**Application User Guide**

**Requirements**

* Python 3: <https://www.python.org/downloads>
* GitHub\*: <https://desktop.github.com>

\*For certain commands, GitHub must be integrated to be used in terminal processes. This is usually automatically done for Linux and OSX platforms. Windows system needs to be configured to work with shell.

**Brief Overview**

The application sends requests to NASA archive and Exoplanet catalogue and retrieves information stored in csv files. These records are then converted into XML files, which gets processed and compared to existing XML files in a local repository. Changes are combined in a temporary folder and can be merged into the local repository after review. System files are used to store information for the application to use such as last commit dates and repository location.

**Environment Set Up**

Before running the program, there are some configurations needed in order for the application to run properly. Run main.py and do the following:

1. Use the ‘date –c <YYYY-MM-DD>’ command to set the last commit date to set the date from which extraction of records should start from. (Alternatively, this can be inputted in the last\_commit\_date.txt file)
2. matchedSystems.txt helps remember previous matches to improve performance of the system. If this file is removed, program might take a long time to go through the local repository.
3. The local repository must be inside the Applications folder. Functionality to work with a repo outside the Application folder is a future feature.

**Use**

Run main.py to get the terminal interface. To list the available commands, enter ‘help’. To get detailed usage of a command, enter ‘help [command]’. The suggested flow would be as follows:

1. Enter ‘extract’ command to extract data from NASA and Exoplanet catalogue. For automatic commits (explained later on), use the ‘-a’ flag
2. Go into Changed\_Systems folder inside the Application folder. It will contain XML file of updated systems since the last commit date. Verification can be done here. cleanUpReport.txt is going to contain a report of the cleanup.py on the changed files.
3. Enter ‘commit’ to copy the modified files into the local repository and synchronize the local repository and the remote

**Troubleshooting**

The program has been tested to prevent failures. However, there are some scenarios where it might not be able to perform some of the commands properly.

1. When extracting, there might be an error that says ‘Could not extract from NASA’. In this case, please ensure that all text and csv files from the ‘extracted’ and ‘updated’ folders are closed, as they may need to be written to.
2. When committing, errors might pop up due to conflicts. Please resolve in GitHub and try again.
3. Some arbitrary ‘File not found’ errors might arise when extracting. This is usually fine, as it might arise due to no matching systems found on a new system.