



**Library App
Initial Design Document
& User Manual
Version 1.1**

Table of Contents

1.0 Introduction	2
1.1 Purpose.....	2
1.2 Background Information	2
2.0 Features	3
2.1 Most Wanted Features	3
2.2 Additional Features.....	4
3.0 Design and Development.....	4
3.1 Suggested Design Pattern	4
3.2 Suggested Development	5
3.3 Suggested Backend Implementation	6
4.0 User Manual.....	7
4.1 Welcome!.....	8
4.2 Transferring Patches to an SD Card	10
4.3 Transferring Patches off an SD Card	11
4.4 Reordering Patches on an SD Card	12
4.5 Sorting Patches	13
4.6 Tagging Patches	14
4.7 Searching for Patches	16
4.8 Patch Version History	17
4.9 Creating Patch Banks	18
4.10 Even More Features	19
Glossary	20
Acknowledgements	20
Appendix I – Supplemental Information	21
Appendix II – All Gathered User Stories.....	22

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide a reference for the requirements and specifications that will be implemented by the application-to-be, henceforth referred to as “Library App”.


The overall purpose of the Library App is to allow users to easily organize, manage, and transfer patches created for the Empress Effects ZOIA. Currently, patches are retrieved via a third-party service known as PatchStorage, henceforth referred to as PS. The Library App will interact with PS through an existing PS *API* to fetch patches created for the ZOIA. Additionally, the Library App will check PS on start up to download the latest versions of patches that users had previously chosen to download. This makes it so users can always be working with the latest version of any given patch.

The Library App does not set out to create a replacement for PS but rather aims to be a supplemental application for ZOIA users to improve their workflow.

Terms that are italicized can be found in the Glossary section towards the end of this document.

1.2 Background Information

Patches created for ZOIA typically follow this standard naming convention¹:



```
0##_zoia_somename.bin
```

Figure 1: Standard ZOIA Patch Naming Convention

Where each patch typically begins with a three-digit identifier and ends with a file extension. The “_zoia_” prefix appears immediately after the three-digit identifier. Patches are binary files and use the .bin extension. However, patches stored on PS can also be stored as zip files, making use of the .zip extension. These zipped packages traditionally consist of at least one .bin file (the patch itself), and occasionally at least one .txt file (See Appendix 1, Figure 1, and Appendix 1, Figure 2). Assumptions about the contents of a .zip package should not be assumed, and the application should be able to process them without error. For the purposes of the Library App, it is suggested to drop the prefixes (both the identifier and “_zoia_”) and leave only the patch name to maintain simplicity. Lastly, uploads stored on PS that take on a file extension other than .zip or .bin should be able to be processed by the Library App without causing the program to crash.

ZOIA interfaces with patches via an SD card. The Library App must be capable of reading and writing to SD cards inserted into a computer to ensure that the application

¹ This was mentioned during a discussion with ZOIA community member Christopher H. M. Jacques.

meets the needs of users as files are placed onto and pulled from the before-mentioned SD cards.

Although the local filesystem can store as many patches as a computer's internal memory will allow, the ZOIA is only able to read 64 patches off an SD card at a time. Of note, if two patches share the same prefix only one will be loaded onto the ZOIA. As such, the Library App should not only detect duplicate prefixes, but it should also provide the functionality necessary to renumber patches with the same prefix to prevent this issue from occurring.

2.0 Features

2.1 Most Wanted Features

Following discussions with members of the ZOIA community and the subsequent creation of user stories (see Appendix II), the following features would be essential to the Library App:

- The ability to use the application offline.
- The ability to drag and drop patches from an SD card to the Library App for local storage.
- The ability to drag and drop patches from the Library App to an SD card for use on a ZOIA.
- The ability to modify tags (either add, remove, or rename) on patches locally stored within the Library App.
- The ability to pull the latest versions of patches from PS such that they can be stored within the Library App.
- The ability to pull the names of all currently uploaded ZOIA patches from PS.
- The ability to select and download any uploaded patch currently on PS directly through the Library App.
- The ability to store multiple versions of the same patch within the Library App.
- The ability to sort patches by various methods (date modified, author, tag, and prefix)
- The ability to search for patches by name, author, date, or tag.
- The ability to identify and work with both .zip and .bin file extensions.
- The ability to rearrange the order of patches on an SD card.
- The ability to create “banks” of patches that can maintain a user-defined order.
- The ability to minimize accidental deletion/overwriting of patches.
- The ability to visually view patches with conflicting prefixes or names.
- The ability to view patch information retrieved from PS within the Library App.

2.2 Additional Features

Following discussions with members of the ZOIA community and the creation of user stories (see Appendix II), the following features would be nice to have incorporated into the Library App:

- The ability to sort patches by multiple parameters (e. g. author and date modified).
- The ability to identify CPU usage per patch.
- The ability to identify and work with additional file extensions (different compressions .7z .tar, etc.).
- The ability to have the application switch automatically to the appropriate tab based on context (e.g. an SD card is inserted and the application switches to the SD card tab menu).
- The ability to assign ratings to patches and sort by said ratings.
- The ability to add custom text notes to patches, not specifically tags.
- The ability to keep track of locally made changes versus the original information stored on PS
- The ability to import folders of patches currently stored on a local filesystem so that they can make use of the features that the Library App provides.

3.0 Design and Development

3.1 Suggested Design Pattern

The nature of the Library App would call for a user-friendly *GUI* to allow users from all technical backgrounds to interact with the program with ease. It is suggested that the design follows the Model-View design pattern. This is to help keep the implementation of features separate from the presentation of said features. As the Library App evolves, the Model-View pattern would allow for concurrent work to be done without interference as developers can focus on different aspects of the application. This can be the *backend* implementation of features and interaction with the PS *API* (Model) or the implementation of a sleek *GUI* interface that makes use of the before-mentioned *backend* implementation (View). Changes to the Model should not have any significant effects on the View and vice-versa.

3.2 Suggested Development

Empress Effects approaches software development from a Test-Driven Development standpoint, henceforth referred to as TDD. TDD is recommended throughout the development of the *backend* implementation of the Library App. The formation of unit tests should be done before any implementation is decided upon, followed by an implementation for the unit tests that have been written. TDD repeats this cycle until the tests are exhaustive and the implementation is simple and correct.

For the *frontend* design, TDD should be followed as much as possible. *GUI* testing is practically different when compared to *backend* testing. Regardless, if TDD cannot be followed, tests should still be written to ensure the *GUI* performs as expected given a variety of inputs.

Features should be organized via costs and time, and development should not begin until this analysis is completed. This is to be completed

It is proposed that development follows a milestone-based approach. This is to ensure that progress can be measured and tracked such that the project does not fall behind schedule. The specifics of each milestone are subject to change, but a sample is outlined below:

- Milestone 1: Initial Implementation
 - The ability to interact with the PS *API* to retrieve patches.
 - The ability to identify both .bin and .zip files retrieved via the PS *API*.
 - An initial *GUI* with two windows that act as independent file explorers.
 - No interaction between the *backend* and *frontend*.
- Milestone 2: *GUI* Integration
 - All features from the previous milestone.
 - The ability to interact through a *GUI* to retrieve the names of all patches currently stored on PS.
 - The ability to download patches directly through the PS *API* that is not integrated within the *GUI*. Only a *backend* implementation should exist.
 - The ability to drag and drop files between two file explorers within the *GUI*.
 - The ability to renumber, search, and sort patches that is not integrated within the *GUI*. Searching should allow for a specifier (search by tag, search by author, search by prefix, search by date, search by name). Only a *backend* implementation should exist.

- Milestone 3: ZOIA Integration
 - All features from previous milestones.
 - The ability to detect an SD card containing ZOIA patches.
 - The ability to transfer patches from the SD card to the local filesystem.
 - The ability to download patches directly through the PS API that is integrated within the GUI.
 - The ability to renumber, search, and sort patches that is integrated within the GUI. Searching should detect the appropriate specifier whenever possible (search by tag, search by author, search by prefix, search by date, search by name).
 - The ability to have the application detect patches with duplicated prefixes.
- Milestone 4: Quality of Life Improvements (Release Candidate)
 - All features from previous milestones.
 - The ability to identify and work with both .bin files and .zip files within the GUI.
 - The ability to change the tags of patches stored locally within the Library App and on patches located on the SD card.
 - The ability to visually see patches with the same prefix within the GUI.
 - The ability to retrieve the latest versions of patches previously downloaded within the GUI.
 - The ability to see a version history of patches previously downloaded within the GUI.
 - The ability to view patch information retrieved from PS, if such information exists, within the Library App. The Library App should indicate if no information for a patch was available.

Additional milestones may be created should development meet the above milestones. All milestones assume correct tests and documentation are created for each feature.

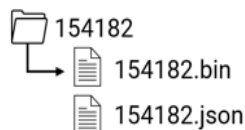
3.3 Suggested Backend Implementation

Following a discussion surrounding the pros and cons of various possible backend implementation, the suggested backend implementation for storage of ZOIA patches for use with the Library App is as follows:

- Patches will be stored within the AppData directory on Windows and the Library directory on macOS, under a directory titled .LibraryApp. For example, this means the path will be ~/Library/.LibraryApp on macOS.
 - For Windows, it is suggested that the sub-directory Roaming is used as files could move with a user so long as they are signed in under the same

account on a different computer. This means the path will be `~/AppData/Roaming/LibraryApp`

- For Linux, it is suggested that the user's home directory is used. It is also suggested that the LibraryApp directory is made hidden by using a "." in front of the directory folder. This means the path will be `~/.LibraryApp`
- Within the LibraryApp directory, there will be additional sub-directories. These sub-directories will be titled with a 6-digit integer number except for the special Banks sub-directory.
 - The Banks sub-directory will contain .json files. The schema is still being finalized.
 - Additional sub-directories will be titled after the unique patch id of patches retrieved from the PS API. This patch id is a 6-digit unique identifier used by PS.
- The patch sub-directories located within the LibraryApp directory will contain .bin and .json files.
 - The .bin files contained within a patch directory are the actual patch files that will be loaded onto a ZOIA.
 - For every .bin file, an accompanying .json file will exist. Within this .json file, metadata for the related patch will be stored. The frontend representation will interface with these .json metadata files to display human-readable data to the view of the Library App. As an example, this is the sample contents of a patch directory:



- The schemas for the .json metadata files and the bank files are being prepared and will be included in subsequent User Manual version revisions.

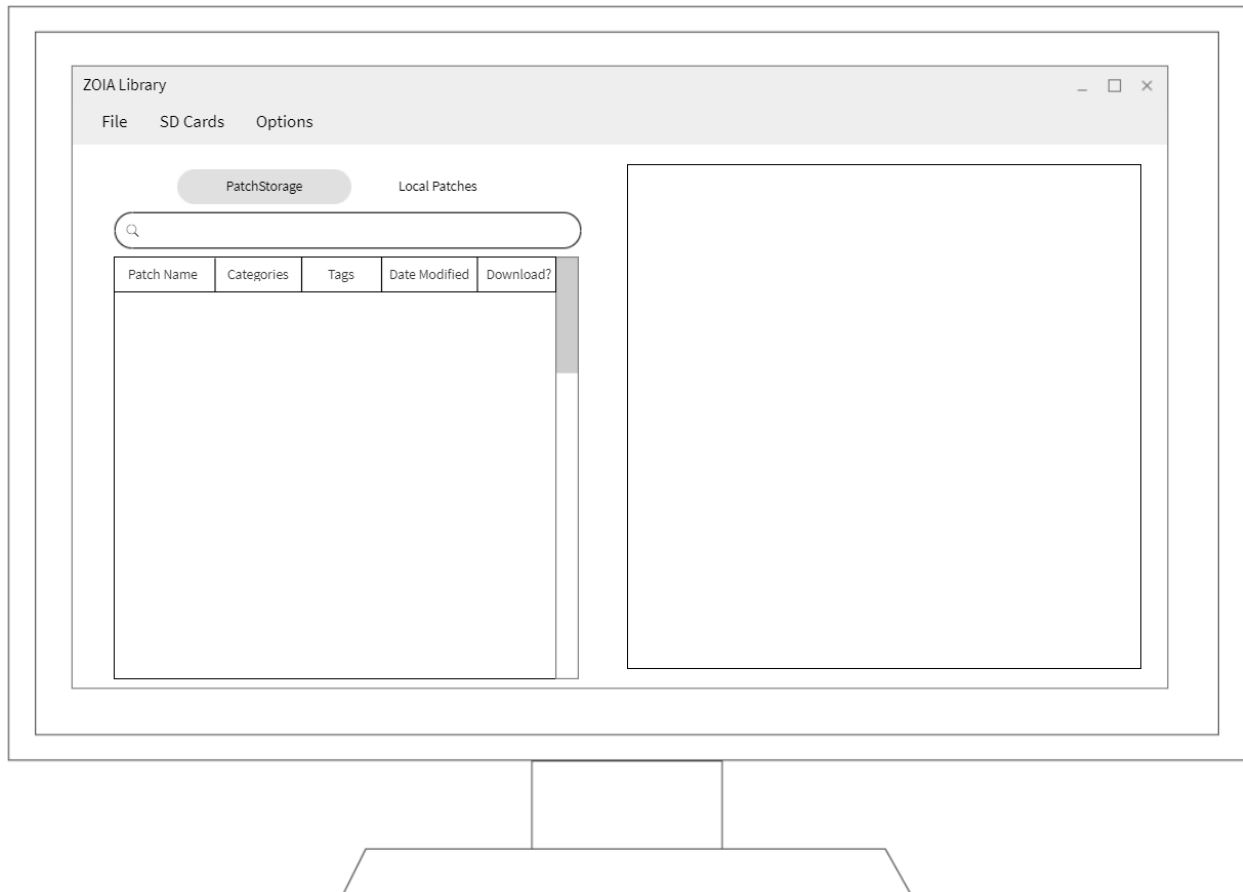
4.0 User Manual

Sections 4.1 to 4.10 outline a possible user manual under the assumption the features from Section 2.1 were fully implemented. It also presents various UI wireframes that help to illustrate how these features could be implemented within a *GUI* format. This should be considered as a standalone document in the context of the design document.

4.1 Welcome!

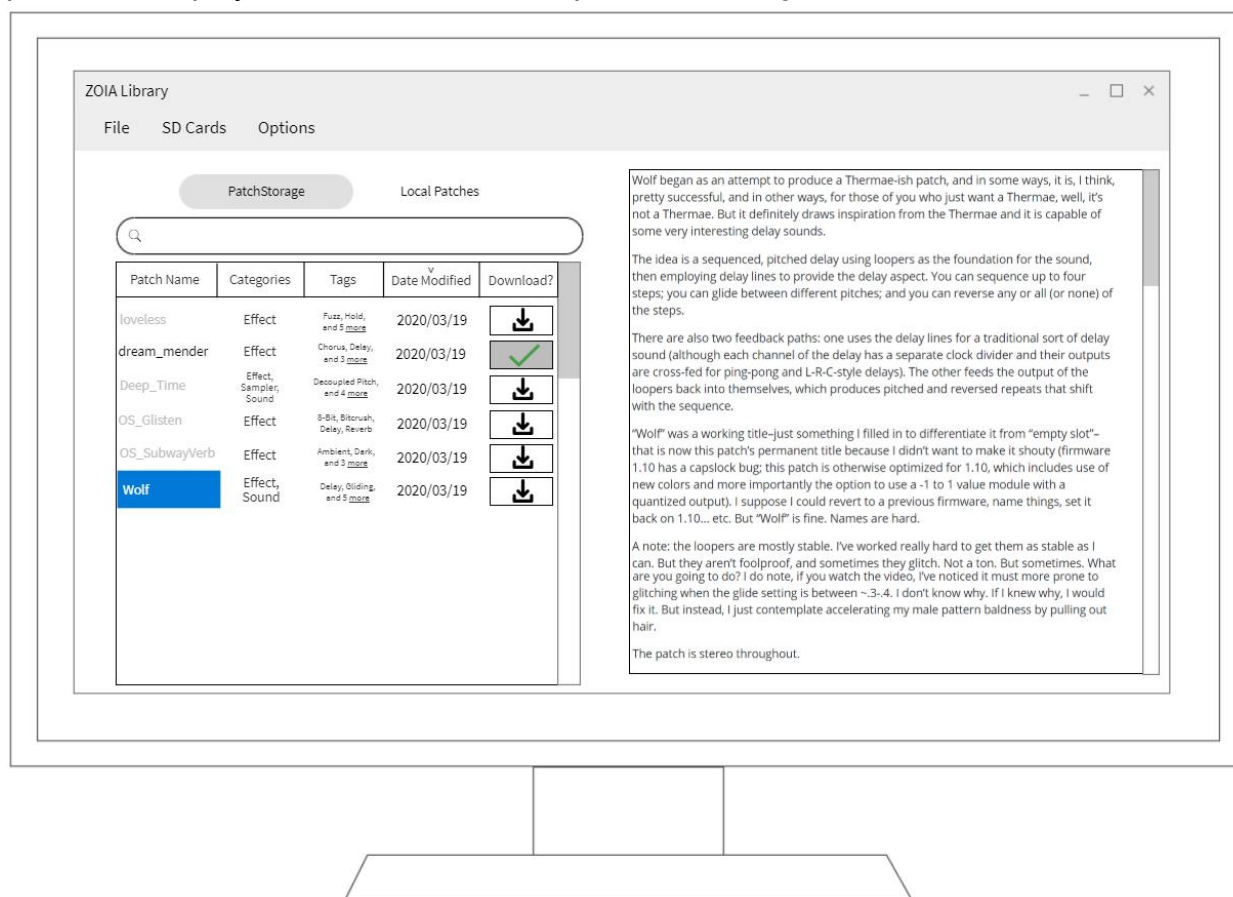
Welcome to the ZOIA Library App! The ZOIA Library App aims to make your life using ZOIA much easier. Whether you are a major patch creator or general ZOIA user, the Library App has features that can help all levels of users.

The ZOIA Library App is contained within a single executable. Double-clicking on the LibraryApp.exe file will start the app, where you will be greeted with the following screen:




Please note, that if this is not the first time you are using the app you may be prompted to download the latest versions of patches that were previously downloaded within the Library App. Additionally, if you are not connected to the internet you will not be able to make use of any of the PatchStorage features offered+ by the application.

If connected to the internet, the app will begin populating the left side of the screen with all ZOIA patches currently uploaded on PatchStorage. From here, clicking to select any patch will display information about the patch on the right side of the screen:



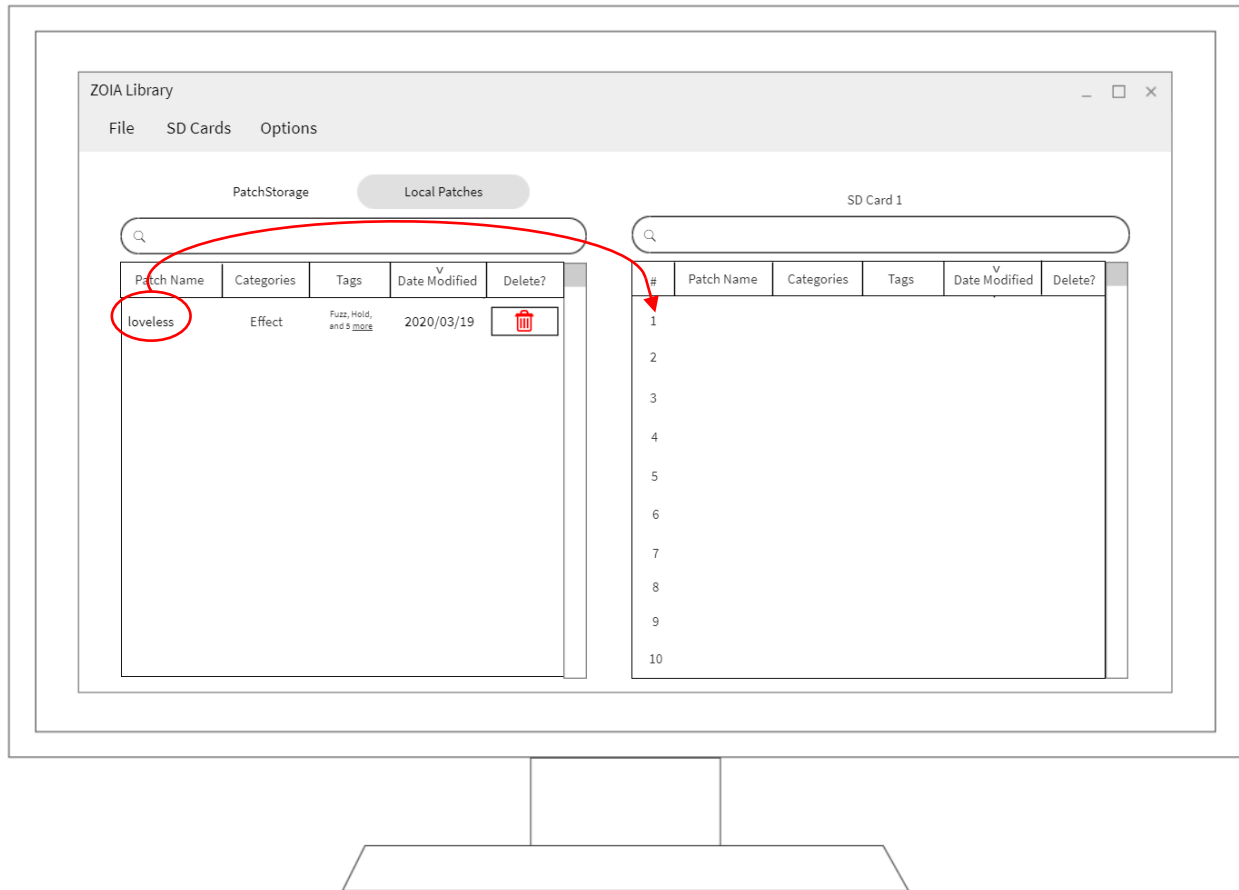
Once you have found some patches you are interested in, you can download them using the download button. Once a patch has been downloaded, the patch name will appear black instead of gray and the download button will no longer be available:

dream_mender Effect Chorus, Delay, and 3 more 2020/03/19 

Now that you have some patches downloaded, it is time to transfer them to an SD card!

4.2 Transferring Patches to an SD Card

Transferring patches has been streamlined to make it as easy as possible. The first thing to do is to switch your view to the Local Patches view by clicking the button at the top of the left side of the display. Next, use your mouse to select the patch you want to transfer over and click and drag it over to the desired slot on the SD Card view located on the right side of the screen. For example, to add a patch to slot 1 on an SD card, drag it to slot 1:

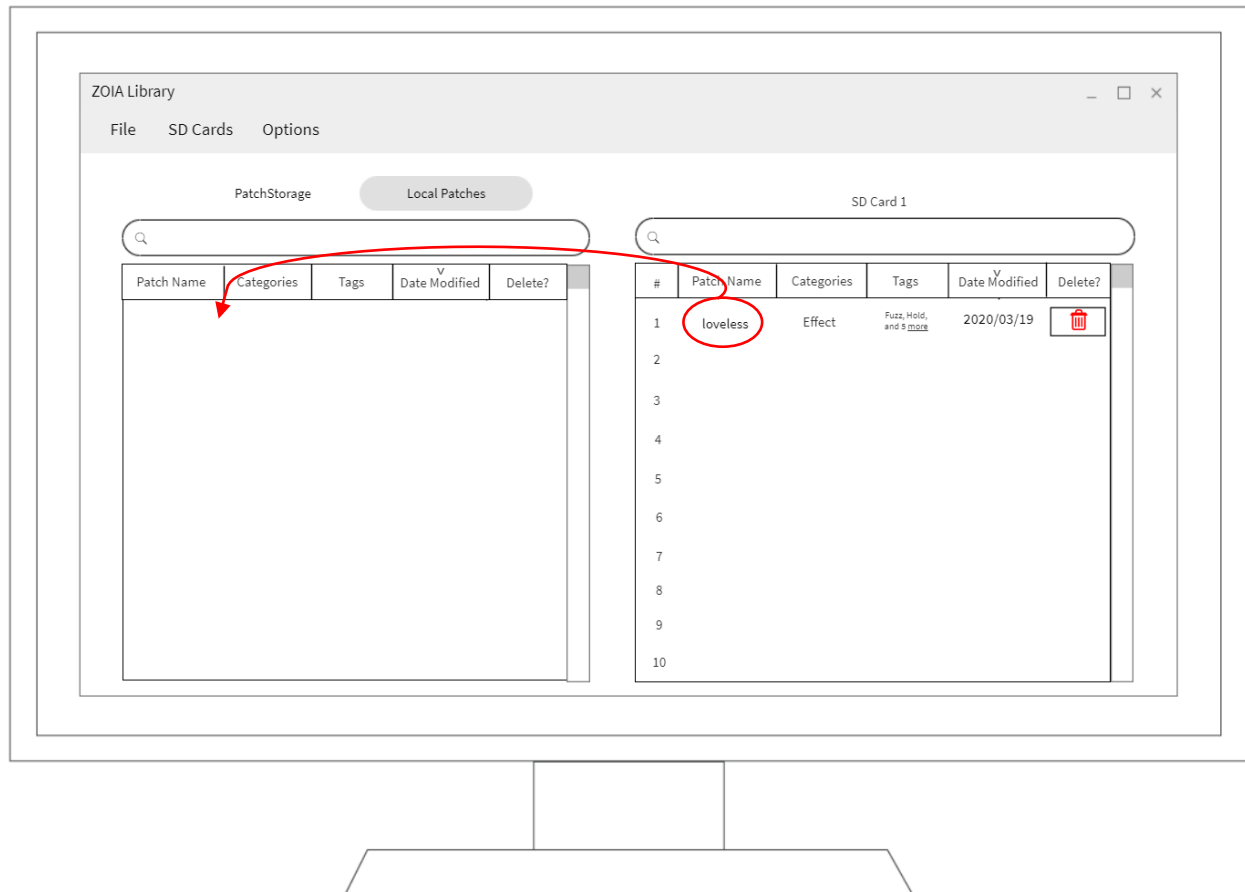


Some helpful notes:

- If there a patch on the SD card that has the same name as the one you are transferring, the Library App will ask if you would like to overwrite it.
- You can select multiple patches by holding down the Ctrl or Command key and then clicking on each patch you would like to select. Once you have selected all the patches, you can transfer them all by dragging them over to the SD Card view. Any naming conflict will appear one at a time where you can manually approve or deny the overwriting of patches.
- After transferring a patch, it will remain in the Local Patches view until it is deleted. Deletion can be done by hitting the delete button. Once a patch is deleted, there is no way to recover it unless it is still on PatchStorage!

4.3 Transferring Patches off an SD Card

Transferring patches off an SD card follows a similar procedure. Use your mouse to select the patch you want to transfer over and click and drag it over to the Local Patches view on the left side of the screen:






Some helpful notes:

- If there a patch that is locally stored that has the same name as the one you are transferring, the Library App will ask if you would like to overwrite it.
- You can select multiple patches by holding down the Ctrl or Command key and then clicking on each patch you would like to select. Once you have selected all the patches, you can transfer them all by dragging them over to the Local Patches view. Any naming conflict will appear one at a time where you can manually approve or deny the overwriting of patches.
- After transferring a patch, it will remain on the SD Card view until it is deleted. Deletion can be done by hitting the delete button. Once a patch is deleted off an SD Card, it is gone for good unless it was already in the local filesystem.

4.4 Reordering Patches on an SD Card

The ZOIA Library App allows you to order patches on an SD card. It is as simple as clicking and dragging the patches around on the SD Card view until they are in the slots you want them to be in:

SD Card 1

#	Patch Name	Categories	Tags	Date Modified	Delete?
1	dream_mender	Effect	Chorus, Delay, and 3 more	2020/03/19	
2	Deep_Time	Effect, Sampler, Sound	Decoupled Pitch, and 4 more	2020/03/19	
3	loveless	Effect	Fuzz, Hold, and 5 more	2020/03/19	

If you want to swap the position of two patches, dragging a patch over an already occupied slot will automatically swap the positions of the two patches. For example, if you wanted to swap “dream_mender” and “Deep_Time” you could drag one on top on the other:

#	Patch Name	Categories	Tags	Date Modified	Delete?
1	Deep_Time	Effect, Sampler, Sound	Decoupled Pitch, and 4 more	2020/03/19	
2	dream_mender	Effect	Chorus, Delay, and 3 more	2020/03/19	

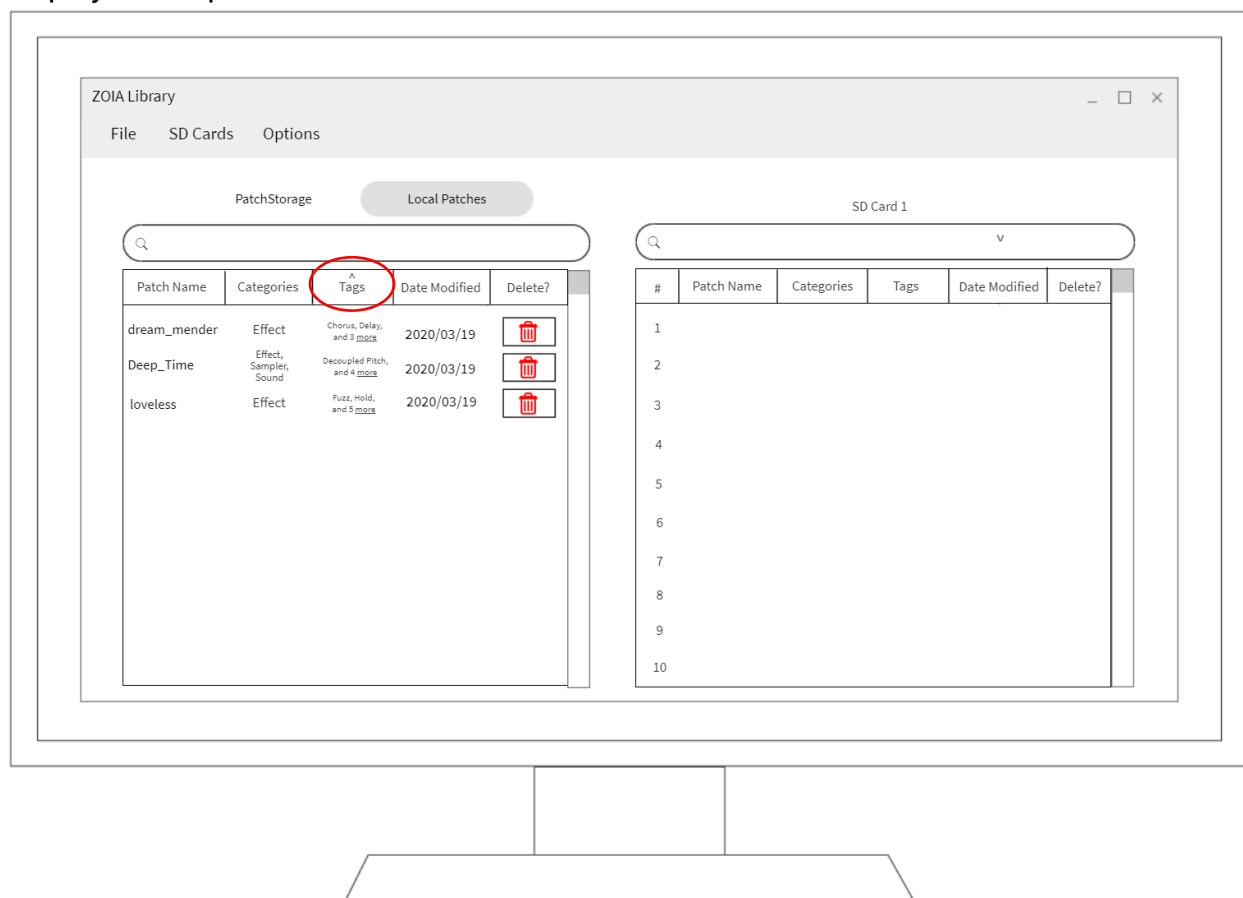
Additionally, if dragging a patch towards the bottom or top of the SD Card view will automatically scroll the panel so you can reach the slot you are looking for quickly and easily.

If you have multiple SD cards inserted into your computer, you can switch between them by navigating to the SD Cards menu item and clicking on it. The ZOIA Library App will automatically scan for any inserted SD cards and list them so you can quickly switch between cards:

SD Cards	Options
	SD Card 1
	SD Card 2
	SD Card 3

4.5 Sorting Patches

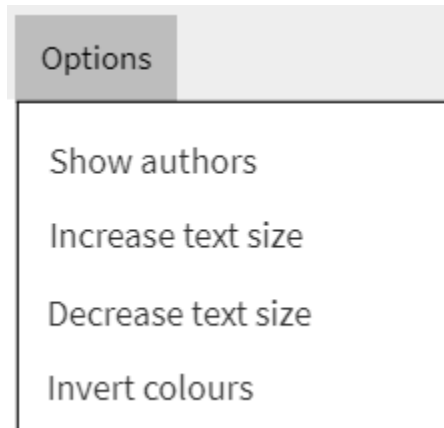
The ZOIA Library App offers a variety of features to organize patches in ways that best fit your needs. Patches can easily be organized via prefix, primary tag, and date modified simply by clicking on the respective box in patch explorer. By default, the patches are organized by Date Modified starting with the most recently modified patch and descending downwards (denoted by the downward arrow). Clicking on Patch Name will sort the patches in that view in alphabetical order. Clicking on Categories will sort the patches by the first category listed. Clicking on Tags will sort the patches by the first tag displayed in alphabetical order:






Clicking on the same box will reverse the way the sorting has occurred (the arrow will flip directions). For example, if you were to click on Tags again:

Patch Name	Categories	Tags	Date Modified	Delete?
loveless	Effect	Fuzz, Hold, and 5 more	2020/03/19	
Deep_Time	Effect, Sampler, Sound	Decoupled Pitch, and 4 more	2020/03/19	
dream_mender	Effect	Chorus, Delay, and 3 more	2020/03/19	

Beyond the sorting options mentioned before, patches can also be sorted by author. Authors are hidden by default, but they can be made visible by navigating to the Options menu and selecting Show authors:

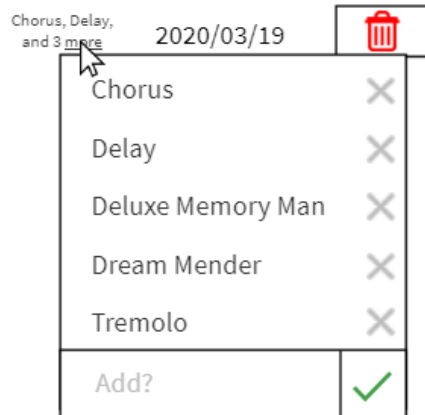


A new Authors box will appear in all views, and the Authors box can be clicked like the Patch Name, Categories, Tags, and Date Modified boxes to sort patches alphabetically based on an author's first name:

Patch Name	Author	Categories	Tags	Date Modified	Delete?
Deep_Time	Christopher H. M. Jacques	Effect, Sampler, Sound	Decoupled Pitch, and 4 more	2020/03/19	
loveless	meagher	Effect	Fuzz, Hold, and 5 more	2020/03/19	
dream_mender	nathanvassiliades	Effect	Chorus, Delay, and 3 more	2020/03/19	

4.6 Tagging Patches

Tags are an important part of any patch. The ZOIA Library App provides many tagging features to help make the experience as easy as possible. Whenever a patch has more tags than can be fit on the screen, you can hover over the “more” text to have all tags related to that patch appear:



Clicking on the X beside any tag will remove it from the list of tags for that patch. Additionally, you can add your own tags by typing them into the textbox at the bottom of the tag list and clicking on the green checkmark.

Some helpful notes:

- Tags in the tag list are sorted alphabetically to help keep things organized.
- Do not worry about adding a tag that is already present. The ZOIA Library App will let you know if that happens, so you do not end up with 30 different “Effect” tags.
- Even if the tag is not shown without hovering over “more” when you search by tag the ZOIA Library App will still include any patches with the tag you entered within the search results.

4.7 Searching for Patches

The ZOIA Library App lets you quickly search for patches, whether they are stored on PatchStorage, locally downloaded, or stored on your SD card. Just enter what you are searching for into the appropriate search box and the list of patches will update to match what you typed in. Whether it be a patch name:

Patch Name	Author	Categories	Tags	Date Modified	Delete?
dream_mender	nathanvassiliades	Effect	Chorus, Delay, and 3 more	2020/03/19	

A date:

Patch Name	Author	Categories	Tags	Date Modified	Delete?
Deep_Time	Christopher H. M. Jacques	Effect, Sampler, Sound	Decoupled Pitch, and 4 more	2020/03/19	
loveless	meagher	Effect	Fuzz, Hold, and 5 more	2020/03/19	
dream_mender	nathanvassiliades	Effect	Chorus, Delay, and 3 more	2020/03/19	

Or a category:

Patch Name	Author	Categories	Tags	Date Modified	Delete?
Deep_Time	Christopher H. M. Jacques	Effect, Sampler, Sound	Decoupled Pitch, and 4 more	2020/03/19	

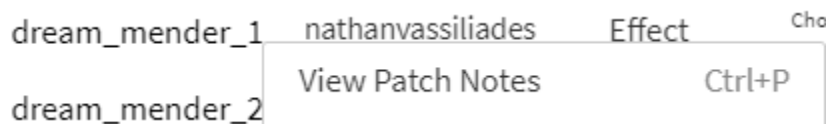
Searches can also be conducted by tag or by author. These features will help to make sure you can find the patches you are looking for quickly and easily.

4.8 Patch Version History

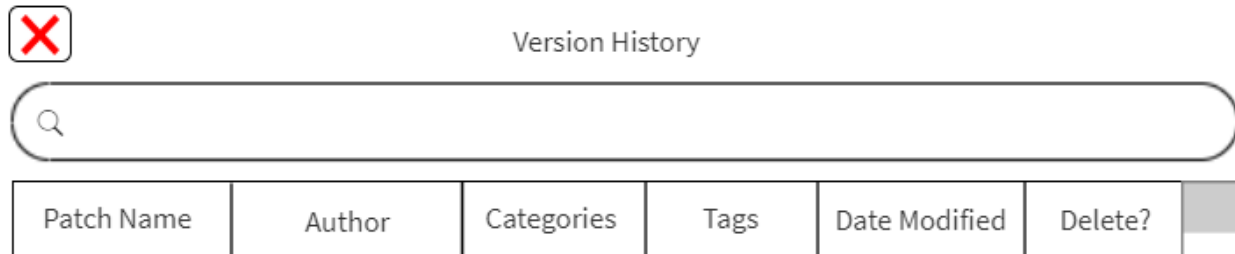
The ZOIA Library App allows you to keep track of past versions of patches if they have been previously downloaded to your Local Patches. Although only the most recent version is normally visible, you can right-click on any patch with a folder icon to the left of its patch name and then click on View Version History to explore past versions of that patch:



Continuing to the Version History page will replace the Local Patches view with a new Version History view, where you can scroll through all available patch versions by date modified. Each patch will feature the version (if available) as a number at the end of the name of the patch. These earlier versions can be renamed, tagged, or moved to an SD card like any other patch, but they can also show the patch notes (if available) for a given version by right-clicking on a patch version and selecting View Patch Notes:



once you are finished looking at the history of a patch, simply hit the X button to the left of Version History to return to the Local Patches view:

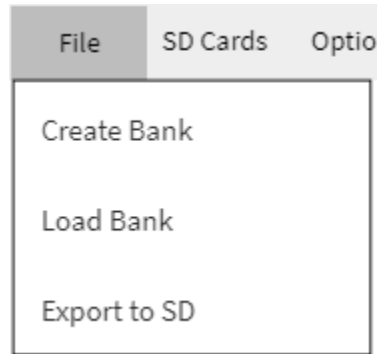


Some helpful notes:

- Only the most recent version of a patch will show up in the Local Patches view. To access previous versions, you must click on View Version History after right-clicking on a patch.

4.9 Creating Patch Banks

The ZOIA Library App has features to let you save banks of patches. Each bank is an ordering of up to 64 patches that can be loaded at any time so that they can easily transfer to an SD card. To create a bank, navigate to File and click on Create Bank:



From here, you will be presented with a typical SD Card view (only now it is called Bank Creator), where you can drag and drop patches into the numbered slots. Once you are finished, you can export the bank to an inserted SD card using the Export to SD option under File, or you can save your bank for later. To save a bank, simply click on the floppy disk icon in the Bank Creator view:



You can store more than one bank at a time within the ZOIA Library App, but you can only have one bank loaded on an SD card at a time. If you need to switch banks quickly, be sure to keep the ZOIA Library App handy to make the process as quick and painless as possible.

4.10 Even More Features

On top of the features already covered in previous sections, the ZOIA Library App has some additional features ready-to-go, including support for patches on PatchStorage that use the .zip extension and accessibility options including colour inversion and font resizing. We encourage you to provide feedback on things you want to see changed or features you would like to see added. Thank you for being an awesome member of the ZOIA community!

Glossary

API – An **A**pplication **P**rogramming **I**nterface. It provides a means for applications to communicate via a created interface. The PS API can be viewed in full at the following link: <https://patchstorage.com/docs/>

Backend – The underlying logic required to make the application function. Typically, does not produce a user-friendly output.

Frontend – The view used to present the output returned by the *backend* logic in a user-friendly medium.

GUI - A **G**raphical **U**ser **I**nterface. It helps users take advantage of an application's *backend* implementation without requiring them to write any code. Many programs, such as Microsoft Word and Google Chrome, feature GUIs.

Acknowledgements

This document would not have been possible without the help and support of the ZOIA community. Of note, both Matt Allen and Mike Moger provided user stories from a variety of sources that helped shape the suggested features presented in Section 2.0 of this document.

Thank you!

Appendix I – Supplemental Information

Name	Type	Compressed si...	Passwor...	Size	Ratio	Date modified
049_zoia_loop_sync.bin	BIN File	3 KB	No	32 KB	93%	2014-12-31 11:00 ...
050_zoia_loop_sync_s.bin	BIN File	3 KB	No	32 KB	93%	2014-12-31 11:00 ...
051_zoia_loop_sync_MIDI...	BIN File	3 KB	No	32 KB	92%	2014-12-31 11:00 ...
052_zoia_loop_sync_sMIDI...	BIN File	3 KB	No	32 KB	93%	2014-12-31 11:00 ...
Loop sync rev1.txt	Text Document	4 KB	No	8 KB	59%	2020-05-03 8:49 PM

Figure 1: Example PS .zip Contents

This is my go at the Mojo Hand FX Dream Mender, which is their go at the 70's style Deluxe Memory Man.

It is a delay with either chorus or tremolo. It is known for being able to self oscillate.

You can turn on the chorus or tremolo, or have them both on(which you can't do on the original).

L Mono output -> Stereo RL Output

↓

Delay -> Stereo RL Output

↓↓↓

↓↓ Chorus -> Stereo RL Output

↓↓

↓ Tremolo -> Stereo RL Output

↓

VCA

↓

Back to original delay (can control oscillation)

Figure 2: Example PS Patch Description

Appendix II – All Gathered User Stories

User stories are grouped by the individual that presented said story. Where context was provided based on which viewpoint the story was being told from, that is indicated at the start of the user story. The order of the user stories below does not reflect any specific order of importance. All user stories that needed to be broken down are indented below the epic user story that required said breakdown:

Christopher

As a power user, I want to have the application support .zip files so that I can continue to use my current style of workflow.

- As a power user, I want to have the application support the ability to see inside .zip files so I can so that I can continue to use my current style of workflow.
 - As a power user, I want to have the application unzip files automatically but keep the contents grouped together so that I can easily view the entirety of a .zip file and its contents easily.
- As a power user, I want to be able to view different versions of the same patch so that I can view previous iterations.
 - As a power user, I want to see what has changed between patch versions via their update notes so that I can decide which patch version to use.
 - As a power user, I want to add multiple versions of the same patch to my ZOIA at the same time so that I can decide which version works best for my needs.
 - As a power user, I want to be able to sort patches by the date they were updated so that I can quickly see the latest version of a given patch.
- As a user, I want to have the application detect duplicate patch prefixes so that I can visually see which patches might have trouble opening on a ZOIA.
- As a power user, I want to add and delete tags associated with patches stored locally so that I can have control over the customization of patches locally.
 - As a power user, I want to be able to create custom tags so that I can organize patches in a way that works for my needs.
 - As a power user, I want to be able to delete tags on patches so that I can customize patch tags to reflect how I would have tagged those patches.
 - As a power user, I want to be able to rename tags so that I can customize the organization of patches locally.
- As a power user, I want to be able to sort patches by multiple parameters, such as by author and date modified, so that I can organize my patches more easily.

Jesse (Former ZOIA user)

- As a user, I want to be able to have options with regards to renumbering patches, so that I can have the patches load in the slot on the ZOIA I want.
- As a user, I want to know if I have the latest version of a patch so that I know I am not working with a previously buggy version of a patch.
- As a user, I want to be able to view if a patch has documentation so that I can decide which patch I should and should not give a try.

Jeremy (Red Means Recording on YouTube)

- As a user, I want to be warned if a patch were to be deleted because they share a name so that I can avoid losing progress on patches I have worked on.
- As a user, I want to be able to avoid the ZOIA “patch needs number” restriction so that I can get to work using patches faster.
 - As a user, I want to be able to have the application automatically add a number to patches that do not have an appropriate number.
 - As a user, I want to be able to mass renumber patches so that I can make sure I do not see the “patch needs number” restriction.

John

- As a developer, I want to be able to use the application offline, so that I can easily organize patches without needing to download any new ones.
- As a developer, I want to have the application capable of supporting concurrency so that the application feels responsive and easy to use.
 - As a developer, I want to be able to use the application without waiting for patches to be retrieved from PatchStorage so that I can work on other things while patches are updated or retrieved.
- As a developer, I want the application to pull the names of patches from PatchStorage without downloading them so that I do not have patches I will never use downloaded on my computer.
 - As a developer, I want to manually select which patches to download from PatchStorage so that I only have the patches I want to use saved locally.

Mike

- As a developer, I want to be asked to download patches on start-up so that I can have all available patches ready to transfer to a ZOIA.
- As a power user, I want to be able to sort patches so that I can keep things organized and find patches that I am looking for.
 - As a power user, I want to be able to sort patches by prefix number so that I can easily organize patches by prefix number to quickly find patches I am looking for.
 - As a power user, I want to be able to sort patches by date modified so that I can easily organize patches by date modified to quickly find the most recent patches that I have worked with.
 - As a power user, I want to be able to sort patches by patch name so that I can easily find a patch that I am looking for.
 - As a power user, I want to be able to sort patches by an author's name so that I can quickly find patches I or others have created.
 - As a power user, I want to be able to sort patches by tag so that I can find patches that contain the tags I want to work with.
- As a user, I want to be able to dump patches onto and off an SD card so that I can easily manage patches on both my computer and my SD card.
- As a developer, I want to be presented with the relevant tab on start, depending on if an SD card is or is not inserted into the computer, so that I can save time using the application.
- As a power user, I want to download updates to locally stored patches without overwriting previous versions, so that I can have the latest versions of patches while maintaining a history of previous versions.
- As a user, I want to be able to renumber patches so that I can have them appear on the ZOIA in a specific organization I decided on.
 - As a user, I want to be able to renumber patches based on the name of the patch so that I can have patches appear on the ZOIA alphabetically.
 - As a power user, I want to be able to renumber patches based on the modification date so that I can have the patches appear on the ZOIA in an order that fits my needs.

Scott (Sound Isles)

- As a power user, I want to be able to import folders of patches I currently have stored on my computer so that they can be organized easily within the application.
- As a power user, I want to be able to get information about what a patch contains so that I can know what the patch is for ahead of time.
 - As a power user, I want to be able to see the type of patch (e.g. utility type of patch like a MIDI buddy for CB) so that I can know what I am working with before I put it on a ZOIA.
 - As a power user, I want to be able to see what the patch needs to work (e.g. if it needs a MIDI keyboard) so that I can know what I am working with before I put it on a ZOIA.
- As a power user, I want to be able to group patches by tag so that I can search for that tag later when I want to use those patches.
- As a power user, I want to be able to read patch notes and patch descriptions so that I can get a feel for the patch before I use it.

Ian (Demoer)

- As a user, I want to be able to favourite patch so that I can quickly find patches that I enjoyed working with in the past.
- As a user, I want to be able to prepare full patch loadouts so that I can have the patches appear in that specified order on a ZOIA.
 - As a user, I want to be able to prepare banks of patches so that I can organize patches the way I would want them to appear on a ZOIA.
 - As a user, I want to be able to quickly transfer these banks of patches to a ZOIA so that I can easily transfer banks during use.
- As a user, I want to be able to group patches by category so that I can quickly browse for patches that I am looking for.
- As a user, I want to be able to access patch notes for patches so that I choose which patch I want to give a try.