ARC (Audio Resource Companion)

Ryan Kildea

University of Missouri – Kansas City

Table of Contents

[Vision Statement 3](#_Toc522714937)

[Requirements 4](#_Toc522714938)

[Product Backlog 5](#_Toc522714939)

[Sprint #1 6](#_Toc522714940)

[Review 6](#_Toc522714941)

[Retrospective 6](#_Toc522714942)

[Sprint #2 7](#_Toc522714943)

[Review 7](#_Toc522714944)

[Retrospective 7](#_Toc522714945)

[Sprint #3 8](#_Toc522714946)

[Review 8](#_Toc522714947)

[Retrospective 8](#_Toc522714948)

[Sprint #4 9](#_Toc522714949)

[Review 9](#_Toc522714950)

[Retrospective 9](#_Toc522714951)

[Sprint #5 10](#_Toc522714952)

[Review 10](#_Toc522714953)

[Retrospective 10](#_Toc522714954)

[Sprint #6 11](#_Toc522714955)

[Review 11](#_Toc522714956)

[Retrospective 11](#_Toc522714957)

# **Vision Statement**

Audio Resource Companion (ARC) is a music application for Android operating systems. The program will allow users to import their collection of music files for playback. ARC will pull artist information from these files and, using an undetermined API, provide the users with a backlog of the most recent albums released by all artists found in their library. Stretch goals for ARC involve features that are often not highlighted in traditional media players. In time, ARC will identify songs from the user’s collection and use APIs in order to retrieve lyrics for the current song, album artwork, and links to music videos.

# **Requirements**

A discussion of what your application is required to have in functionality. It should identify user roles and goals/actions, and what the key features of the app should be.

|  |  |
| --- | --- |
| **Actor** | **Goal** |
| User | Provide an interface for easy navigation between the library, artist lists, album lists, and song lists. |
|  | Feature for importing music from the user device into the library. |
|  | Audio playback functionality; play, pause, rewind, fast forward, skip, repeat, randomize. |
|  | Presentation and, if applicable, editing of audio file tags, including artist, album, song title, running length, release date, lyrics, etc. |
|  | (For recently released albums feature) An API is needed that requests access to the backlog of all albums released by a given artist for all artists in the user library. |
|  | (For lyric look-up functionality) An API is needed that requests lyrics for a given song |
|  | (For music video functionality) An API that receives information on if a music video is available for a given song |

### Product Backlog

This will be updated throughout the semester as new PBIs are added, larger items are broken into smaller ones, and completed items removed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Story ID** | **Story** | **Story Points**  **(in est. hours)** | **Priority** | **Status** |
| 12 | The user should see a splash screen upon starting the application | 2 | Low | Complete |
| 1 | The user should be greeted by a home screen | 0.5 | Very high | Complete |
| 2 | The user should have access to a toolbar that allows access to their library | 0.25 | Very high | Complete |
| 3 | The user should be able to import music files into their library; music files should be listed with Artist, Album, Song Title, and Track Length | 6 | Very high | Unfinished |
| 4 | The user should be able to click a song and play that song | 4 | High | Complete |
| 5 | The user should have access to audio controls; player volume, play, pause, skip to previous/next track, skip X seconds back/forward, randomize, repeat, view lyrics | 35 | High | Unfinished |
| 6 | The user should be able to swap between different library view modes; “Artists” view, “Albums” view, “Songs” view | 2 | Medium | Unfinished |
| 7 | The user should be able to view the properties of a file in their library | 2 | Low | Unfinished |
| 8 | The user should be able to edit song properties such as artist, album, song title, lyrics, and album artwork | 4 | Low | Unfinished |
| 9 | An API should be implemented to query all of the artists in the users’ library and return a chronological list of latest releases of all their artists | 6 | Very low | Unfinished |
| 10 | An API should be implemented to pull in album artwork for all albums in the users’ library, should the user request it | 3 | Very low | Unfinished |
| 11 | An API should be implemented to pull in lyrics for a user’s song, should the user request it | 3 | Very low | Unfinished |
| 12 | The user should see a splash screen upon starting the application | 2 | Low | Complete |
| 13 | The artist page should display clickable buttons with an artist picture on them | 4 | Low | Complete |

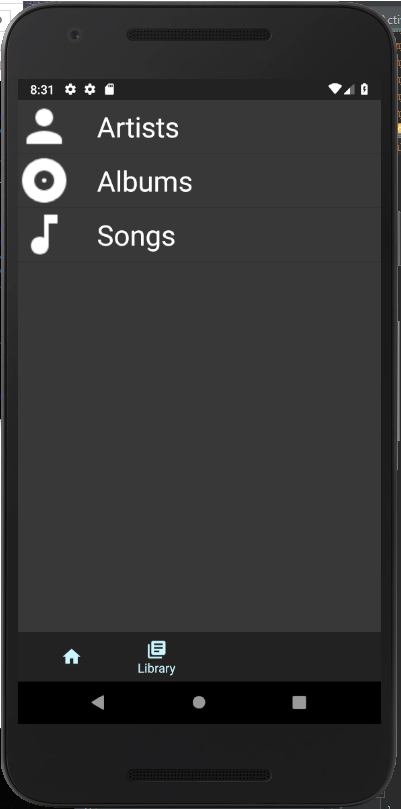
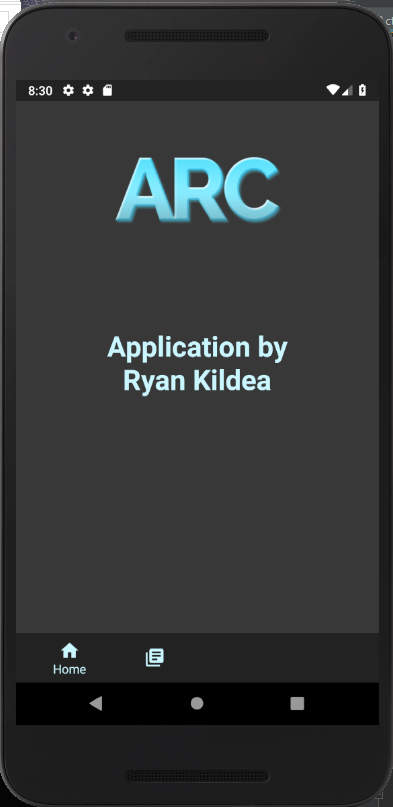
# **Sprint #1**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 1 | The user should be greeted by a home screen | 0.5 | 1 |
| 2 | The user should have access to a toolbar that allows access to their library | 0.25 | 4 |
| 3 | The user should be able to import music files into their library; music files should be listed with Artist, Album, Song Title, and Track Length | 6 | N/A |
| 4 | The user should be able to click a song and play that song | 3 | N/A |
| 12 | The user should see a splash screen upon starting the application | 2 | 3 |
|  |  |  |  |
|  |  |  |  |

## 

## Review



Added during this sprint was the splash screen for the app, a dynamic bottom action bar, the home screen, and the interface for the library. The ability to import files and play songs has not yet been finished.

## Retrospective

I think the process went well in general. The biggest things were first that I overestimated how much could be accomplished while being semi-new to Java and totally new to Android Studio. In the future, I will try to include this into my factoring of time. Also, since this app is not for an organization, I went ahead and completed a low-priority item (the splash screen) because it seemed to help polish the application a bit and will give me consistent results moving forward. I probably should have stuck to my priority plan, but regardless, the feature was completed and I am happy with my application so far.

# **Sprint #2**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 3 | The user should be able to import music files into their library; music files should be listed with Artist, Album, Song Title, and Track Length | 6 | Not finished |
| 4 | The user should be able to click a song and play that song | 4 | 4 |
| 13 | The artist page should display clickable buttons with an artist picture on them | 4 | 5 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

## 

## Retrospective

[This is where you discuss the process. What went well (and are you planning to do more of that?) What didn’t go so well (and do you have a way to do less of that)? What changes are you planning to make in how you plan & carry out the next sprint?]

The things that went really well are that I learned a lot of new features for both Android and Java this sprint and I am starting to feel more comfortable with both. I have constantly been venturing out of my comfort zone while making this app and it is working out and very cool. The things that didn’t go well is that I seem to spend an inordinate amount of time on more trivial details. For example, I did not want the artist icons to show up as solid squares – I wanted rounded ones instead, and figuring out how to do this added many hours to my project. I think the result in the end is worth it, I just don’t think it would be a good idea to do so in a corporate setting. For the next sprint, I hope to stay focused and get done with the key functionalities, instead of focusing on more trivial details.

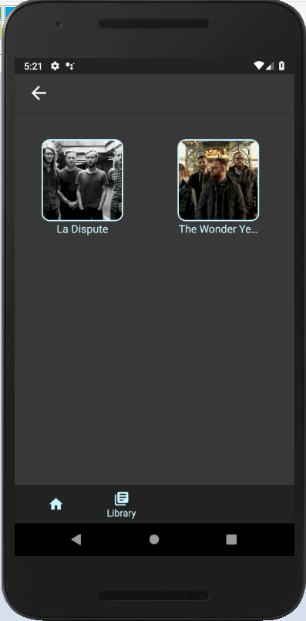
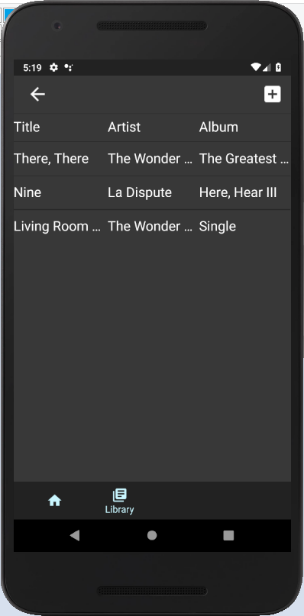
# **Sprint #3**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 3 | The user should be able to import music files into their library; music files should be listed with Artist, Album, Song Title, and Track Length | 10 | Not finished |
| 5 | The user should have access to audio controls; player volume, play, pause, skip to previous/next track, skip X seconds back/forward, randomize, repeat, view lyrics | 6 | Not finished |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

[Screenshots, etc go here. This is where you discuss the product, describing what was done this sprint (potentially shippable product increment) and what was planned for the sprint but was not done. ]



None of my stories were fully completed during this Sprint. That said, I did a major overhaul of the code that I had and did a ton of refactoring to make what I hade easier to build upon. This included redoing how I was creating my fragments; now, they are added to a stack, so that when I am in a subfragment of a fragment I can go back to the previous fragment without exiting all the way back to the activity. A toolbar now exists for all of the subfragments of the library class. MP3 attribute creation was not working properly, so I heavily modified the code there. Also, library now keeps track of unique artists, their unique albums, and the unique songs on each album via the implementation of sets.

## Retrospective

[This is where you discuss the process. What went well (and are you planning to do more of that?) What didn’t go so well (and do you have a way to do less of that)? What changes are you planning to make in how you plan & carry out the next sprint?]

Though no story was completed, progress was made and an immense amount of code was restructured to make moving forward easier. A lot of “technical debt” was paid off this sprint. My plan to make next sprint more successful is to, now that so much refactoring has been done, try to tackle the major tasks that have been holding me up the past two sprints.

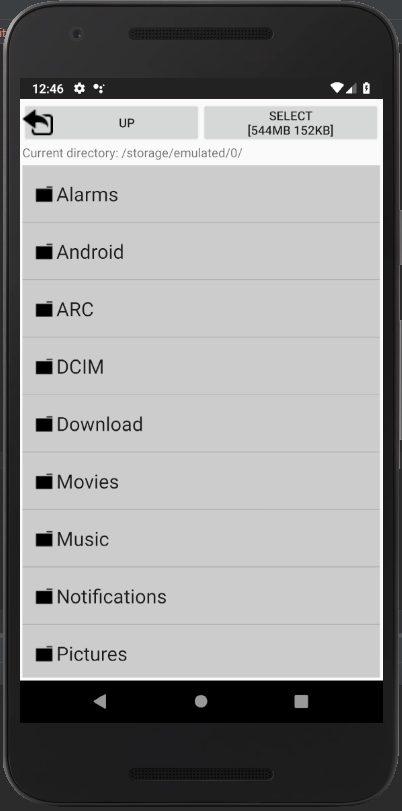
# **Sprint #4**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 3 | The user should be able to import music files into their library; music files should be listed with Artist, Album, Song Title, and Track Length | 16 | Not Completed |
| 5 | The user should have access to audio controls; player volume, play, pause, skip to previous/next track, skip X seconds back/forward, randomize, repeat, view lyrics | 6 | Not Completed |

## Review

[Screenshots, etc go here. This is where you discuss the product, describing what was done this sprint (potentially shippable product increment) and what was planned for the sprint but was not done. ]



## This sprint still did not see the completion of my two stories, but a ton of progress was made. The two most notable parts of this sprint was the implementation of a library for file importing as well as an even more in-depth overhaul of my fragment/activity system. For the file importing, you are now able to add MP3 files to ARC from outside of the SD Card “Music” folder. However, I have not marked it as “complete” because I still have not found out how to get access to the complete device, and so right now only the “Emulated Storage” folders are accessible. The responsibility of exchanging fragments was removed from each fragment and now occurs solely through the main activity. This has improved consistency, shortened the code tremendously, and made it possible for me to implement (which I did) the ability to save fragments instead of destroying them and recreating them each time a user tries to navigate through the menus. In doing so, the app now runs incredibly smoothly and is more responsive. Additional updates increasing functionality were added to the artist, album, and song classes, and a prototype of the Artist-Album-Song view was implemented.

## Retrospective

[This is where you discuss the process. What went well (and are you planning to do more of that?) What didn’t go so well (and do you have a way to do less of that)? What changes are you planning to make in how you plan & carry out the next sprint?]

I continue to make a ton of great and fun changes to the app while still failing to accomplish the goals laid out in my sprint. Hopefully soon there will be no refactoring or abstracting to do and I can focus solely on completing my stories.

# **Sprint #5**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 3 | The user should be able to import music files into their library; music files should be listed with Artist, Album, Song Title, and Track Length | 25 |  |
| 5 | The user should have access to audio controls; player volume, play, pause, skip to previous/next track, skip X seconds back/forward, randomize, repeat, view lyrics | 6 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

[Screenshots, etc go here. This is where you discuss the product, describing what was done this sprint (potentially shippable product increment) and what was planned for the sprint but was not done. ]

In this sprint, I worked on integrating the Spotify API into my project to handle the import and music playing process. I was able to authorize and authenticate my app with Spotify’s server, but got stuck when trying to access data using a curl command. I spent a lot of time working on this but was unable to figure it out. To make matters worse, I attempted to push these changes to a development branch using git, but made a huge mistake and overwrote all my progress I made this sprint. I should be able to relatively quickly return to where I was, but this is frustrating nonetheless.

## Retrospective

[This is where you discuss the process. What went well (and are you planning to do more of that?) What didn’t go so well (and do you have a way to do less of that)? What changes are you planning to make in how you plan & carry out the next sprint?]

I need to make sure next sprint that I do not overwrite all of my changes, and try to continue working on how to perform curl commands within Java.

# **Sprint #6**

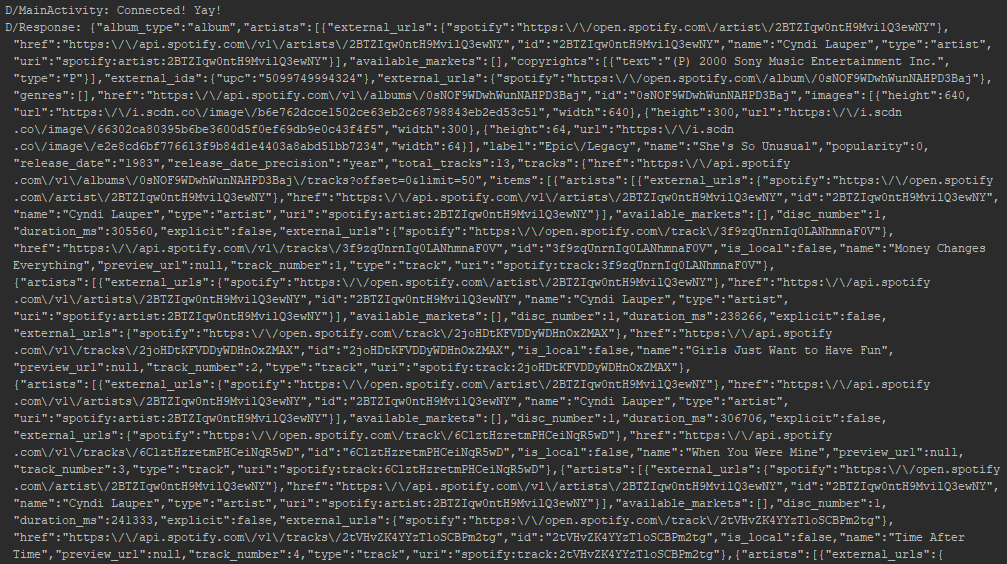
Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| 3 | The user should be able to import music files into their library; music files should be listed with Artist, Album, Song Title, and Track Length | 35 |  |
| 5 | The user should have access to audio controls; player volume, play, pause, skip to previous/next track, skip X seconds back/forward, randomize, repeat, view lyrics | 6 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

[Screenshots, etc go here. This is where you discuss the product, describing what was done this sprint (potentially shippable product increment) and what was planned for the sprint but was not done. ]

The biggest thing finished this sprint was finally getting connected and running to the Spotify API. With this, I now have access to music controls as well as access to all of Spotify’s artist catalog, allowing me to import artwork, names, find missing albums, etc. This was extremely exciting to get functioning. Also, I started working on Play/Pause functionality, and restructured my AudioHandler to be easier to use and more modular.



## 

## Retrospective

[This is where you discuss the process. What went well (and are you planning to do more of that?) What didn’t go so well (and do you have a way to do less of that)? What changes are you planning to make in how you plan & carry out the next sprint?]

This final sprint was exciting and it is cool to look back on what I created this semester. I definitely learned a lot about SCRUM principles and how working in an Agile frame of mind can make development easier and more cohesive. This was a fun project and I am looking forward to continuing it in the future.