

# Online Scientific Reference System's Documentation

Lisbeth Cardoso Guerra

Kyle Walsh

EGN 4912 Engineering Directed Independent Research

CIS 4914 Senior Project

Advisor: Dr. Mark Schmalz, e-mail: [mssz@cise.ufl.edu](mailto:mssz@cise.ufl.edu)

Department of CISE, University of Florida

Gainesville, Florida

October 31st, 2017

## Structure of SRS

The online Scientific Reference System (SRS) currently includes several views that allow the user to see and interact with data differently. Each view can be accessed from the SRS tab of the application.



Figure 1: “View Select Page.”

### Directory List

The directory list view presents directories and notefiles similarly to how a normal file system would. Notefiles represent the basic file storage units within the system and contain notecards. By clicking on a directory you are directed to a new location within the file system, which can be tracked by referring to the “Path” field at the top right of the SRS window. By clicking a notefile, you are able to access the notefile and can then proceed to a main menu with each of the notecards stored within that notefile.



Figure 2: "Directory List."

## Notefile List

The notefile list view presents all the notefiles in the system, regardless of what directory they're stored in. You can access these notefiles by clicking them, in the same way you would in the directory list.



Figure 3: "Notefile List."

## Image Archive

The image archive view presents all the image files in the system, regardless of what notecard they are associated with. The images can also be clicked to expand them and see additional details.



Figure 4: “Image Archive.”

## Video Archive

The video archive view presents all the video files in the system, regardless of what notecard they are associated with. The thumbnails can be clicked to open a modal with additional details and play the video clip.

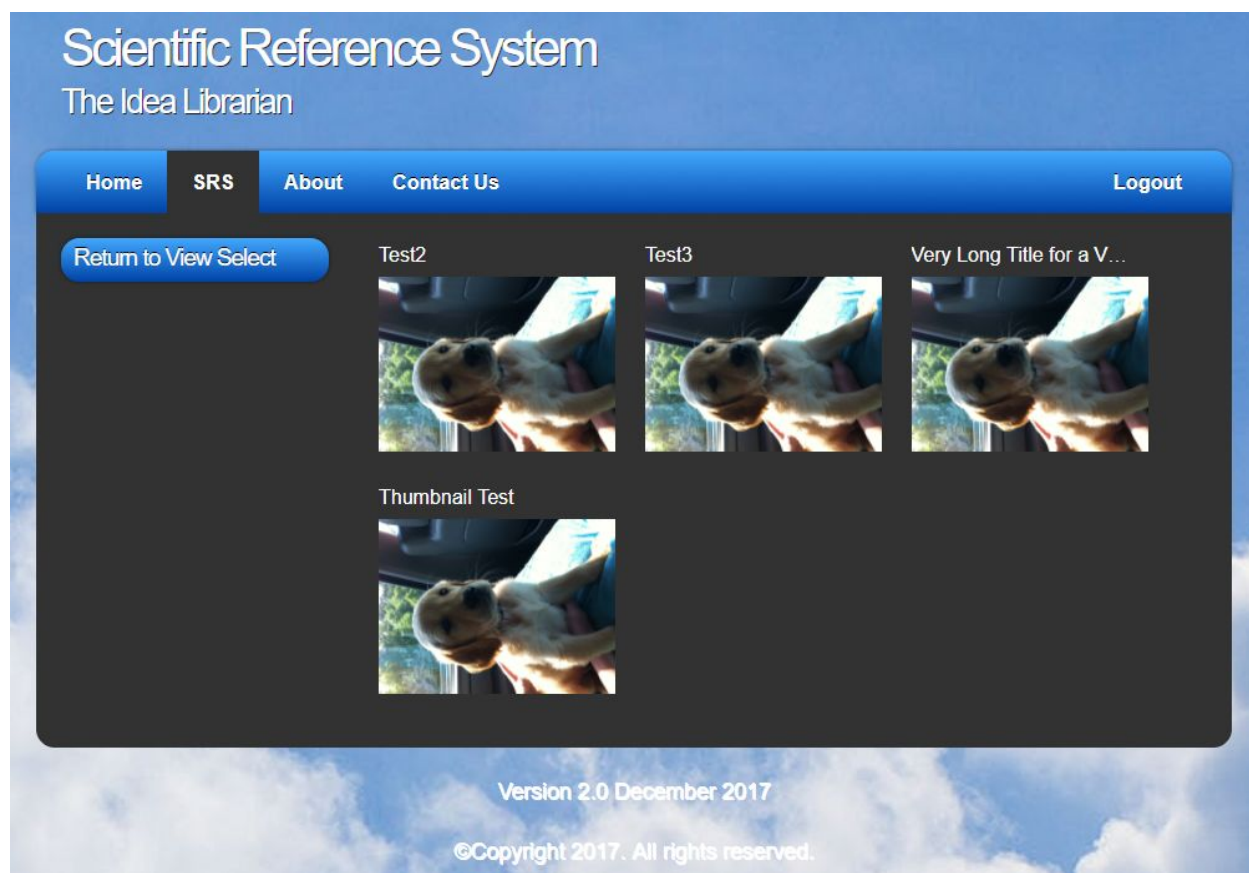


Figure 5: "Video Archive."

## Audio Archive

The video archive view presents all the audio files in the system, regardless of what notecard they are associated with. The buttons can be clicked to open a modal with additional details and play the audio clip.





Figure 6: “Audio Archive.”

## Document Archive

The document archive view presents all the documents in the system, regardless of what notecard they are associated with. The links can be clicked to download the file.

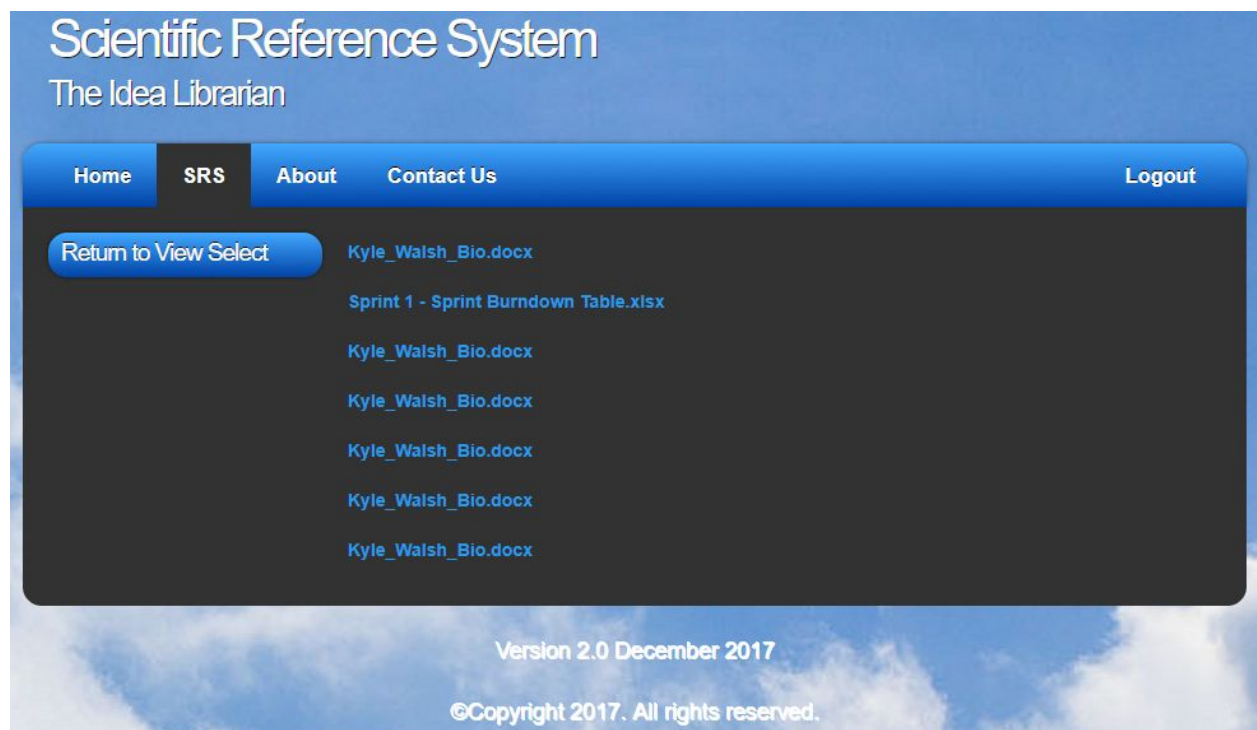


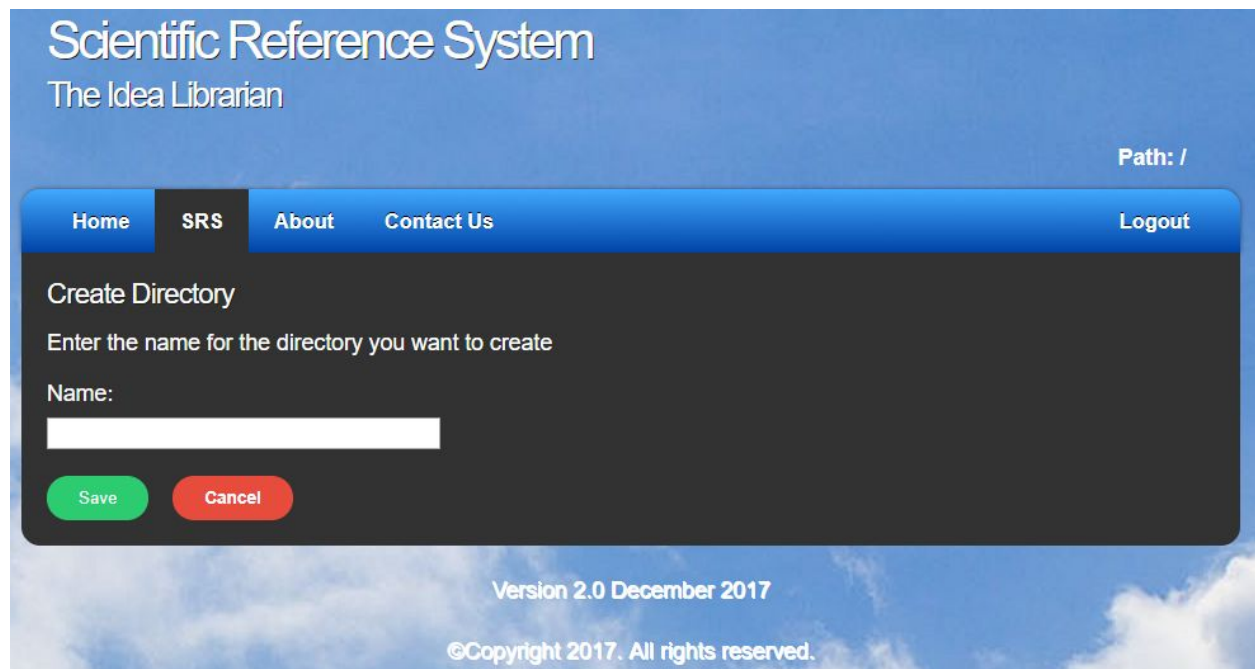
Figure 7: “Document Archive.”

## Functions of SRS

The online Scientific Reference System (SRS) currently has the following features: creation of directories, creation of notefiles, creation of notecards, import and export of notecards, notecard search functionality, addition of equations (both inline and display) to notecards, addition of images to notecards, addition of videos to notecards, addition of audio to notecards, addition of documents to notecards, and the ability to duplicate notecards and select what components you would like to keep.

### Create Directory

SRS allows to create directories. Directories permit to group and organize information within folders. A directory can contain notefiles and other directories. A directory has a name, an author (which is the user that is logged in), a date of creation, and a parent directory. By default, the parent directory is called Home.



The screenshot shows the 'Create Directory' form within the Scientific Reference System (SRS) interface. The header includes the title 'Scientific Reference System' and the tagline 'The Idea Librarian'. A navigation bar contains links for 'Home', 'SRS', 'About', and 'Contact Us', along with a 'Logout' button. The form itself is titled 'Create Directory' and prompts the user to 'Enter the name for the directory you want to create'. It features a text input field labeled 'Name:' and two buttons: 'Save' (green) and 'Cancel' (red). The footer of the interface displays 'Version 2.0 December 2017' and '©Copyright 2017. All rights reserved.'.

Figure 8: “Create a directory.”



## Create Notefile

Besides creating directories, SRS allows for the creation notefiles. Notefiles are contained within the current directory you have selected. They have name, author (which is the user that is logged in), label, body, date of creation, keywords and a list of notecards associated. A note file can contain zero, one or many notecards. So far, all of these fields are only text-based.

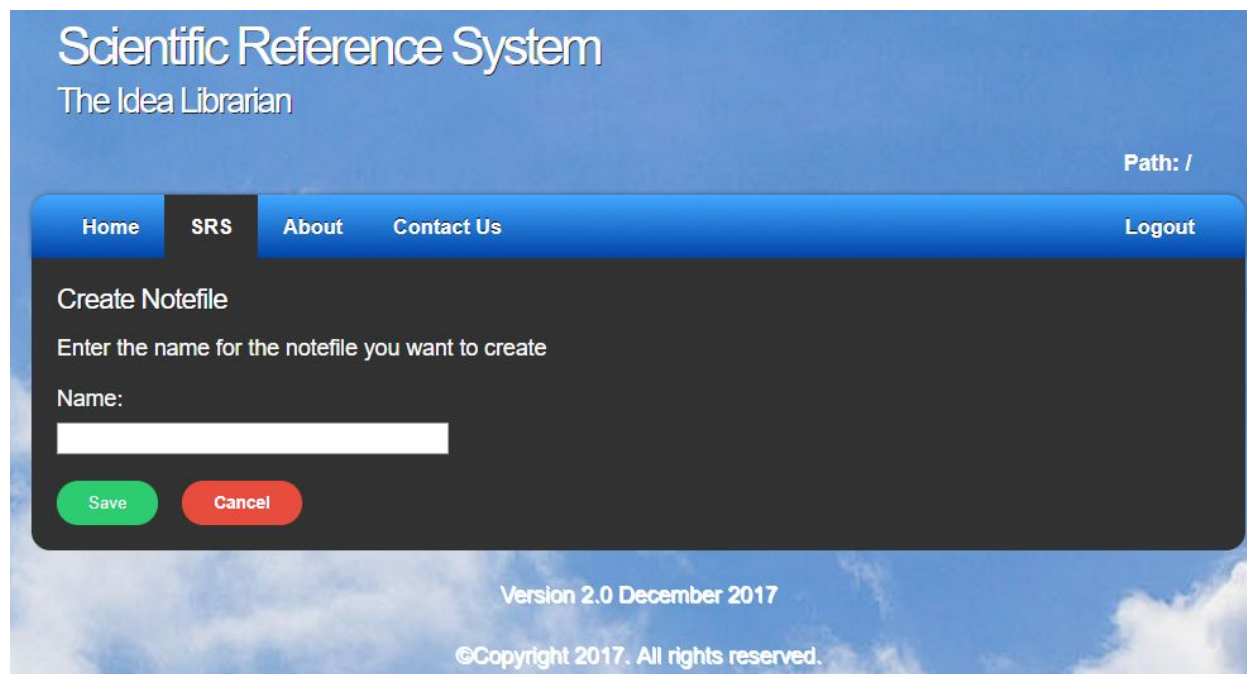
The screenshot shows the 'Scientific Reference System' interface with the subtitle 'The Idea Librarian'. The background is a blue sky with clouds. At the top right, it says 'Path: /'. Below this is a navigation bar with links: 'Home', 'SRS' (which is highlighted), 'About', and 'Contact Us'. On the far right of this bar is a 'Logout' link. The main content area is a dark grey box titled 'Create Notefile'. Inside this box, it says 'Enter the name for the notefile you want to create'. Below this is a label 'Name:' followed by a white text input field. At the bottom of the box are two buttons: a green 'Save' button and a red 'Cancel' button. At the very bottom of the page, it says 'Version 2.0 December 2017' and '©Copyright 2017. All rights reserved.'

Figure 9: “Create a notefile.”




## Notecards

The SRS also shows the contents of notecards. Notecards contain text-based information regarding a reference, such as name, author (which is the user that is logged in), label, body, date of creation and list of keywords associated. As shown in Figure 10, and similar to SQUARENOTE, SRS shows only some of these fields, particularly name (title), list of keywords and body. Although they are not specifically stored as part of the schema for the notecard, each notecard also has images, videos, audio, equations, and documents that are associated with it as separate database objects in the database and displayed on the notecards page.

# Scientific Reference System

The Idea Librarian

Path: /NewNote/OS\_Notecard1

[Home](#)[SRS](#)[About](#)[Contact Us](#)[Logout](#)[Return to View Select](#)[Add Equation](#)[Add Image](#) [Add Video](#) [Add Audio](#) [Add Document](#) [Duplicate Notecard](#)[Back](#)[Help](#)

Title:

OS\_Notecard1

Keywords:

OS, memory

Body:

Memory management is the functionality of an operating system which handles or manages primary memory and moves processes back and forth between main memory and disk during execution. Memory management keeps track of each and every memory location, regardless of either it is allocated to some process or it is free.

Equations:

$$\sqrt[3]{2} \times \int_1^2 2y$$

Images:

Puppy4



Puppy5



Videos:

Thumbnail Test



Audio:

[Voice](#)[My Audio](#)[Test](#)[Audio](#)

Documents:

[Kyle\\_Walsh\\_Bio.docx](#)

Version 2.0 December 2017

©Copyright 2017. All rights reserved.

Figure 10: “Notecard layout.”

As seen in Figure 11, one or more notecards can be grouped under one notefile. Similarly to SQUARENOTE’s design, only five notecards will be shown on cascade at any time. Users can move up and down the list of notecards using the keyboard's up and down. The list will be updated accordingly and the users will be able to see all the notecards in the list.

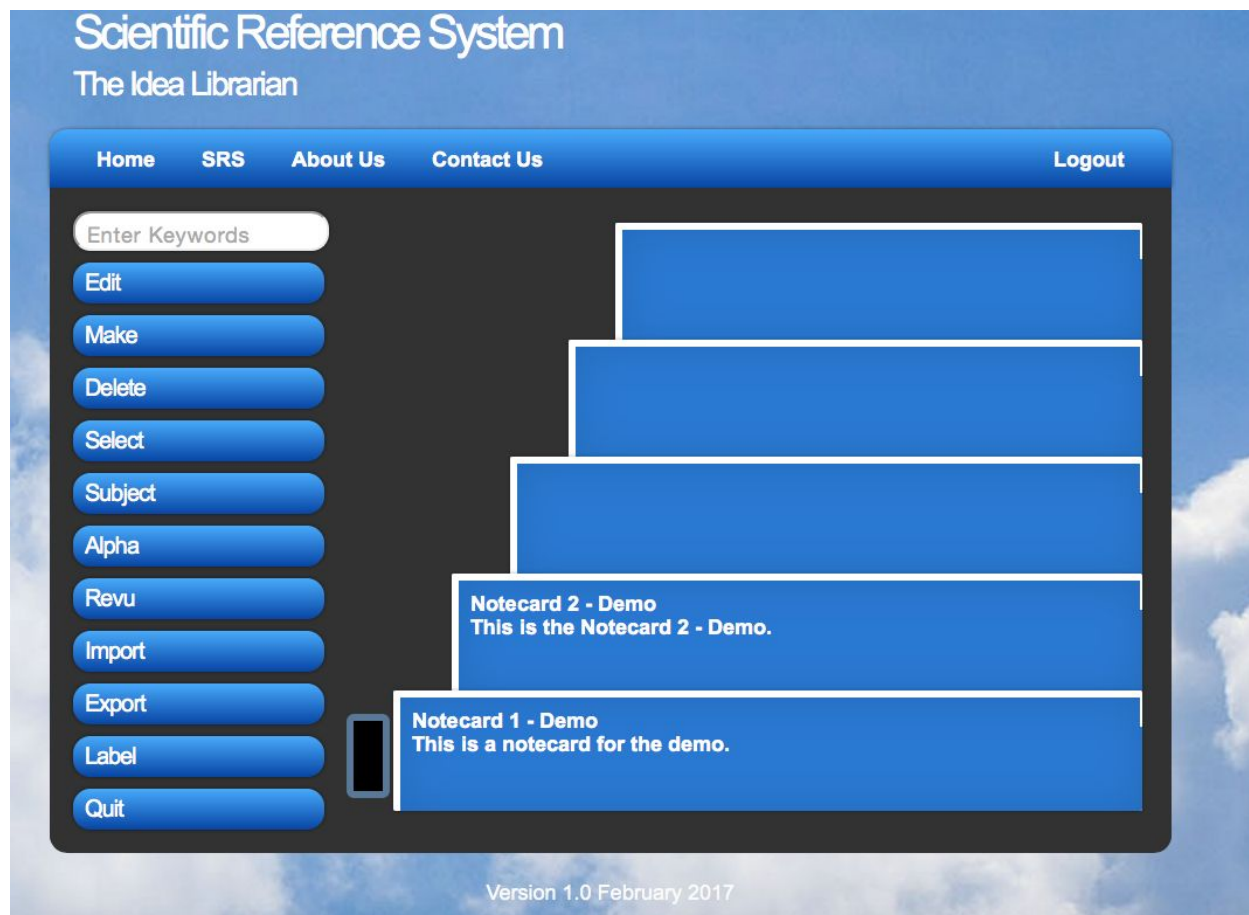


Figure 11: “List of notecards contained within a notefile.”

Notecards can be created by being imported (see section below) or by clicking the “Make” button on the notecard list page seen in Figure 11. By filling out the form shown in Figure 12 with the pertinent details, notecards can be initialized and populated. In addition, there are two extra buttons beneath the form that say “Insert Equation” and “Insert Link”. These buttons produce modals that allow you to add inline equations or links into the body of the notecard. The

equation modal, see Figure 13, will inject a MATHML structured statement into the body that will be rendered as the actual inline equation on the notecard page. The link modal, see Figure 14, will inject a link in the form of an a tag that will be rendered as a link with the anchor text matching the display text entered and point to the given URL.

The screenshot shows a web application titled "Scientific Reference System" with the subtitle "The Idea Librarian". The page path is "/NewNote/". A navigation bar includes links for Home, SRS (which is active), About, Contact Us, and Logout. The main content area is titled "Create Notecard" and contains four input fields: "Name:", "Keywords:", "Label:", and "Body:". Each field has a small diagonal icon in the bottom right corner. At the bottom of the form are four buttons: "Save" (green), "Cancel" (red), "Insert Equation" (blue), and "Insert Link" (yellow).

Figure 12: "Create Notecard."

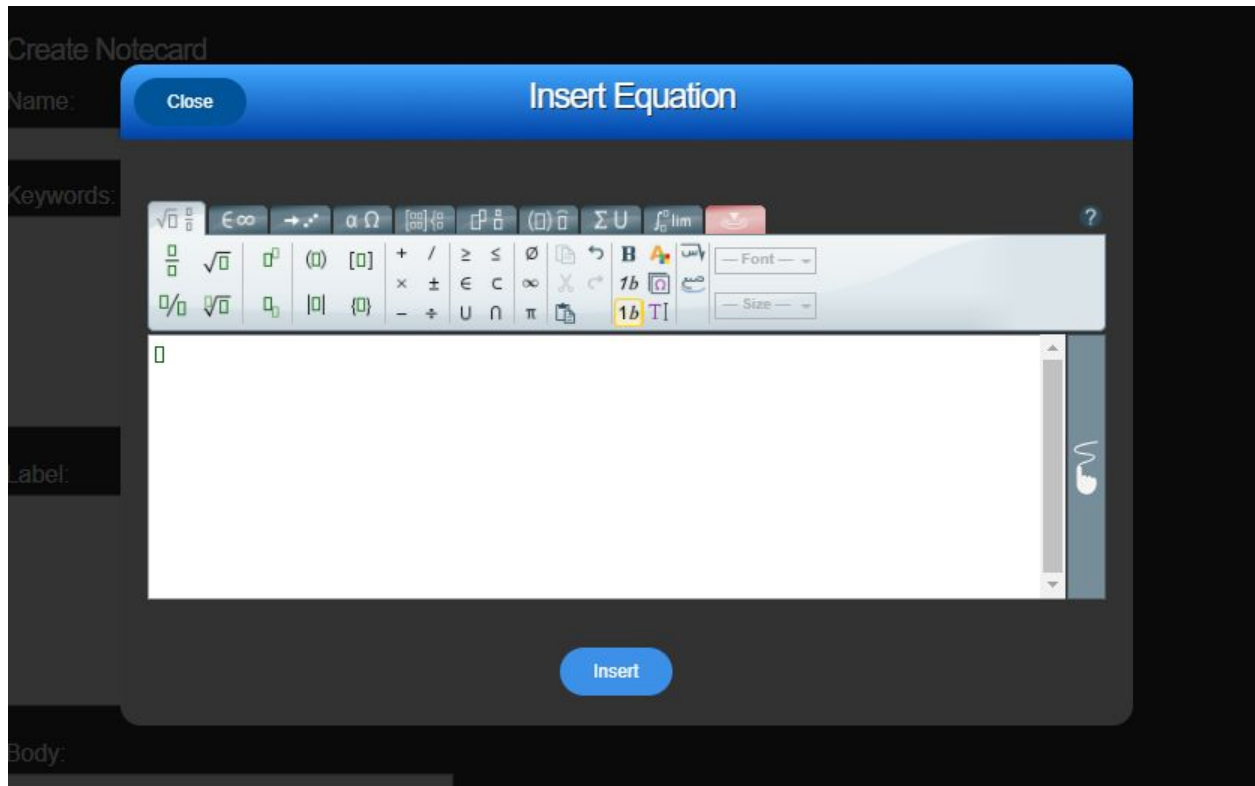


Figure 13: "Insert Inline Equation Modal."



Figure 14: "Insert Link Modal."

## Import Notecards

The SRS also has the ability to import notecards, as shown in Figure 15. So far, SRS imports all notecards contain within a binary file with SQI format. In the future, SRS should allow to import notecards from files on other formats, such as SQN format. Moreover, there is no option to import a particular notecard within the list of notecards. This might be implemented in future development as well.

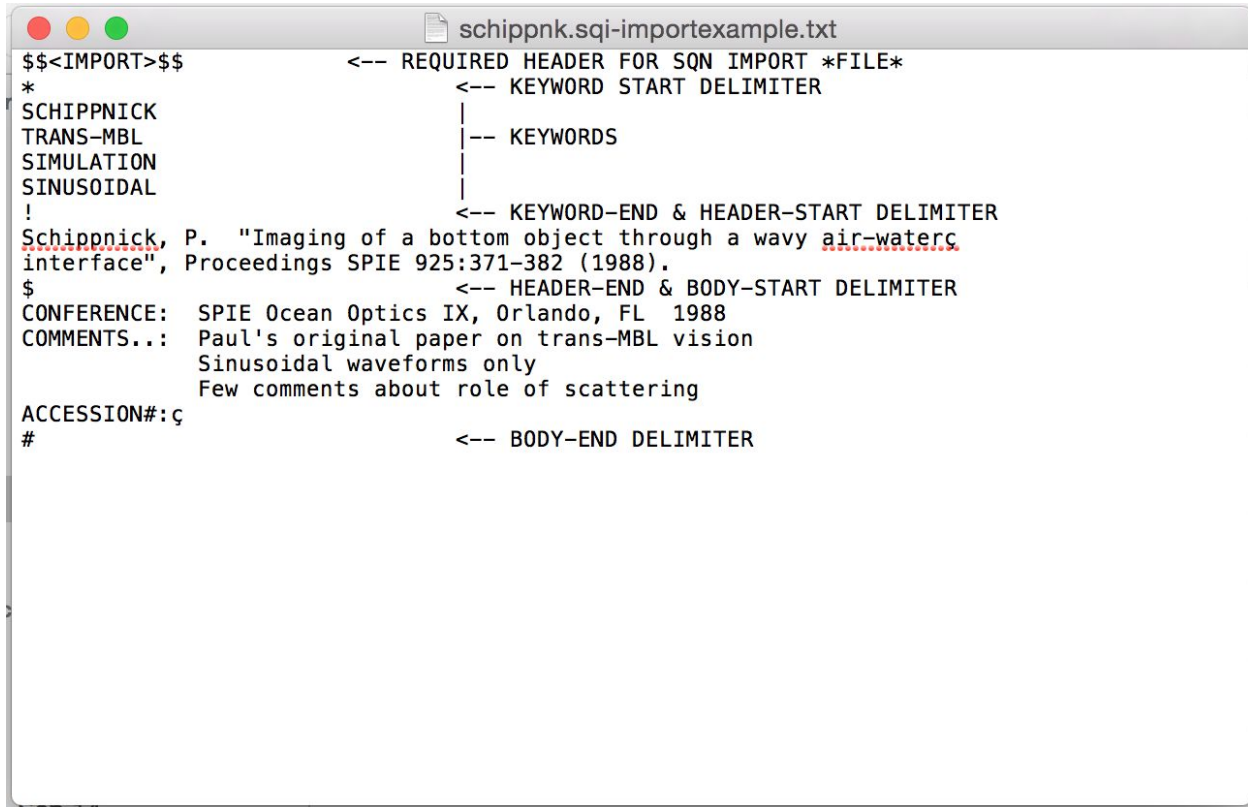
Figure 15: “Import notecards from a binary file.”

In order to import notecards from a binary file, users only need to enter a valid path, assuming that the path points to a correct binary file. If the binary file’s format is correct, then SRS will create notecards containing all the information within the binary file. Only when the path is valid and the binary file’s format is correct, then SRS will import the notecards. Otherwise, it will show an error message.

As shown in Figure 16, the correct format of a binary file includes the *header* for the file, the *keyword-start delimiter*, the keywords (which might be empty if there are no keywords associated to the notecard), the *keyword-end and header-start delimiter*, the name or title of the notecard, the *header-end and body-start delimiter*, the body of the notecard, and the *body-end*



*delimiter*. Except for the first field, the *header* for the file, all other fields repeat themselves for each notecard. Of course, the information regarding keywords, the name or title of the notecard, and the body of the notecard will likely change for each notecard.



```

$<IMPORT>$<
*
SCHIPPNICK
TRANS-MBL
SIMULATION
SINUSOIDAL
!
Schipnick, P. "Imaging of a bottom object through a wavy air-waterc
interface", Proceedings SPIE 925:371-382 (1988).
$
CONFERENCE: SPIE Ocean Optics IX, Orlando, FL 1988
COMMENTS... Paul's original paper on trans-MBL vision
Sinusoidal waveforms only
Few comments about role of scattering
ACCESSION#:ç
#

```

Figure 16: “Format of a binary file.”

## Export Notecards

In addition to import notecards to a notefile, SRS can also export notecards to a binary file with SQI format. In the future, SRS should allow to export notecards not only to binary files but to SQN files. Similarly to the import feature, users need to enter a valid path for the binary file to be created (see Figure 17).

The correct format of the binary file is depicted in Figure 16. At this moment, SRS exports all notecards contained within a notefile to a binary file; it should also have the option to export notecards one by one, as SQUARENOTE does.

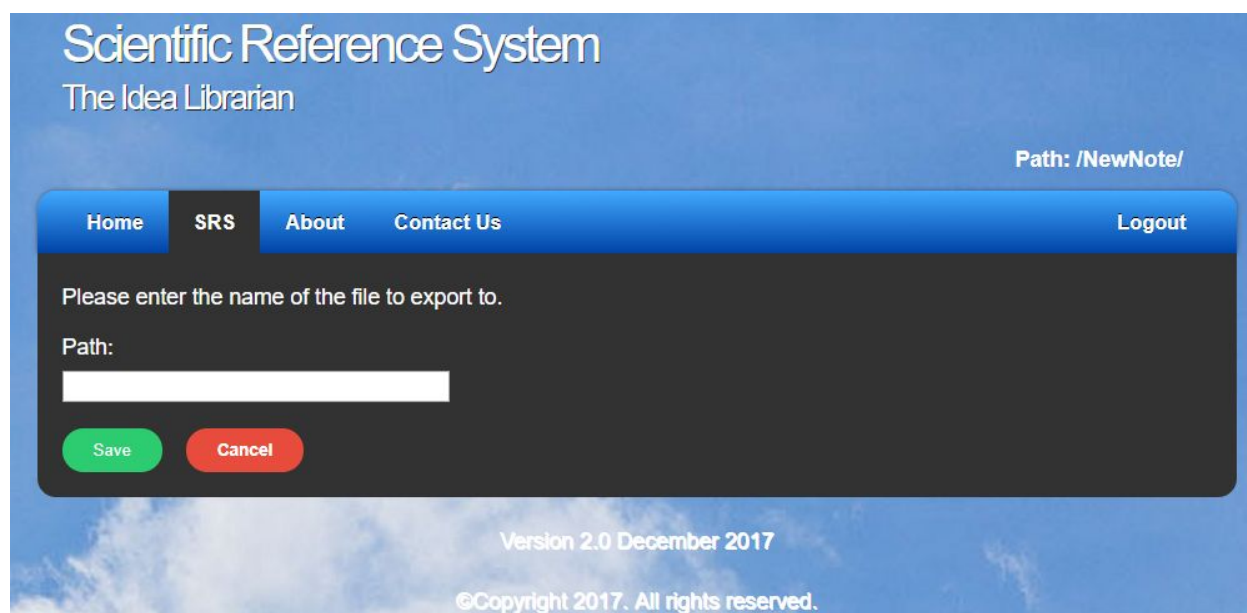


Figure 17: “Export notecards to a binary file.”

## Search

Another feature of SRS is the ability to search keyword within notecards contained in a notefile. As seen on Figure 18, users can enter the keyword or keywords they are looking for, and SRS will update the list of notecards displayed. The new list of notecards will contain only the notecards that have at least one of the keywords being searched for. In the future, SRS should also have the option to search keywords among notefiles, not only notecards.

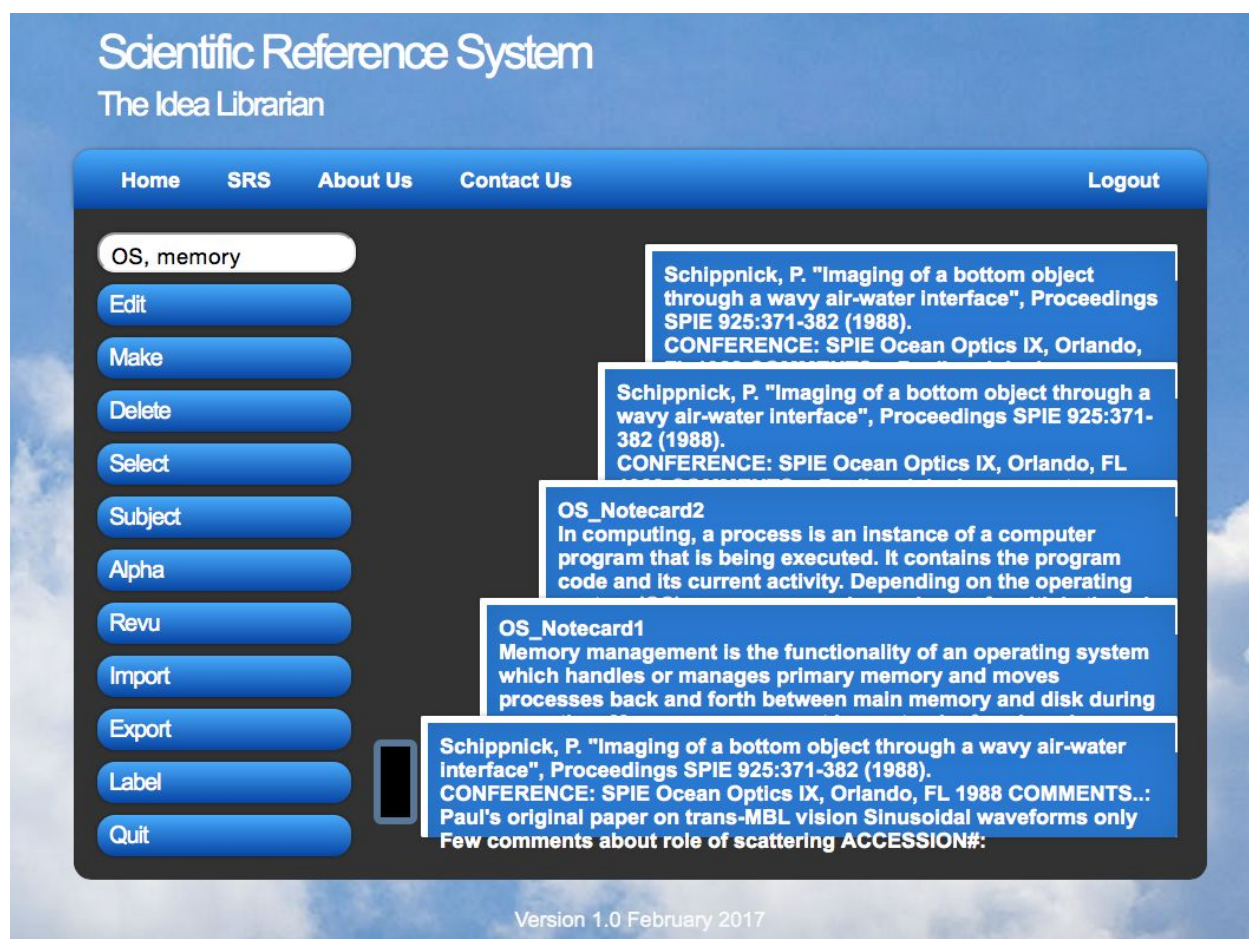


Figure 18: “Search keywords within notecards and notefiles.”

## Add / Display Equations

Within a notecard, users will be able to add equations using the equation editor shown in Figure 19. The equation editor allows to include a myriad of symbols, such as fractions, variables, matrices, parentheses, limits and series symbols - just to mention a few. Once users have finished creating their equations, they can choose to save or discard them by clicking the button ‘Save’ or ‘Cancel’ respectively. Afterwards, users will be redirected to the notecard detail page where they can see the newly created equation and also they can keep editing the notecard.

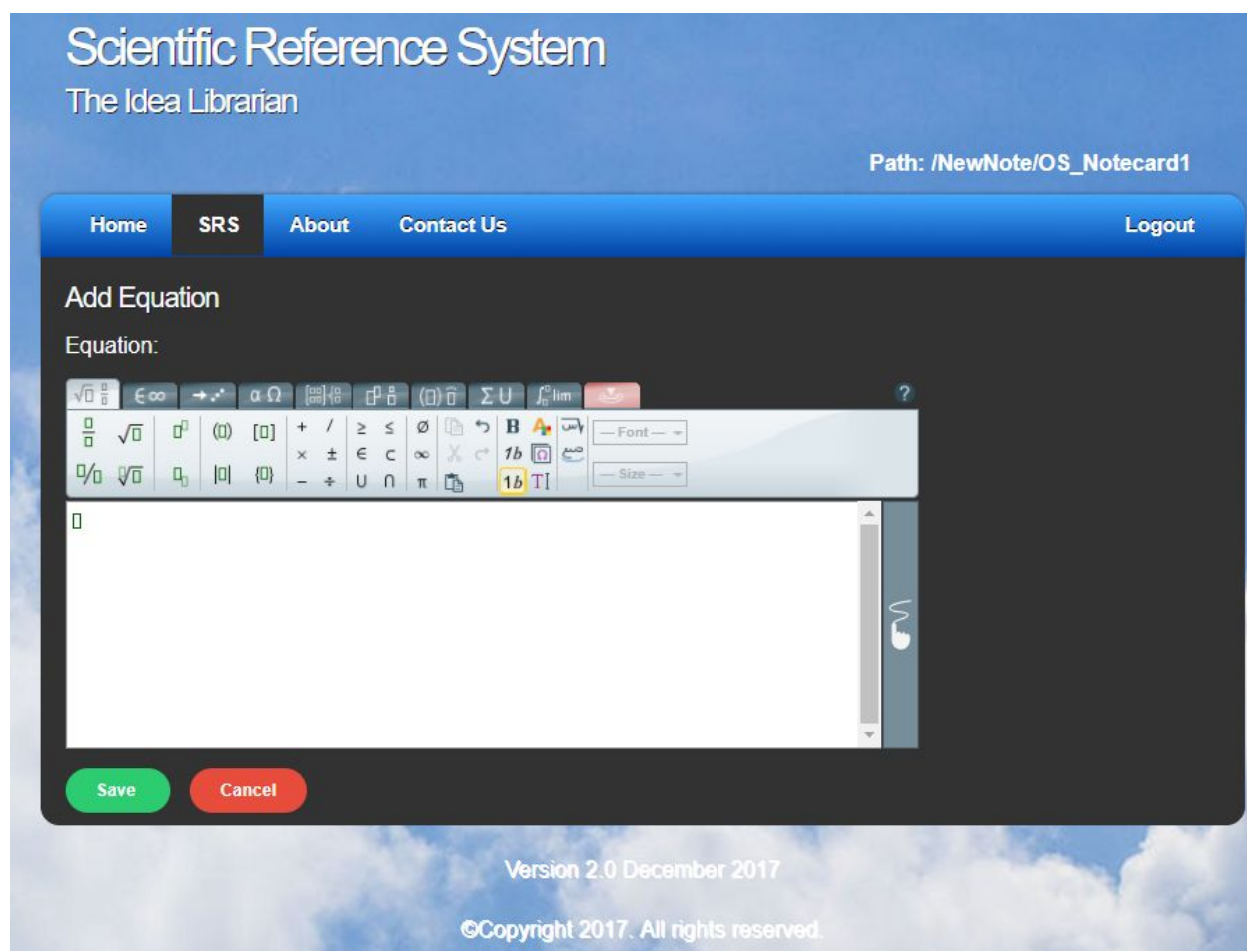


Figure 19: “Add Display Equation.”

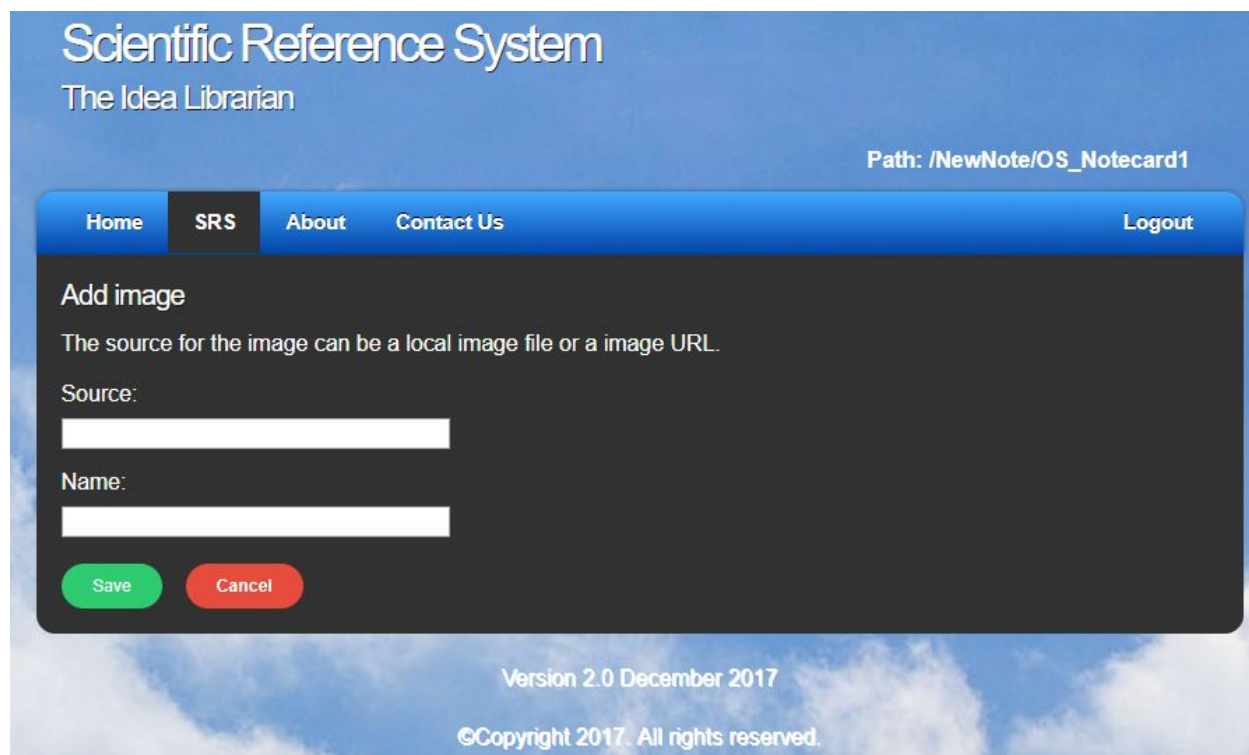
## Add Images

Besides adding equations to a notecard, SRS allows adding images to a notecard (see Figure 20). In order to add an image, users need to provide a *title* and a *source* path - source path can be a local or an online path. Before saving the image, SRS validates that the source path provided actually contains a supported image extension.

At this moment, supported image extensions are ‘.ani’, ‘.bmp’, ‘.cal’, ‘.fax’, ‘.gif’, ‘.img’, ‘.jbg’, ‘.jpe’, ‘.jpeg’, ‘.jpg’, ‘.mac’, ‘.pbm’, ‘.pcd’, ‘.pcx’, ‘.pct’, ‘.pgm’, ‘.png’, ‘.ppm’, ‘.psd’, ‘.ras’, ‘.tga’, ‘.tiff’ and ‘.wmf’.

Once users have finished entering the name and path fields, they can choose to save or discard their changes by clicking the button ‘Save’ or ‘Cancel’ respectively. Afterwards, users will be

redirected to the notecard detail page so they can keep editing the notecard. The notecard detail page will show the changes recently made.



The screenshot shows the 'Scientific Reference System' interface with the subtitle 'The Idea Librarian'. The top navigation bar includes 'Home', 'SRS' (selected), 'About', 'Contact Us', and 'Logout'. The current path is '/NewNote/OS\_Notecard1'. The main content area is titled 'Add image' and contains the instruction: 'The source for the image can be a local image file or a image URL.' Below this, there are two input fields: 'Source:' and 'Name:'. At the bottom of the form are two buttons: 'Save' (green) and 'Cancel' (red). The footer of the page indicates 'Version 2.0 December 2017' and '©Copyright 2017. All rights reserved.'

Figure 20: “Add Image.”

## Add Videos

In addition to equations and images, SRS allows to add videos to a notecard (see Figure 21). Videos can be local (videos stored in your device), online and Youtube videos. Similarly to images, users need to provide a *title* and a *source* path - source path where the video resides. Before saving the video, SRS validates that the source path provided actually contains a supported video extension.

Currently supported video extensions are '.264', '.3g2', '.3gp', '.3gp2', '.3gpp', '.3gpp2', '.3mm', '.3p2', '.60d', '.787', '.89', '.aaf', '.aec', '.aep', '.aepx', '.aet', '.aetx', '.ajp', '.ale', '.am', '.amc', '.amv', '.amx', '.anim', '.aqt', '.arcut', '.arf', '.asf', '.asx', '.avb', '.avc', '.avd', '.avi', '.avp', '.avs', '.avs', '.avv', '.axm', '.bdm', '.bdmv', '.bdt2', '.bdt3', '.bik', '.bin', '.bix', '.bmk', '.bnp', '.box', '.bs4', '.bsf', '.bvr', '.byu', '.camproj', '.camrec', '.camv', '.ced', '.cel', '.cine', '.cip', '.clpi', '.cmmp', '.cmmtpl', '.cmproj', '.cmrec', '.cpi', '.cst', '.cvc', '.cx3', '.d2v', '.d3v', '.dat', '.dav', '.dce', '.dck', '.dcr', '.dcr', '.ddat', '.dif',

'dir', '.divx', '.dlx', '.dmb', '.dmsd', '.dmsd3d', '.dmsm', '.dmsm3d', '.dmss', '.dmx', '.dnc', '.dpa', '.dpg', '.dream', '.dsy', '.dv', '.dv-avi', '.dv4', '.dvdmedia', '.dvr', '.dvr-ms', '.dvx', '.dxr', '.dzm', '.dzip', '.dzt', '.edl', '.evo', '.eye', '.ezt', '.f4p', '.f4v', '.fbr', '.fbr', '.fbz', '.fcp', '.fcproject', '.ffd', '.flc', '.flh', '.fli', '.flv', '.flx', '.gfp', '.gl', '.gom', '.grasp', '.gts', '.gvi', '.gvp', '.h264', '.hdmov', '.hkm', '.ifo', '.imovieproj', '.imovieproject', '.ircp', '.irf', '.ism', '.ismc', '.ismv', '.iva', '.ivf', '.ivr', '.ivs', '.izz', '.izzy', '.jss', '.jts', '.jtv', '.k3g', '.kmv', '.ktn', '.lrec', '.lsf', '.lsx', '.m15', '.m1pg', '.m1v', '.m21', '.m21', '.m2a', '.m2p', '.m2t', '.m2ts', '.m2v', '.m4e', '.m4u', '.m4v', '.m75', '.mani', '.meta', '.mgv', '.mj2', '.mjp', '.mjpg', '.mk3d', '.mkv', '.mmv', '.mnv', '.mob', '.mod', '.modd', '.moff', '.moi', '.moov', '.mov', '.movie', '.mp21', '.mp21', '.mp2v', '.mp4', '.mp4v', '.mpe', '.mpeg', '.mpeg1', '.mpeg4', '.mpf', '.mpg', '.mpg2', '.mpgindex', '.mpl', '.mpl', '.mpls', '.mpsub', '.mpv', '.mpv2', '.mqv', '.msdvd', '.mse', '.msh', '.mswmm', '.mts', '.mtv', '.mvb', '.mvc', '.mvd', '.mve', '.mvex', '.mvp', '.mvp', '.mvy', '.mx', '.mxv', '.mys', '.ncor', '.nsv', '.nut', '.nuv', '.nvc', '.ogm', '.ogv', '.ogx', '.osp', '.otrkey', '.pac', '.par', '.pds', '.pgi', '.photoshow', '.piv', '.pjs', '.playlist', '.plproj', '.pmf', '.pmv', '.pns', '.ppj', '.prel', '.pro', '.prproj', '.prtl', '.psb', '.psh', '.pssd', '.pva', '.pvr', '.pxv', '.qt', '.qtch', '.qtindex', '.qtl', '.qtm', '.qtz', '.r3d', '.red', '.rcproject', '.rdb', '.rec', '.rm', '.rmd', '.rmd', '.rmp', '.rms', '.rmv', '.rmvb', '.roq', '.rp', '.rsx', '.rts', '.rts', '.rum', '.rv', '.rvid', '.rvl', '.sbk', '.sbt', '.scc', '.scm', '.scm', '.scn', '.screenflow', '.sec', '.sedprj', '.seq', '.sfd', '.sfvidcap', '.siv', '.smi', '.smi', '.smil', '.smk', '.sml', '.smv', '.spl', '.sqz', '.srt', '.ssf', '.ssm', '.stl', '.str', '.stx', '.svi', '.swf', '.swi', '.swt', '.tda3mt', '.tdx', '.thp', '.tivo', '.tix', '.tod', '.tp', '.tp0', '.tpd', '.tpr', '.trp', '.ts', '.tsp', '.ttxt', '.tvs', '.usf', '.usm', '.vc1', '.vcpf', '.vcr', '.vcv', '.vdo', '.vdr', '.vdx', '.veg', '.vem', '.vep', '.vf', '.vft', '.vfw', '.vfz', '.vgz', '.vid', '.video', '.viewlet', '.viv', '.vivo', '.vlab', '.vob', '.vp3', '.vp6', '.vp7', '.vpj', '.vro', '.vs4', '.vse', '.vsp', '.w32', '.wcp', '.webm', '.wlmv', '.wm', '.wmd', '.wmmp', '.wmv', '.wmv', '.wot', '.wp3', '.wpl', '.wtv', '.wve', '.wvx', '.xej', '.xel', '.xesc', '.xfl', '.xlmv', '.xmv', '.xvid', '.y4m', '.yog', '.yuv', '.zeg', '.zm1', '.zm2', '.zm3', and '.zmv'.

Once users have finished entering the name and path fields of the video, they can choose to save or discard their changes by clicking the button ‘Save’ or ‘Cancel’ respectively. Afterwards, users will be redirected to the notecard detail page so they can keep editing the notecard. The notecard detail page will show the changes recently made.



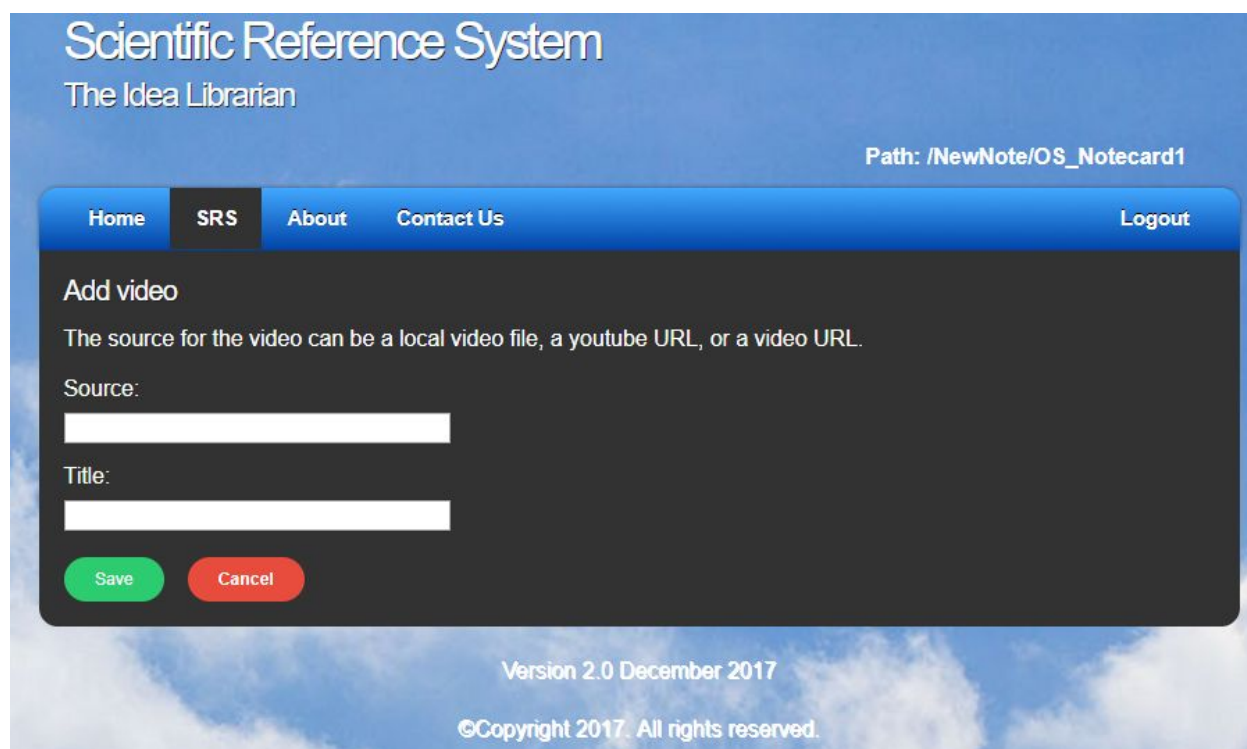


Figure 21: “Add Video.”

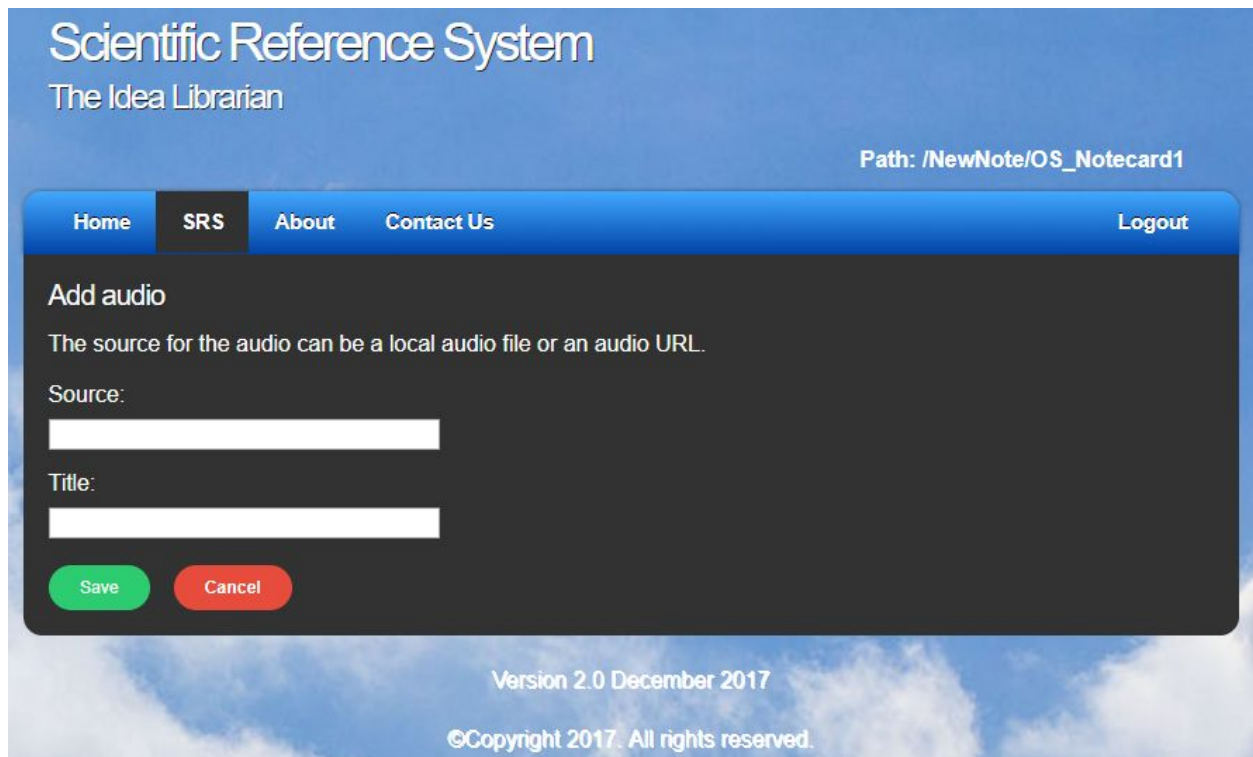
## Add Audio

Audio files can also be associated to a notecard. One of SRS’ features allows to save audio files, as shown in Figure 22. These audio files can be stored locally in your computer or reside in the internet. Adding an audio file behaves likewise images and videos: users need to provide a *title* and a *source* path - source path where the audio file is hosted. Before saving it, SRS validates that the source path provided actually contains a supported audio file extension.

SRS currently supports the following audio extensions: '.3gp', '.aa', '.aac', '.aax', '.act', '.aiff', '.amr', '.ape', '.au', '.awb', '.dct', '.dss', '.dvf', '.flac', '.gsm', '.iklax', '.ivs', '.m4a', '.m4b', '.m4p', '.mmf', '.mp3', '.mpc', '.msv', '.ogg', '.oga', '.mogg', '.opus', '.ra', '.rm', '.raw', '.sln', '.tta', '.vox', '.wav', '.wma', '.wv', '.webm', and '.8svx'.

Same as with image and video files, users can decide whether to associate or not an audio file to a notecard once they have finished entering the name and source fields of the audio. They can do this by clicking the button ‘Save’ or ‘Cancel’. Then users will be redirected to the notecard detail page so they can keep editing the notecard. The notecard detail page will show the newly

associated audio file, if one was created.



The screenshot shows the 'Scientific Reference System' interface with the subtitle 'The Idea Librarian'. The top navigation bar includes 'Home', 'SRS' (highlighted), 'About', 'Contact Us', and 'Logout'. The current path is '/NewNote/OS\_Notecard1'. The main content area is titled 'Add audio' and contains the instruction: 'The source for the audio can be a local audio file or an audio URL.' Below this are two input fields: 'Source:' and 'Title:'. At the bottom of the form are two buttons: 'Save' (green) and 'Cancel' (red). The footer indicates 'Version 2.0 December 2017' and '©Copyright 2017. All rights reserved.'

Figure 22: “Add Audio.”

## Add Documents

SRS permits to associate equations, images, videos and audio files to a notecard. It also allows to associate documents. As seen in Figure 23, SRS only requires a *source* field to allow adding a document. The path to a document stored locally or the link to an online document can be entered at this *source* field. SRS will check if the provided *source* path is one of its currently-supported document extensions.

These document extensions include ‘.pdf’, ‘.doc’, ‘.docx’, ‘.xls’, ‘.xlsx’, ‘.ppt’, ‘.pptx’, ‘.pps’, and ‘.ppsx’.

Once users determine whether to associate or not a document to a notecard (they can do this by clicking the button ‘Save’ or ‘Cancel’) they will be taken back to the notecard detail page. The notecard detail page will show the changes.

The screenshot shows a web application titled "Scientific Reference System" with the subtitle "The Idea Librarian". The background is a blue sky with clouds. In the top right corner, the path "/NewNote/OS\_Notecard1" is displayed. A navigation bar contains links for "Home", "SRS" (which is highlighted), "About", "Contact Us", and "Logout". The main content area is a dark grey box titled "Add Document". It contains the instruction "The source for the document can be a local file or an online file URL." and a label "Source:" followed by a white text input field. Below the input field are two buttons: a green "Save" button and a red "Cancel" button. At the bottom of the page, the text "Version 2.0 December 2017" and "©Copyright 2017. All rights reserved." are visible.

Figure 23: “Add Document.”

## Structure of Django Files

Once all the installation and pull-from-GitHub steps are done, the directories of the SRS project should have a structure similar to the one shown in Figure 24. We called our main directory *djangoirls*, just as the Django Girls Tutorial prompts us to; you could have chosen any other name. If you followed the aforementioned steps correctly, you should see the nested directories: *mysite*, *myvenv* and *srs*. There are also important files, such as *manage.py*.

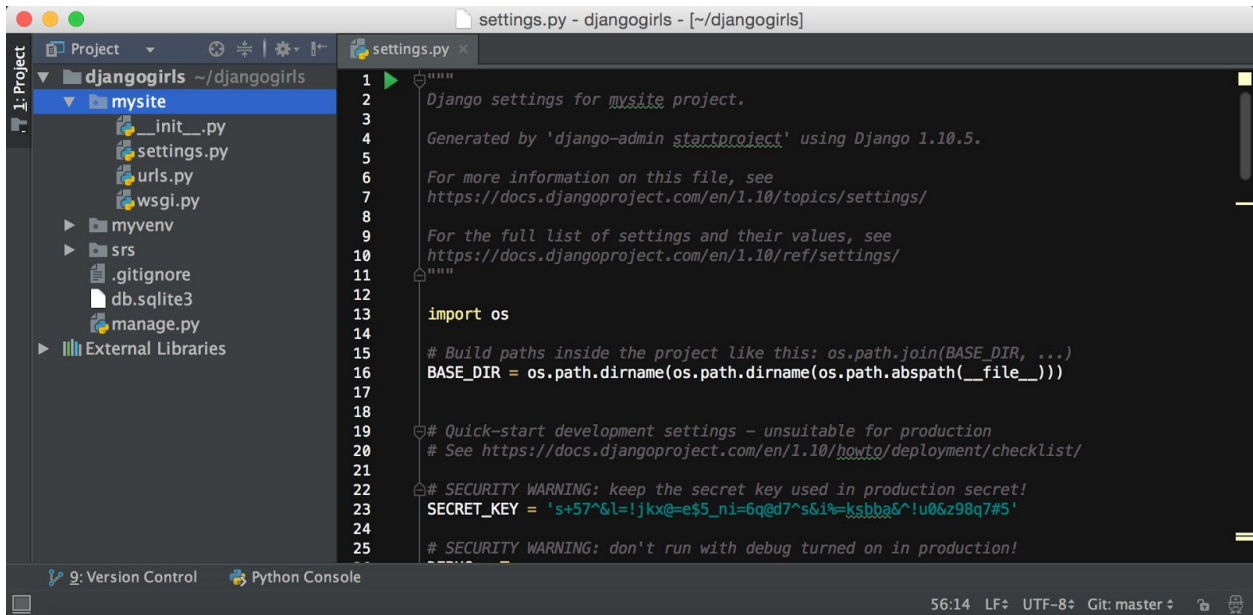


Figure 24: “Directory structure of the SRS project.”

The directory *mysite* contains configuration files for the SRS website. File *settings.py* is of particular relevance, since most of the settings are done within this file.

The directory *myenv* contains packages and libraries that were added after the installation and configuration steps. It contains the information for the virtual environment to work along the Django framework. This directory does not need to be modified by developers.

The directory *srs* is where the majority of the source code lives. It contains several nested subdirectories (*migrations*, *static*, *templates*, and *templatetags*) and important files, as shown in Figure 25.

- The files within *migrations* are just files that are generated by Django before applying the changes made to the models into the local SQLite database; these files do not need to be changed by the developers.
- The *static* subdirectory, however, does contain css and javascript files, icons and images that will probably be modified in future development efforts. This is where the developers want to add or modify images, the website’s style or behaviors using javascript code.
- The subdirectory *templates* has a nested directory called *srs* that contains all html files

used in the SRS website. If developers want to add or change any html file, this is the place to do so.

- The subdirectory *templatetags* contains a couple of python files. One of these files contains the filters applied in SRS.

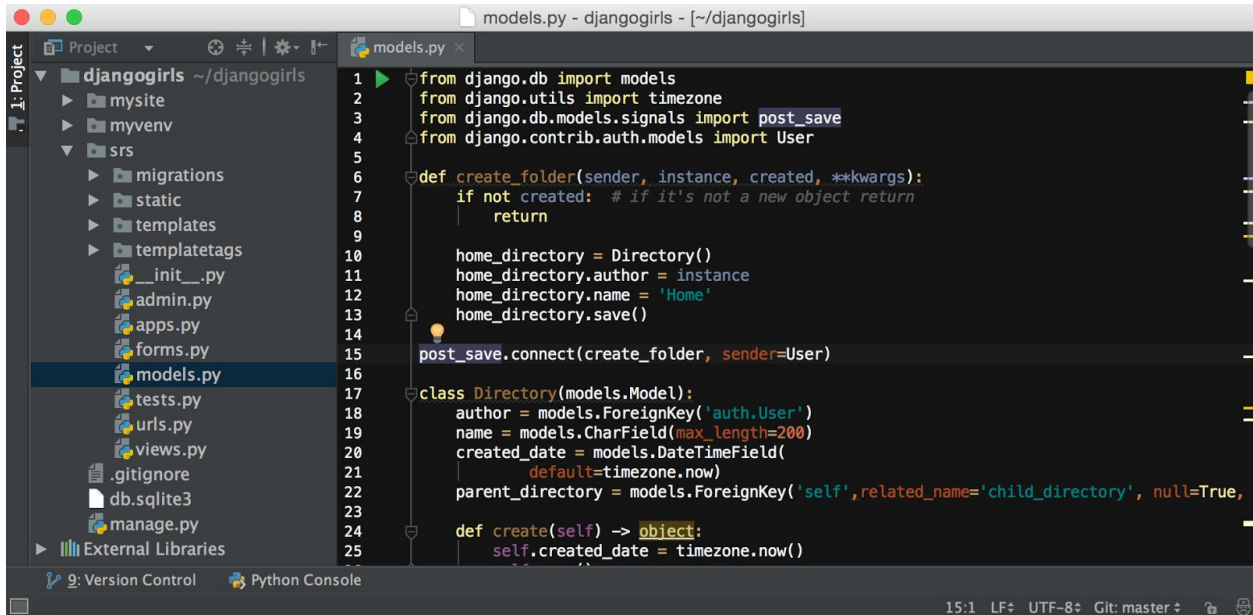


Figure 25: “Directory structure of the *srs* directory.”

As mentioned earlier, directory *srs* nested within directory *djangogirls* contains important files (see Figure 25).

- The Django framework facilitates the communication between python code and the local SQLite database. In other words, if developers need to create a new table in the database, they only need to create a model of the database object in *models.py* and apply the changes later on using certain commands in the command prompt ( `python manage.py makemigrations` and then, `python manage.py migrate` ).
- Whenever a model is created in the SRS website, the model needs to be register in *admin.py*.
- SRS uses forms, a feature implemented in the Django framework that eases developers’ work. Every time a developer creates a new form, he or she should create it in *forms.py*.
- File *urls.py* contains all urls that the SRS web application uses to render or redirect the

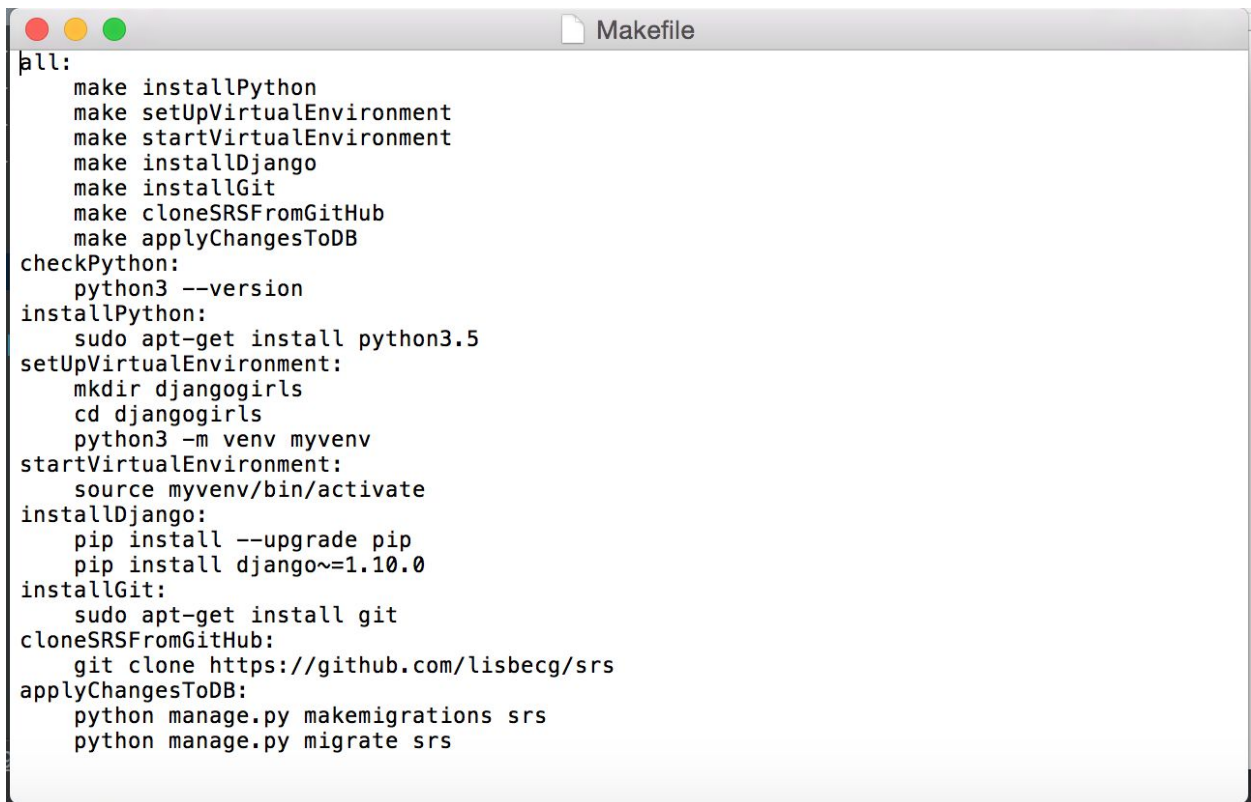
html pages.

- File *views.py* contains all the existing views in the source code. A view will define the behavior of a page.

The directories and files aforementioned are the more relevant ones to any development purposes. Nevertheless, developers are welcome and highly encouraged to view all files and directories to have a better understanding of the SRS web application's behavior behind scenes.

## Creating an SRS Executable

The best way to produce an executable for the most current version of SRS is to follow the steps given the Django Girls Tutorial because it will show you the steps you need to follow depending on which operating system you are using [1]. Nevertheless, we produced a makefile to apply on Linux operating system that should install all programs and files required to continue the development of SRS, as you can see in Figure 26.



```
all:
    make installPython
    make setUpVirtualEnvironment
    make startVirtualEnvironment
    make installDjango
    make installGit
    make cloneSRSFromGitHub
    make applyChangesToDB
checkPython:
    python3 --version
installPython:
    sudo apt-get install python3.5
setUpVirtualEnvironment:
    mkdir.djangogirls
    cd.djangogirls
    python3 -m venv myenv
startVirtualEnvironment:
    source myenv/bin/activate
installDjango:
    pip install --upgrade pip
    pip install django~=1.10.0
installGit:
    sudo apt-get install git
cloneSRSFromGitHub:
    git clone https://github.com/lisbecg/srs
applyChangesToDB:
    python manage.py makemigrations srs
    python manage.py migrate srs
```

Figure 26: “Makefile with installation steps required in Linux operating system.”



In order to execute this code, you only need to copy this makefile in the directory in which you want to create your Django project. Then, open the command prompt or like program in your computer and position yourself in the earlier-mentioned directory. Then, run the following line.

```
make
```

If your installation goes smoothly, you are ready to start development. Otherwise, please refer to the Django Girls Tutorial for any possible command instruction variations that you might need.

## Installation Program Explained

In order to set up the environment for SRS, you need to follow certain main steps: install Python, setup the virtual environment for Django, and install Django, a code editor and Git. In addition, you need to create a GitHub account (if you do not have one already) and be able to log into the existing PythonAnywhere account for SRS. Finally, you need to clone the existing source code into your machine and apply the changes made to the models to your local SQLite database before start collaborating in this project.

The instructions listed within the following steps target the Linux operating system. If you have a different operating system, that is OK - just read the Django Girls Tutorial and find your instructions' format [1].

### Install Python

Django is written in Python; therefore, we need to have Python installed. The Django Girls Tutorial suggests to install Python 3.5 or later [1]. First, check whether you have an installation of Python 3.5.

```
python3 --version
```

If you have a version of Python 3.5 or later installed, go to the next step. Otherwise, you need to install it.

```
sudo apt-get install python3.5
```

Just to be sure, check which version of Python you just installed.

```
python3 --version
```

## Set up Virtual Environment

Once Python 3.5 or later has been installed, you can proceed to install the virtual environment. This step is not necessary, but highly recommended. The virtual environment will isolate your Python/Django setup on a per-project basis; as a result, any changes you make to one website will not affect any other websites you are also developing [1]. In order to create the virtual environment, you need to create or select a directory in which SRS project is going to live. In our case, we chose a directory called *djangogirls*; you can name it as you like.

```
mkdir djangogirls
cd djangogirls
```

Then, we create a virtual environment called *myenv*. You can again choose any name you like for the virtual environment, but it is highly recommended that you use lowercase and no spaces in your name [1].

```
python3 -m venv myenv
```

If you encounter any errors, please refer to the Django Girls Tutorial [1]. Otherwise, a folder called *myenv* has been created. The next step is to start the virtual environment.

```
source myenv/bin/activate
```

After running this line of code, the prefix in your console's prompt will show (*myenv*).

## Install Django

Once the virtual environment is running, the next logical step is to install Django. In order to install Django, though, we need to check for the latest upgrade of pip, the program being used to install Django.

```
pip install --upgrade pip
```

Afterwards, just install Django using the latest version of pip.

```
pip install django~=1.10.0
```

Now you are ready to create a Django application.

## Install a Code Editor

It is highly recommended to use a code editor for your development efforts, such as PyCharm, Gedit, Sublime Text and Atom. One of the reason to use a code editor is that code needs to be plain text, and programs like Word or Textedit produce rich text ( with fonts and formatting ) which will produce problems at compilation. In addition, most code editors will highlight code or detect problems before you do. In our case, we preferred to use PyCharm, but you are welcome to use the one you feel the most comfortable with.

## Install Git

Git is a type of version control system used by many developers. It tracks the changes made to files over time, and it is very useful for development efforts. If you have not done it yet, install Git.

```
sudo apt-get install git
```

## Create a GitHub Account

If you do not have a GitHub account yet, we recommend you to create one as soon as possible by going to [GitHub.com](https://github.com). GitHub allows you to collaborate with other developers in real time. It is a extremely useful platform for coders worldwide.

The latest code of SRS is located at <https://github.com/lisbecg/srs>.

## PythonAnywhere Account

SRS website is live in <http://srs.pythonanywhere.com>. In order to log in to SRS, you can use the following credentials: username: admin, password: admin123. The email associated with this account is [onlinesrsproject@gmail.com](mailto:onlinesrsproject@gmail.com).

## Clone SRS Project from GitHub

Once you have installed git in your virtual environment, the next step is to clone the SRS project to your folder *djangogirls*. Make sure that you are located within the folder *djangogirls*.

git clone <https://github.com/lisbecg/srs.git>

At this point, you should be able to have the latest source code in your computer.

## Create Tables for Models in Your Database

Once you have the latest version of the SRS project's code in your computer, you need to update the SQLite database that Django framework has by default. In order to do this, you need to run two commands. Before running these commands, make sure that you started the virtual environment *myvenv* and that you are positioned within the *djangogirls* directory.

```
python manage.py makemigrations srs
python manage.py migrate srs
```

The first command lets Django know that some changes have been made to your models (objects that represent the tables in your SQLite database) and created a migration file. The second command applies this migration file to the database.

At this point, you should be ready to start collaborating towards the completion of the SRS project.

## Bugs and Problems Encountered

Even though SRS web application has several features properly implemented, there are still problems that need to be fixed. These problems are listed next.

### Valid Import/Export Path is Invalid in Live Website

In addition to the directory icon not being display, SRS has another problem: import and export features do not work properly in the live website ( <http://srs.pythonanywhere.com> ). As shown in Figure 27, even though the path is valid, the website does not recognize the given path as a valid address. This feature works properly in the localhost, though. If the given path is valid, then the behavior of import and export features are correct in the localhost. Therefore, it should be some setting that needs to be updated in the live website. A solution to this problem needs to be found.

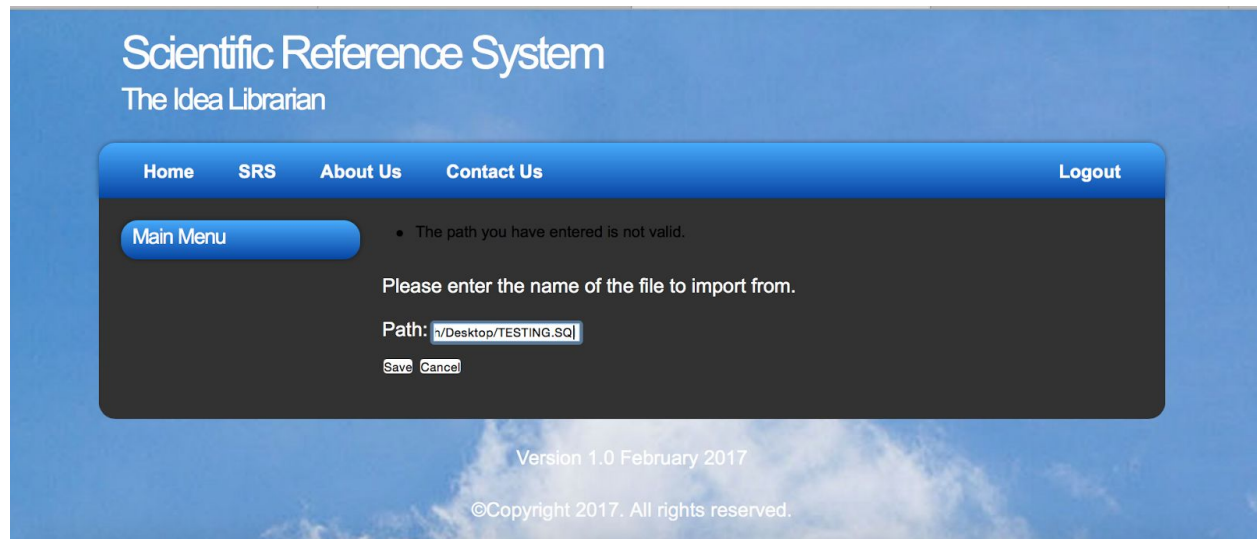


Figure 27: "Valid path is invalid in live website."

## References

- [1] "Django Girls Tutorial," tutorial.djangogirls.org [Online]. Available: <https://tutorial.djangogirls.org/en/>. [Accessed: Apr. 25, 2017].