Asymmetrical Android App

Christopher Swayne

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**

Asymmetrical is an application that keeps a directory of your favorites. Most people have an assorted mental list of your favorite bands, songs, books, movies, video games, etc., or they have their library in individual applications that contains a portion of their favorite media.

What Asymmetrical is aimed to achieve is a storage system that helps organize all your favorite media, and not only your favorites but all the media you have seen, heard, played, or read. The immediate release requirements should allow:

* Nested data management
* Thumbnail upload/storage for each title
* User reviews of media
* Local device storage

Nested data management would allow for simple and intuitive organization by categorization of media.

Thumbnail support would give a visual queue for the users to be able to immediately recognize the media by not only its’ name but by the associated picture.

User reviews can help with allowing for media descriptions, as well as the user able to ascertain their opinion/review of the entry in the directory.

Local device storage is important as the data must be kept persistent.

Future development of the application would have:

* Online storage, potentially on the Users’ Google Drive, iCloud Drive, or OneDrive cloud storage.
* Shareable User Reviews

Most people who have their library of media divided between many sources, whether that be physical or digital libraries would have an easy appreciation to have a single place to document what they own, stream, or license.

# **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** |
| Super User | Add tags to items to identify/group them |
| Novice User | Easily add entries |
| Novice User | Dark Mode for ease of use at night |
| Super User | Rate/Review media |
| Parental User | Parental controls allow parents to see how their kids are using the app. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### 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** |
| A | Entry Database | 4 | 1 |  |
| B | User Interface | 6 | 2 |  |
| C | Parental View | 10 | ~ |  |
| D | Entry Editor | 10 | ~ |  |
| E | Online storage | 8 | ~ |  |
|  |  |  |  |  |

# **Sprint #1**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| A1 | Entry Data storage | 2 | 1 |
| A2 | Reference Storage (Thumbnails,Ratings,Reviews) | 2 | ~ |
| B1 | Main User Interface | 4 | 1 |
| B2 | Entry Screen | 2 | 3 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

This sprint, the UI for the New Entry screen was created, as well as the framework for the home screen was created. The properties for entries were created, and the planning for the center UI design was finished.

However, the Data storage system was not completed, and is still in the planning phase as to the methodology of storage.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

## Retrospective

The process of this sprint was not adequate enough. Next sprint I need to plan effectively throughout the period, rather than upfront for each sprint. In figuring out the sprint goals that were too broad, I managed to complete two of the sprint back-log items.

I also had a lack of time management, whereas I only worked on the project in spare time I had, rather than dedicating a full period of time to the project.

Next sprint, I will be fully designing the data storage mechanism, and thoroughly planning out the backend structures, getting tests done on those and moving forward with only working code before the 3rd sprint.

The biggest change will be to plan the time for the project to be every other day exactly, for at least 3 hours, as to keep progress being made on the project, and the sprint backlog. To help keep myself following through with this planning procedure, I will be using a series of calendar events and alarms to help myself focus time to what needs to be done, and when.

# **Sprint #2**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| A1 | Directory/EventData Storage | 6 | 12 |
| A2 | Home Screen UI | 2 | ~ |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

Work on the Home Screen UI hasn’t progressed much, as the UI wasn’t completed, in large part due to the database storage taking up most of the time this sprint. The Database has entries, and compiles properly. However, because the UI hasn’t been completed, we cannot screenshot the finalized product working, though the test cases showed the demonstration of database working, with a basic View that tests the database as working. This view is not the final product, nor should be used as an example of the final product.

A screenshot of a cell phone

Description automatically generated

## Retrospective

This sprint worked out will in terms of proceeding further than the previous sprint. However, the drawbacks to the effectiveness of scheduling were not significant in terms of shippable product. For the next sprint, I will proceed to have a better use of time in ensuring all the work gets done and addressed, as while I gave myself enough time to complete a task, I didn’t adequately plan the time out to ensure that I was efficiently ensuring that the work got done in the most effective manner.

# **Sprint #3**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| A1 | Folder/Note Storage | 6 | 8 |
| B1 | Folder View | 2 | 3 |
| B2 | Home Screen/Main View | 2 | ~ |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

The progress that has been made on the product is quite substantial for this sprint. However, there was a step back after the previous sprint, as a change from using the Room Interface for a SQL lite database made it challenging to integrate within the hierarchical structure that is necessary for a “File Structure” like design. This sprint focused on implementing the designs of the views, as completed above, and changing the design from SQL to using JSON as persistent information storage locally to the device running the application.

A majority of the time taken in the last sprint was used learning how to integrate a Room Database to work with the Design Requirements of a File System structure for the application. This sprint, while doubling back on work that was done, has improved upon the design of the previous sprint, using a different method, as well as continuing to add the required backend of features for user needed functions, like deletion, creation, and editing notes and folders.

## Retrospective

This sprint went well in coming to reality with the troubles of design. The changes that were made had been needed, but difficult to swallow cutting out hours and hours of work that had previously been done. The UI had not changed too significantly, other than refactoring and changing the names of items to adjust the app to be more than simply a “directory” of shows, books, and the like to be stored within a file system. This will allow for Notes, and additional data to be stored, like other notetaking apps, you have the ability to type and store masses of text. However, the Main UI, Main View had not been completed due to requiring that the folder view be completed beforehand, as the Main UI will base itself more so off the *in folder view* than being a unique view in itself.

Due to the software changes, more time next sprint will be needed to continue to push forward the design and construction of the application, but we are coming closer than before, as one of the bigger hurdles was this persistent storage system that has been completed. Next sprint, I plan to continue to work every other day on the software to improve it, dedicating a slightly more significant amount of time to the project to ensure that everything in whole is completed.

# **Sprint #4**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| A1 | Card/Main View | 2 | ~ |
| A2 | Card View Handler | 3 | ~ |
| B1 | Main Controller | 6 | 6 |
| B2 | GSON framework | 3 | 4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

This sprint, I was able to achieve a strong amount of progress on the backend of the app, where I simplified and got more essential functions of the product working, though they are not yet a part of the view to be interacting with the user.

Getting the main view and the Cards inside the recycler view did not follow through. The development of use of the Google Gson library continued, as the refinements needed to get more of the app useable were performed, such as simplifying methods and classes to reduce coupling with the Folder/Note Data Controllers (with plans to combine them into a single simplified object within the Main Controller) The use of two separate controllers made for an increase in coupling, and unnecessary dependencies upon each other to creating additional JSON files.

## Retrospective

Unfortunately, this sprint, I had an unplanned injury, however, was not significantly debilitating, and was able to complete more than half of my estimated work, with another portion having to be set aside for the next sprint. In this next sprint, I will be able to get more work done, as due to the stay at home order, and my workplace now allowing us to work from home will give me some more time to set aside to get what I could not finish this passed sprint in the next.

# **Sprint #5**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| A1 | MainController | 4 | 5 |
| A2 | MainActivity/View | 3 | 6 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

This Sprint, the MainController was updated to further follow *Information Hiding* guidelines, and measures to reduce the amount of data accessible directly by the View/Activities.

Furthermore, the product had a few bugs that would cause the application to crash when launching views that used the MainController. These bugs were fixed, and allowed for progress to continue. Aside from these two additions, the older file structure, and original design had leftover classes that were cleaned up, and removed.

## Retrospective

This sprint, we had some trouble while hitting a roadblock, as part of our team had issues with testing, as some cropped up bugs prevented some testing of new components. These bugs were discovered, and the cause of which has been fixed. However, due to the setbacks of these bugs, not a lot of progress was made.

Next sprint we will be delving more into ensuring that the code base formerly containing the removed Objects, is squared up. Currently there is no errors or bugs detected from where references used to exist for the removed classes, but to ensure that everything is functioning normally, business logic that were in these modules will need to be validated and verified to be running smoothly without fail.

# **Sprint #6**

Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Story ID** | **Story / Task** | **Estimated**  **Hours** | **Actual**  **Hours** |
| A1 | SQLite DB | 6 | 6 |
| B1 | Book/Chapter | 4 | 4 |
| B2 | Database Modifiers | 3 | 2 |
| C1 | UI | 4 | ~ |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review

The product in this sprint was changed significantly due to poor planning in the previous sprints, thus the storage method was changed from JSON to SQLite, with a change from using a “Folder” and “File” type system, to using a Book/Chapter/Page system. These back end changes have been implemented in full, and are ready to be built off of in this next sprint. Next sprint, will be taking the framework laid out by the backend from this sprint, and creating the full UI for the end user to actually interact with and interface with the back end’s new storage system.

This sprint, though we had planned for the UI to be done, poor design on my part with what I thought was a solid base turned out to be a poor design, that was challenging to implement.

## Retrospective

I’m significantly happier about the progress that I’ve made in this sprint than the last sprint. Rather than continuing to try and implement difficult methodologies with my app, I’ve decided that I would give a fresh take from other resources for the app. The first big change was making the shift away from a “File/Folder” format, and using a simpler, more *“finite”* thinking. Rather than having a (theoretical) unlimited set of nested folders, I made the decision to redesign the app to use a simple nesting system, one that’s finite, and would more easily make sense for the users that the app is targeting to use.

As the app is geared towards users who read books, comics, webtoons, watch tv series, movies, and listen to podcasts, the metaphor of using a “notebook” or simply, a “book” made more sense instead of the tech-head’s file system paradigm.

This sprint, I’ve laid out a much clearer design path for the development to follow this next sprint, and the sprint after that.

I made a change from using JSON files for persistent storage, to using SQLite as a local database. In large part to the difficulty of storing the “File System” methodology behind the use case for the end user actually appears to be rather mundane, and inefficient for the usage of *logical* space. Due to the new design that is more resistant to scrutiny than before, I will be continuing the development of the project in at most 3 more sprints for the finished product. The only remaining needs for the app include:

* Redesigned UI
* Sorting Features (for the user to sort by Date, Alphabetic Order, or Reverse Order)
* Final Graphics (App Icon, in-app custom icons)

The most time-consuming features to add will be the UI and the Sorting features. However, after the next sprint, I plan on getting some further opinions on necessary features that users would suggest adding to app that they would find useful, and further ways to better the app, without leaving it in a perpetual development cycle.

From here on out, once the last sprint is done for the core functions, I can plan on testing with more devices, and while not necessarily shipping the .apk to the Play Store, I can share it with friends who found it necessary to have an application like this.