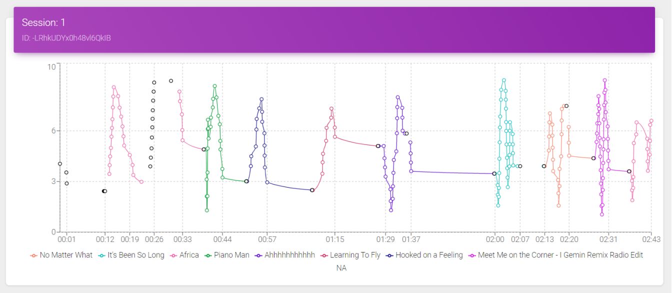


Fig. 14. Iteration 2 Aggregated Data View

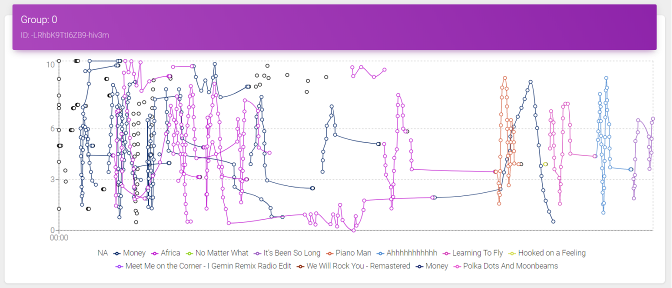
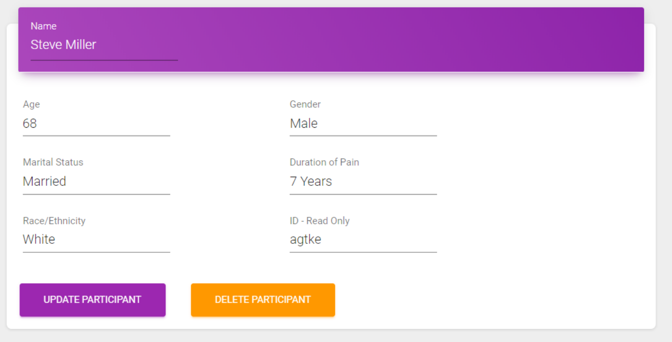


Fig. 15. Iteration 2 Aggregated Data View

continue using Chart.js as the graph rendering library. After significant refactoring, the session charts display which songs were played at what times. Each song played in a session has a distinct color, which is used on the lines of the graph to identify when that song is being played. This allows researchers to easily visualize which songs had effects on the participant's pain levels. The graph renders black dots with no lines when no music is being played (such as when the participant is rating their initial pain). This allows researchers to easily tell when music is being played.

In addition to labelling session graphs with songs, the ability for researchers to view aggregated sessions from a group was added. Aggregated group session graphs overlay data from every session in the group.


Name
Steve Miller

Age
68

Gender
Male

Marital Status
Married

Duration of Pain
7 Years

Race/Ethnicity
White

ID - Read Only
agtke

Fig. 16. Iteration 2 Researcher Participant Entry Form

In the second week the team focused on allowing researchers to create new participant profiles, enter information, and invite that participant to use the application. The participant profiles are accessed through the same website as the session data. Researchers can create new participants by pressing the ”New Participant” button. Participants have seven attributes:

- Name
- Age (Years)
- Gender
- Marital Status
- Duration of Pain (Months)
- Race\Ethnicity
- ID (Read Only)

These attributes were chosen based on the IMAGS research proposal. When a new participant is created, a new 5 digit alphanumerical ID is automatically generated. Researchers never need to edit this ID (and do not have the ability to). New participants have no values entered in any fields except the ID field. Researchers can populate any number of these these fields; if the researcher only has records for the participant's name and age, they can simply enter those fields and update the profile. Once a profile is created, the researcher can send the participant ID to the participant. When the participant downloads the application, they are prompted with the question ”Have you been provided an ID by a researcher?”. If they press ”Yes”, they can enter that ID and see the attribute values the researcher entered in the phone application form. They could edit those fields (in the case the researcher was incorrect), and enter attributes into fields the researcher did not populate.

Researchers may prefer participants create their own accounts from the phone application. If a participant downloads the application and is not provided an ID, they can press ”No” to the initial prompt. This creates a new participant profile with a new ID and empty fields. The participant is then directed to the form to enter their information.

After receiving feedback and reflecting on the iteration, the team identified several features to develop for the next iteration. Though the participant data entry fields store data correctly, they do not enforce types of values to enter. For example, the field ”Age” does not require a user to enter a number. The team plans to enforce data entry on these fields so that the data is in a consistent format for researchers. Furthermore, the session graph currently does not display the participant who created the session or attributes of songs. The team plans to expand the amount of data displayed in session graphs in the next iteration.

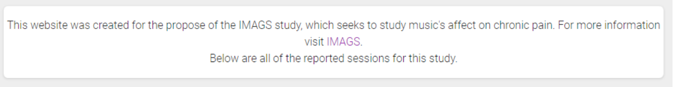
D. Iteration 3

In this iteration, the team developed the researcher’s ability to group participants and view a groups aggregated sessions from a single screen on the web application. The team worked on the following user stories:

- As a researcher, I can assign existing patients to groups and edit existing groups, so that I can aggregate data.
- As a researcher, I can search for and view a group’s aggregated session data, so that I can analyze the effects of the music.

In the first week of the iteration, the team worked to refactor the app to allow for groups to exist. Participants gained a group field, which could be edited by researchers in the web application, but not by participants on the phone application. To give researchers more flexibility, the group name can be any string, such as ”Group B” or ”New Jersey”.

After the first week, the team gathered more specific feedback on how exactly the aggregated group session view should look. Every group will have a single graph, similar to a session graph (with time on the x axis and pain on the y axis), which plots every session of every participant in that group.



This website was created for the propose of the IMAGS study, which seeks to study music's affect on chronic pain. For more information visit IMAGS.
Below are all of the reported sessions for this study.

Fig. 17. One of the several headers added in Iteration 4 to provide clarity on the purpose of the website and each page

In the second week, the team spent time modifying the existing Chart.js code to work with additional sessions. After successfully implementing the features for this iteration, the team got feedback that the group sessions screen should also show an ”All” group, which always shows every session, and a ”No Group” group, which shows all the sessions of every participant that does not have a group assigned.

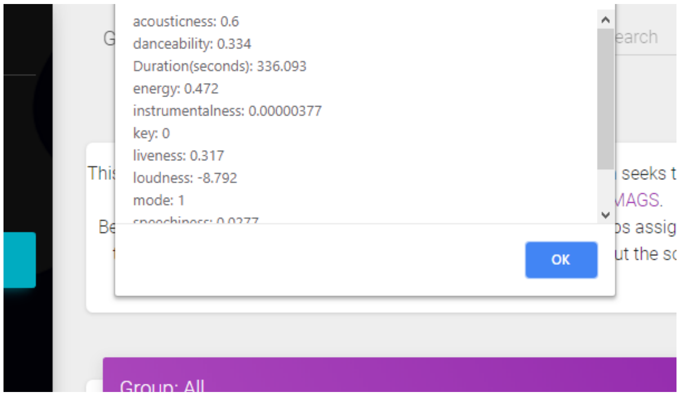
For next iteration the team planned to complete those 2 addition groups specified by the professors. After reviewing the features that have been created, the professors and team agreed it is time to focus on polishing the interfaces and text on the application, and prepare to try to create an iOS application.

E. Iteration 4

In this iteration, the team focused on polishing the interfaces of the web and Android application before attempting to port the application to iOS in the next iteration. While this iteration focused mostly on polish, it did address one user story:

- As a researcher, I can view the characteristics of a song, so that I can correlate the characteristics to the effects of the song.

In the first week of development the team worked to capture Spotify’s song attributes on the phone app and save them to Firebase. Then, the session charts in the IMAGS website were modified so that researchers could click on a



acousticness: 0.6
danceability: 0.334
Duration(seconds): 336.093
energy: 0.472
instrumentalness: 0.00000377
key: 0
liveness: 0.317
loudness: -8.792
mode: 1
speechiness: 0.0277

Fig. 18. Iteration 4: attributes of the song Africa by Toto being displayed on the IMAGS web app

song in the chart legend and get a pop-up listing the song's attributes, completing the user story of this iteration. In terms of polish, a header was added to each page of the IMAGS web app with a description of what the IMAGS study is and a link to the MQP paper. Further useful information about specific pages was added when necessary. The team also refactored the IMAGS website to be mobile friendly for ease of use. Before this iteration, the landing page of the IMAGS website had little useful information. Now it contains information on the number of sessions, groups and participants. As requested at the end of last iteration, 2 additional group session charts were added. The ”all” chart displays all the sessions across all of the participants, and the ”no group” chart displays all sessions from participants not in a group.

Looking back on this iteration the team believes that we have accomplished our goals for the phone and web application. The focus of the next iteration is on making the phone application iOS compatible. Several small problems with the presentation of the song attributes were identified in the most recent meeting and were fixed in the following and final iteration.

F. Iteration 5

With the web and Android application polished, the team spent this iteration using React-Native to create a iOS phone application from the same JavaScript code used for the Android application. Additionally, the team worked on allowing researchers to view Spotify attributes of songs. In order to port the application to an iOS device, the team needed to reconfigure each of the libraries used for the Android application inside of Xcode to compile the iOS version. After configuring all of the libraries the team had to remove conflicts between libraries. React-Native Elements and NativeBase libraries both had a package for a font that had to be resolved. Finally, the team was able to open the application from within an iOS simulator in Xcode. The team also continued to polish the app by reformatting the way the song attributes were displayed to make it more clear that duration was in seconds, and that tempo was in bpm. Additionally, the team worked on adding more text throughout the website to make it more clear what each page was showing. This was the teams last iteration and afterwards we had a reflection in which we assessed each critical user story and user story to ensure we met all of the requirements. With the fully functional website, Android application, and iOS application the team believes they accomplished the goals for the project and met all of the software requirements.

VII. Testing

A vital part of the Agile workflow is testing. This is because the application is built in several small iterations, and in order to construct the backlog for the next iteration, work from previous iterations must be tested. There are several different testing strategies, some of which are unit tests, integration testing, user-level testings, and regression testing. The primary testing strategy used by the team was smoke testing. Smoke testing is a type of build verification test which aims to ensure the most important functions of the application work by executing core use cases. Our smoke test was ran after nearly every commit, where one of us would ensure the basic critical users stories still worked, namely by creating a new user, listening to a song, changing the pain rating, and then verifying that it displayed properly on the researcher's application. By the principles of smoke testing, if these basic use cases still worked, then our build was verified and could be further tested. Alongside of this smoke testing strategy was our regression testing strategy, in which every few commits we went back and tested the functional and non-functional requirements of previously developed code. Next, we performed user-level testing every iteration. Our user level testing strategy was to install our application on a physical phone and have a user perform all of the use cases. In addition to confirming that our use cases worked, we asked the users what features they thought could be changed or added, and what parts of the application they thought were not clear. The team then used this information to construct the next backlog, which included the improved features.

VIII. Assessment

The team's goal for this project was to create an application that allowed both participants to rate their pain while listening to music and researchers to view that session data. After five two-week iterations of development, the team believes that it has succeeded in that goal. All of the stated functional requirements, non functional requirements, and user stories were satisfied. This project helped the team refine and develop various software engineering skills. The extensive two-month planning phase allowed the team to thoroughly study a variety of software stacks to identify the most effective one for the project. The software stack selected significantly accelerated the software development process, reinforcing the belief that time spent planning is highly valuable when approaching a large software project. The team also gained experience with cross-platform development. IMAGS uses applications across iOS, Android, and web platforms. Identifying and implementing a software stack that was designed for cross platform deployment allowed the team to familiarize themselves with powerful, modern, software tools.

IX. Future Work

The team considered several features that would be valuable additions to IMAGS in future iterations. Participants searching for songs in the phone application should be prompted with a pop-up to select between the songs or albums that are related to the search. Though the group session screen is fully functional based on requirements gathering, the team recommends further investigation into what new forms of data visualization researchers may find valuable.

X. Conclusion

IMAGS successfully uses applications across the iOS, Android and Web platforms to make conducting self-selected music analgesia studies significantly easier by automatically collecting, managing, and displaying study information. This project was a success because of the team's effective usage of time allotted to planning, researching, and prototyping in the early months of work. Spending several weeks researching applicable software tools (such as languages, technology stacks and IDEs) allowed the team to find and learn tools which saved them more time later in the project. The use of Firebase to store IMAGS data saved significant development time because the team did not need to manage synchronizing data across devices. The use of React-Native allowed the team to rapidly deploy the phone application developed on

Android to iOS. Taking time to gather requirements from mentors and creating a prototype early in the project allowed the team to easily overhaul the design of their application before development began. When developing, the Scrum Agile methodology enabled the team to continuously add features and receive feedback from mentors. Small modifications to previous work were made every iteration, which improved the overall polish of the application. IMAGS, when utilized by researchers, will aid in the identification of pain-reducing attributes in music as an alternative to opioids.

XI. References

Cepeda, M. S., Carr, D. B., Lau, J., Alvarez, H. (2006). Music for pain relief. Cochrane Database of Systematic Reviews, (2) doi:10.1002/14651858.CD004843.pub2

Cloud functions for firebase. Retrieved from https://firebase.google.com/products/functions/

Garcia, R. L., Hand, C. J. (2016). Analgesic effects of self-chosen music type on cold pressor-induced pain: Motivating vs. relaxing music. Psychology of Music, 44(5), 967-983. doi:10.1177/0305735615602144

Jibb, L. A., Stevens, B. J., Nathan, P. C., Seto, E., Cafazzo, J. A., Stinson, J. N. (2014). A smartphone-based pain management app for adolescents with cancer: Establishing system requirements and a pain care algorithm based on literature review, interviews, and consensus. JMIR Research Protocols, 3(1), e15. doi:10.2196/resprot.3041

Mahalakshmi, M., Sundararajan, M. (2013). Traditional SDLC vs scrum MethodologyA comparative study. International Journal of Emerging Technology and Advanced Engineering, 3(6), 192-196.

Manzo, V.An exploration of utilizing listener-selected music in chronic pain management research plan. ().

Most popular music streaming services in the united states as of march 2018, by monthly users (in millions). (2018). Retrieved from https://www.statista.com/statistics/798125/most-popular-us-music-streaming-services-ranked-by-audience/

Overdose death rates. Retrieved from https://www.drugabuse.gov/related-topics/trends-statistics/overdose-death-rates

Shah, N., Jonassaint, J., De Castro, L. (2014). Patients welcome the sickle cell disease mobile application to record symptoms via technology (SMART). Hemoglobin, 38(2), 99-103. doi:10.3109/03630269.2014.880716

Spotify for developers. Retrieved from https://developer.spotify.com/documentation/web-api/

Volkow, N. D., McLellan, A. T. (2016). Opioid abuse in chronic pain misconceptions and mitigation strategies. N Engl J Med, 374(13), 1253-1263. doi:10.1056/NEJMra1507771

XII. Appendix A: Manual

This section outlines the steps participants, researchers, and developers need to take to utilize IMAGS.

A. Participants

Participants will be contacted by researchers with instructions on where to go to download the IMAGS application. In that message, the researcher may also specify a 5-digit code. If so, press ”yes” when asked if a researcher provided a code, and enter this code to connect to your account. If no code was provided, press ”no” when asked, and continue. Enter your medical information into the fields, and press next when done. After entering your medical information, you must log in to Spotify. Either use an account provided by the researcher or your own account. On the music player screen of the application, rate your initial pain by moving the dot along the slider. Moving the dot all the way left represents no (0/10) pain. Moving the slider all the way right represents extreme (10/10) pain. Type in the full name of a song in the top search bar and press search to play the song. Continuously move the slider as pain changes. Press ”end session” when you are done listening to music, and report weather or not you plan to take medication after the session.

B. Researchers

To conduct a study using IMAGS, researchers will need to ensure every participant has access to a Spotify premium account and an Android or iOS phone. If researchers have medical information about the participants they wish to enter themselves, they should visit https://imags-web.firebaseapp.com/users. For each participant, press ”new participant” and modify their data accordingly. researchers do not need to create profiles for their participants, instead, invited participants can create profiles on their own and enter their information themselves. Whether or not the researcher created profiles, they must invite the participants by emailing them a link to the application on the Android or iOS app store. If the researcher created a participants information beforehand, they must also add that participants 5 digit ID to the email so they can access their account. Once participants are invited, researchers can view all sessions created by those participants from the sessions tab. If researchers wish to view all sessions from a group of participants, they should edit those participants ”group” field to be the same name (such as ”Group B” or ”New Jersey Group”). To view group data, go to the group tab.

C. Developers

This application uses the React Native technology stack. This allows developers to write one set of JavaScript code that can be deployed to many platforms. Learn more about the basics of React Native at https://facebook.github.io/React-Native/ This application stores its data on firebase at https://console.firebase.google.com/project/imags-web/database/imags-web/data/

Firebase is a Google service which allows for fast and easy data upload and real-time synchronization across devices.

To access the IMAGS website repo and see instructions for building and deployment, visit https://github.com/linuxboss182/imags_web

To access the IMAGS phone app repo and see instructions for building and deployment, visit https://github.com/linuxboss182/IMAGS