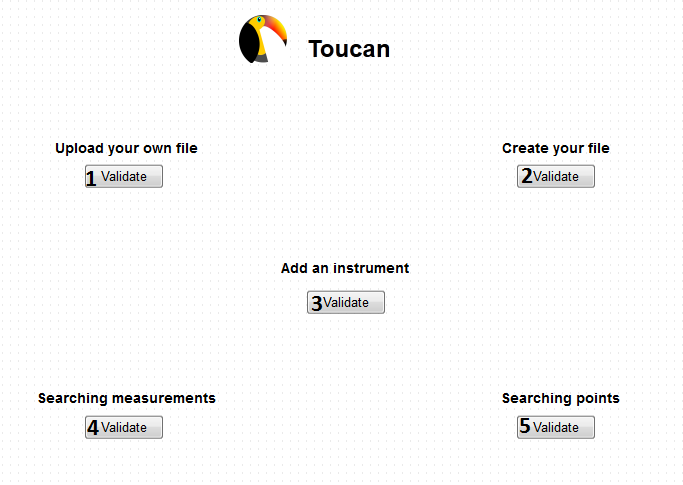
Documentation about the Toucan interface – mock up

# 1 – Menu



Each button is linked with another page, which contains a different form for each action you need to do.

1. Linked with the page which makes the user able to upload directly his metadata file and his image

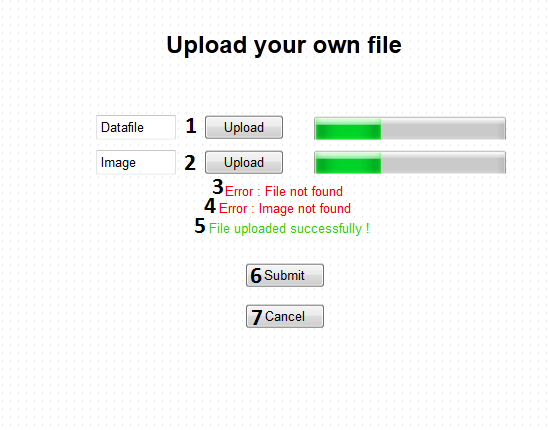
2. Linked with the page which makes the user able to enter data to build the json file by filling textfields

3. linked with the page making the user able to add an instrument by inform the name and the wavelengths of this instrument

4. Linked the page where the user is able to searching a measurement

5. Link to the page which permits to search for points using latitude and longitude

# 2 – The file uploader



This page allows the user to upload a file. json he owns and the corresponding image.

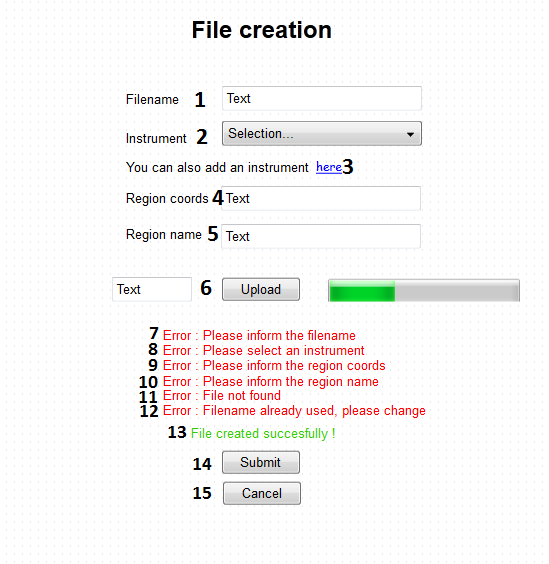
1. The user will be able there to choose a data file to upload. The upload button opens a window allowing the user to choose his file into his computer. The progression bar shows how long the process will take.
2. It’s the same principle as the one just above, excepted this one is done for an image
3. This error will appear when you try to submit the form when no data file is uploaded – or if the file is not a .json file
4. Same as 3. Excepted it is for an image there, not a data file
5. This message will appear when the upload is successfully completed after you try to submit
6. Submit button which permits to complete the form. If the form is not correctly completed, an error will appear (3 or/and 4), if it’s not the case the success message will appear.

The submit button do not redirect you on the menu as you may need to upload many files in one time

1. Cancel button – Used to return to the menu

This page needs to be protected well; because if it isn’t the user might be able to upload lots of .json file which aren’t supposed to be uploaded, like a .json file with another structure (filename, instrument, region coords, region name). If another .json file is uploaded this will make the database going wrong. That’s also why we need to verify if the files are really a .json type and an image type.

# 3 – The file creator



This page allows the user to create the file directly using this form instead creating and upload it after. The json is created and uploaded into the staging area instantly.

1. The user will put the filename into the field
2. This select will contains every instrument already known by the database, like this the user will just have to choose the one he wants to use
3. This will be linked with the Instrument creation page, so if the user need to create a instrument because the one he wants to use is not in the list, we will be able to do so, using this link
4. This field will be used to put the regions coords
5. Same as 4. but for the region name
6. This field will be for the image you need to upload, the Upload button will open a browser allowing the user to choose the file he wants to upload. The progression bar shows if the file is ready to be uploaded or not.
7. An error which will appear when the user will try to submit the form without informing the filename
8. This error will occur when the user will submit the form without any instrument.

To do this test, we need the select to be initialized on a default value. That’s why the select will be initialized on such a value.

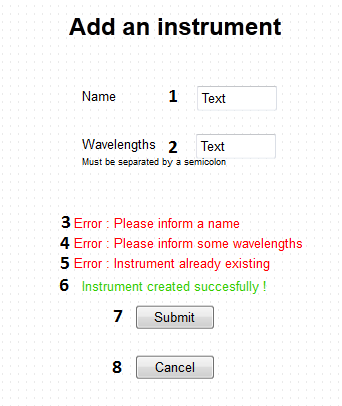
1. This error will show when the user will try to submit the form without informing any region coords.
2. Same error as 9. but for the region name
3. This error will occur when the user will submit while he won’t have upload any image file or a file which isn’t a image.
4. This error will occur when the filename chosen has been already used

!!! This error may not exist if we can use twice the sale file !!!

1. This message will appear when the submit will be successful
2. Submit button
3. Cancel button ; use it to return to menu

As for the upload page, we need to put some protections on this one, for the image file.

# 4 – Adding an instrument



This page is the one which permits to the user to create a new instrument.

1. The textfield where you put the name of your new instrument
2. The field where you put the wavelengths of the instruments, separated by semicolons
3. If you submit with no name put into the field, you got this error
4. Same as 3. But with the wavelengths
5. If the instrument is already existing into the database, you got this message to avoid to create something which is already existing
6. This message appears when the submit succeeds
7. Submit button
8. Cancel button – return to menu

There is a variant, maybe easier – it can be possible to put one per one the wavelengths, like this we won’t have any problems, and it might be easier. If this solution is chosen, then the error message called 5. will probably be used in case we found a instrument with the same name and the same wavelength. This exception is also to avoid the creation of an object already existing.