Title: “EEG WEAVER REPORTER. A open source app for neurophysiologic reports.”

Tags:

* Python
* Electroencefalogram
* Database
* Clinical neurophysiology
* Brain
* EEG

Authors:

* Name: Paulo Afonso Medeiros Kanda
* Orcid: 0000-0002-0959-0540
* Equal-contrib: true
* Affiliation: 1
* Name: Felipe Guimarães Kanda
* Orcid: 0000-0000-0000-0000
* Equal-contrib: true
* Affiliation: 2
* Name: Rafael Guimarães Kanda
* Orcid: 0000-0000-0000-0000
* Equal-contrib: true
* Affiliation: 3

Afilliations:

* Index: 1
* Neurovale Neurophysiology Laboratory, Taubate, Sao Paulo Brazil.
* Index: 2
* Departament of Medical Sciences, University of Taubate, Sao Paulo Brazil.
* Index: 3
* Universidade Estadual de Campinas, Sao Paulo, Brazil.

date: 11.08.2022

# Summary

EEG Weaver – Reporter (EWR) is a windows app intended to be used by neurophysiologists to report eletrencephalogram (EEG) findings. We wrote EWR because there are a lot of commercial EEG recording softwares but some of them don’t have satisfactory report options. This problem is mostly important in developing countries. Consequently, EWR can help researches and clinical neurophysiologists to describe their EEG analysis results. It is built in python 3.10 with use of tkinter and report lab libraries to work in windows 10-11 systems. With EWR the user can create as many data banks as needed in sqlite3. Each of those databanks can store many individual EEG reports, classified by patient or record name.

What is EEG Weaver – Reporter (EWR) and its purpose. The EEG study consists of three basic steps: recording, reviewing data, discussing or explaining results to clinical staff. The latter is “the report”, the stage where the neurophysiologist conveys a written result of EEG visual analysis, along with an interpretation of its clinical significance. Here a clinical correlation is offered, directed at the specific diagnostic implications. The report is characterized by (1) an outline of the study parameters, (2) a description of the record, (3) an interpretation or summary that includes an impression of whether the study is normal or abnormal, the degree of abnormality, and (4) the correlation of the EEG with the clinical picture1. EWR is a package that aims to provide a free, automatic and easy to use alternative of EEG report creation, although, it can be used to report almost anything. EWR was created because it is difficult to find a similar free alternative to generate EEG reports and because there are scenarios where some EEG software resources are limited or report capabilities are not well integrated to applications used. Furthermore, there are some very good options of reporters in some free apps, for example, the reporter module of MNE2, but it depends on proficiency in programming, and it generates a specific summary of results in HTML. Conversely, EWR can be used by in any situation where an objective and simple report is needed because it is similar to a clean canvas to receive text. The most impressive advantage of EWR is its capability of sqlite3 databanks creation which allows the user to create as many reports databanks as needed. Reports can be duplicate, edited, deleted, stored and searched with easy. For instance, one can create a ‘normal EEG’ databank, a ‘child EEG’ databank, a ‘temporal lobe EEG’ databank, etc. Every database consists of one or more "pages". Within a single database, every page is the same size, but different databases can have page sizes that are powers of two between 512 and 65536, inclusive. The maximum size of a database file is 4294967294 pages. At the maximum page size of 65536 bytes, this translates into a maximum database size of approximately 1.4e+14 bytes (281 terabytes, or 256 tebibytes, or 281474 gigabytes or 256,000 gibibytes). This particular upper bound is untested since the developers of SQLite do not have access to hardware capable of reaching this limit. However, tests do verify that SQLite behaves correctly and sanely when a database reaches the maximum file size of the underlying filesystem (which is usually much less than the maximum theoretical database size) and when a database is unable to grow due to disk space exhaustion 3. This is huge and more than needed for a lifetime. Consequently, one can create just one databank and classify all his reports inside it.

What EWR is not. EWR is not a complete text editor with a myriad of options but it can headily generate a Portable Document Format (PDF), standardized as ISO 32000. PDF is a file format developed by Adobe in 1992 to present documents, including text formatting and images, in a manner independent of application software, hardware, and operating systems4. Consequently, reports generated are easily printed and or distributed.

EWR is distributed freely on GNU GENERAL PUBLIC LICENSE, Version 3, 29 June 2007. Copyright (C) 2022. Free Software Foundation, Inc. <http://fsf.org/> . Everyone is permitted to copy and distribute verbatim copies of this software, but changing or use in a commercial product it is not allowed.

A statement of need: There is a small number of EEG softwares specific to write reports and store them in “shareable” databanks and not to our knowledge that there is any free.

Material and methods

EWR was written in a windows 11 machine. It was not tested in other operational systems or other configurations, but should work in windows 8 and 10. It is not expected to work in Linux or mac computers because some lines of code in python are different in those machines and EWR is packaged in an windows exec file. The ideal screen resolution of use is 1920x1080 pixels full HD. Some tools were used to write EWR, python program language, tkinter library, Reporlab library, PyCharm, books and web sites.

EWR is written in python 3.10.45. Python appears as computer language in the late 1980s created by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands. Its implementation began in December 1989. Van Rossum was the lead developer, until middle 2018. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions6. In 2022, Python 3.10.4 was released. Today python ranks as one of the most popular programming languages. Python characteristically is multi-paradigm, can be or not object-oriented and structured. It also supports functional programming, aspect-oriented programming and magic methods7. Many other paradigms are supported via extensions, including design by contract and logic programming. We used an object-oriented approach to build EWR.

The interface was built in Tkinter which is integrated to Python. Tkinter works with a collection of Tk widgets that support most application needs. Tkinter is the Python interface to Tk, the GUI toolkit for Tcl/Tk. Tcl/Tk is a scripting and graphics library developed by John Ousterhout, who was originally at University of California at Berkeley and later at Sun Microsystems. Currently, Tcl/Tk is developed and supported by the Scriptics Corporation, which Ousterhout founded. Tcl/Tk adapts very well to Win32 systems and MacOS8.

To create the final EWR report document we use ReportLab. It is a software library that lets you directly create documents in Adobe's Portable Document Format (PDF) using the Python programming language. ReporLab is a free open-source document creation engine for generating PDF documents and custom vector graphics.

The code was written in PyCharm Community, a dedicated Python Integrated Development Environment (IDE) with a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development9.

Books were also necessary, to cite Meier’s10, Summerfield’s11 and Roseman’s12 as well as 2 youtube channels Codemy.com13 and RfZorzi14. When we had doubts about how to fulfill a determinate result or the best way to do it, we went to the invaluable site <https://stackoverflow.com> where an enthusiastic and vibrant community of programmers join to learn, teach and help each other.

Icons for buttons are from <https://icons8.com/icon/>. It is a free site with lots of options for application implementation.

Results

EWR software consists of three core modules which are fully integrated in the background: EEG\_weaver\_multiple\_sqlite.py that build the databanks, when necessary, EEG\_weaver\_Reporter\_funcs.py that store main functions and EEG\_weaver\_Reporter.py which have some more functions and starts main application. To EWR interface with the user we created a graphical user interface (GUI) with interactive visual components. The GUI displays the icons for actions, texts objects for interaction, and canvas to write the actual report. EWR (GUI) has four tabs with different purposes.

Tab one “Edit Report and create PDF” (figure1) is the editing tab where you write the name and address of your clinic using four or five lines, the body of your report, reporter’s name and at last one line commentary. Header and Doctor fields must be limited to few lines, End field to one line, but EEG Report Body supports as many lines as you wish. Pdf Pages will be automatically added to the report. Each page will have a Header and a page number. Last page has the name of the author of the report. An image can be added to the reader and a signature image can be added to the bottom alongside reporter’s name.

Tab two “Add Clinical History” (figure2) is a page where the neurophysiologist registers and store additional information about the patient. This info is not printed in the main pdf but separately if needed. One must remember to update the document each time it is changed. The button “Save as New Report” creates a copy of what is on the canvas. For instance, if it is pressed while the document is being edited, a new report is added to the databank. To update the edited document, you use button “Save Changes before Closing”.

Tab three “Report List” (figure 3) shows the databank in use with all its individual reports in list. Columns in list have self explaining names. Here reports can be duplicated, deleted individually, deleted in groups or databank can be completely cleaned. Besides, any term can be searched in a report name and shown in list. After that all reports are restored in list hitting “Show All Reports” button. Arrows can align reports in ascending or descending order by the order of indexation. One last feature of this tab is “Data Bases” text button (DBb) (figure 4). DBb opens a new window where you create databanks instead of Reports. You can create as many databanks as you wish, for example, tda.db, abscense.db, West.db, etc . In Tab three EWR shows a list of reports. In ‘Data Base”, however, EWR conveys a list of databanks created that can be selected for use in item (13) of figure 3.

EEG\_weaver\_Reporter\_1.0.exe and a pdf example can be downloaded from <https://sinapsy.com.br/eeg_weaver_reporter_1-0/>. It must be copyed to an empty folder. When you click on the app two files are created (kanda\_multi\_sql3.db and test.db) that must remain in the same folder as EWR . To start working the first thing to do is to create a specific databank for reports (item 7, Tab2), such as, alzheimers.db, epilepsy.db, my\_reports.db or multiple databases. Then select created databank (item13, Tab2). Go to Tab1, fill the blanks save and print reports. Going back to Tab2 the report is saved in list.

Conclusion:

In developing countries many local companies digitally record and process EEG, unfortunately, some of their solutions to report EEG are very limited. EEG Weaver Reporter is a free and compact python solution to write reports keeping them in databanks for ease review and restoration.

Kanda RG, Cury IJ, Lovatel ATB, Kanda PAM 2014. Sugestões práticas para a descrição do EEG de rotina. *Revista Neurociências*, *22*(3), 464-468.

2. Gramfort A, Luessi M, Larson E, et al. MNE software for processing MEG and EEG data. 2014;86:446-460.

3. Documentation S. Limits In SQLite [online]. Available at: <https://www.sqlite.org/limits.html>.

4. Bienz T, Cohn R, Systems A. Portable document format reference manual: Citeseer, 1993.

5. Inden M. Short Introduction to Python 3.10. Python Challenges: 100 Proven Programming Tasks Designed to Prepare You for Anything. Berkeley, CA: Apress, 2022: 635-643.

6. Peterson BJPI. Python Insider: Python 2.7. 18, the last release of Python 2. 2020.

7. The Python Language Reference. Python Software Foundation. Archived from the original on 15 December 2018. Retrieved 14 june 2022. [online]. Available at: <https://docs.python.org/3.0/reference/datamodel.html#special-method-names>.

8. Grayson JE. Python and Tkinter programming: Manning Publications Co. Greenwich, 2000.

9. 2022.1 JP. Choose the best PyCharm for you﻿. [online]. Available at: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>.

10. Meier B. Python GUI Programming Cookbook: Develop functional and responsive user interfaces with tkinter and PyQt5: Packt Publishing Ltd, 2019.

11. Summerfield M. Python in practice: create better programs using concurrency, libraries, and patterns: Addison-Wesley, 2013.

12. Roseman M. Modern Tkinter for Busy Python Developers: Quickly Learn to Create Great Looking User Interfaces for Windows, Mac and Linux Using Python's Standard GUI Toolkit: Late Afternoon Press, 2020.

13. Elder J. Create Graphical User Interfaces With Python And TKinter. Codemy.com [online]. Available at: <https://www.youtube.com/channel/UCFB0dxMudkws1q8w5NJEAmw>.

14. Serafim R. Tkinter course.RfZorzi [online]. Available at: <https://www.youtube.com/watch?v=RtrZcoVD1WM&t=41s>.

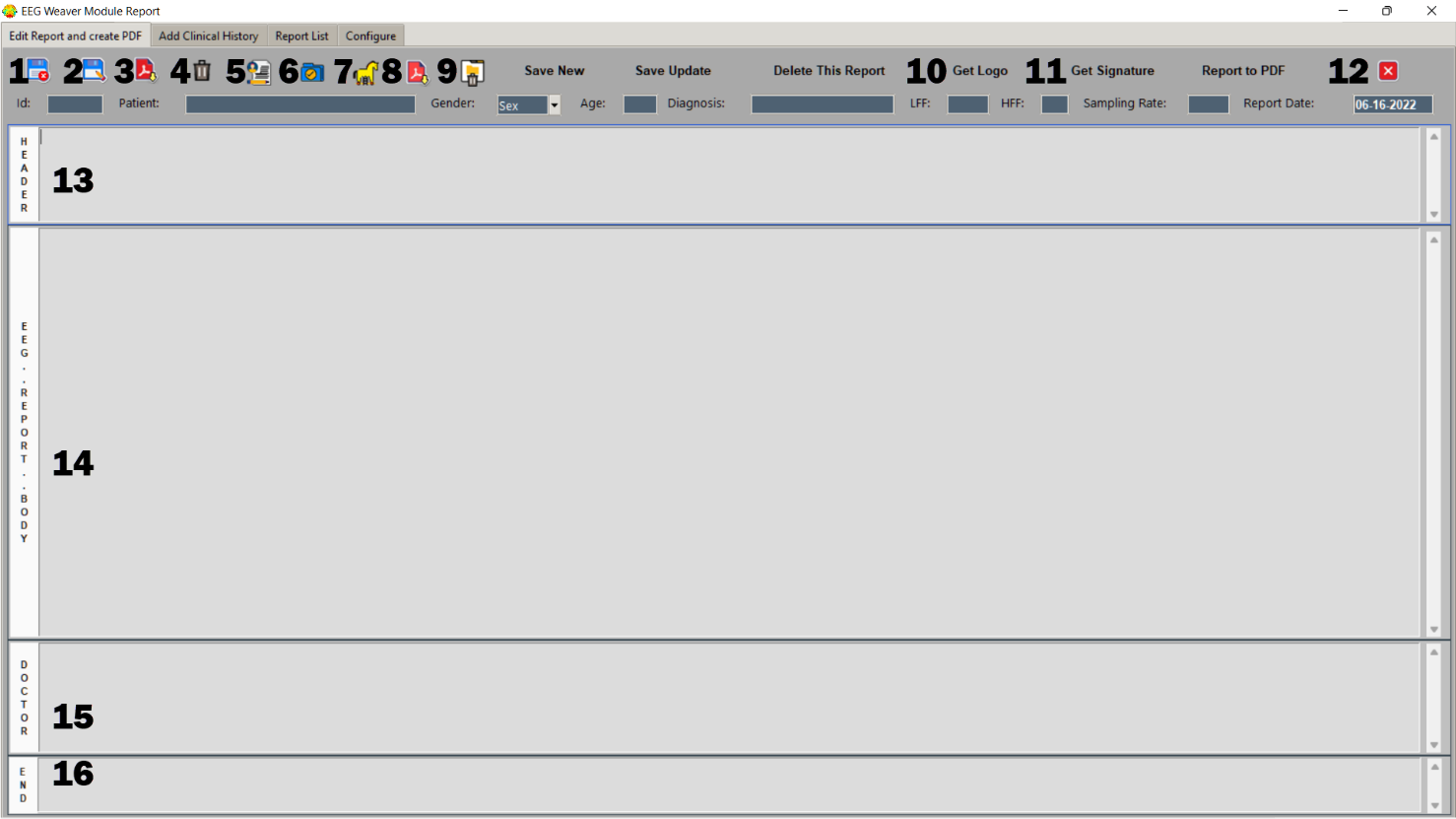


Figure 1. Tab one “Edit Report and create PDF”. Each number corresponds to a specific task: (1) save what is typed as a report in the databank, (2) after rewriting the text it is updated here, (3) generates final PDF, (4) delete everything typed, (5) delete report but don’t modify clinical story in next page, (6) save clinical story next page, (7) delete history only (small troyan horse), (8) generate PDF only for clinical story, (9) complete remove this report and History from databank, (10) get image (jpg or png) from folder to decorate the header, (11) get image of a signature to decorate footer, (12) close application, (13) header with clinic name and address, (14) multipage body of report, (15) footer where reporter types his name and credentials, (16) free 1 line space for final message.

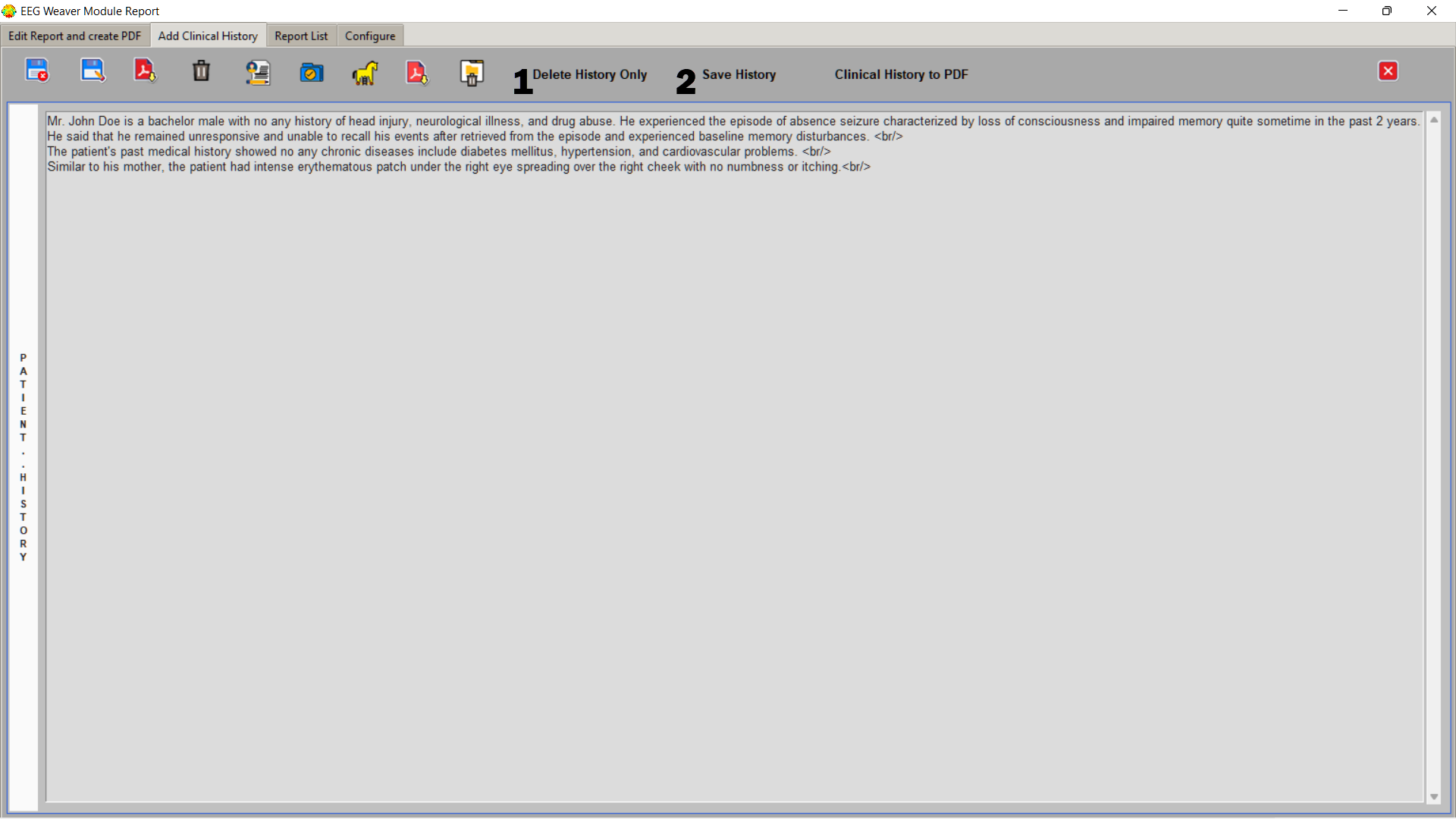


Figure 2. Tab 2 “Add Clinical History” has two new text options alongside with buttons: (1) “Delete history only” which erases only this page content and (2) “Save History” to update patient’s clinical log before closing.

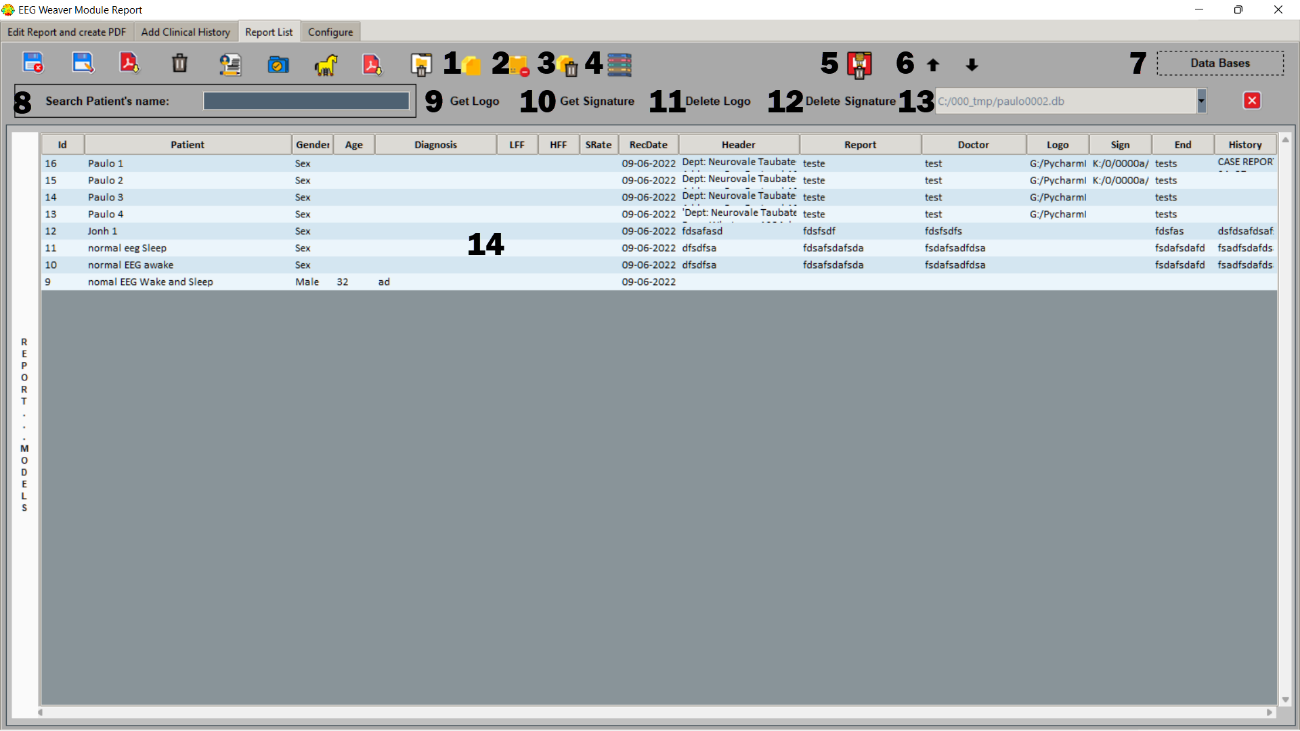


Figure 3. Tab 3 “Report List”. Some new functionalities (1) button “Duplicate Report”, (2) ‘Delete Selected”, (3) “Delete Many”. More than one record can be selected with [Ctrl-left mouse button]. Button (4) show all reports after your search, (5) delete all reports, (6) arrows change order of report appearance in list, (7) open new window for databank creation. Button (8) is an entry widget to insert s a term searched in “Patient” column. To insert a logo image and a signature image in one patient report you use buttons (9) and (10) respectively. To remove them you use buttons (11) and (12). Choose a databank (13) created in “Data Bases”. Reports inside a databank are shown in (14).

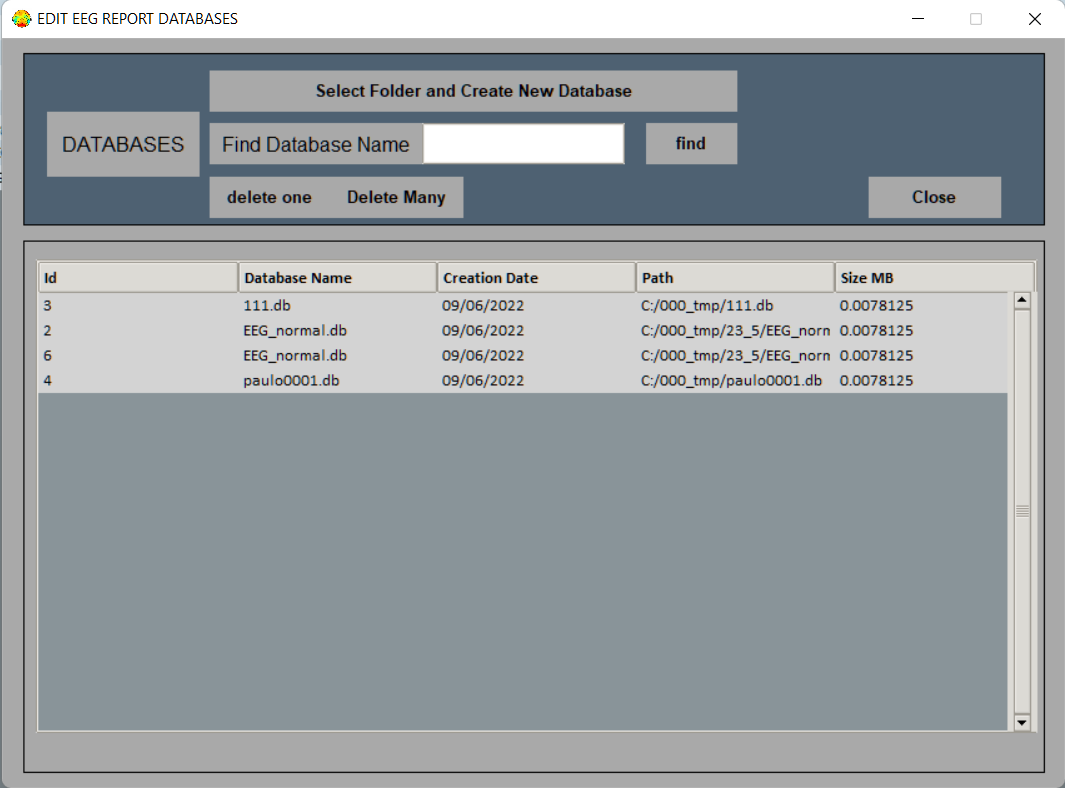


Figure 4. “Edit EEG Report Databases” is self explaining. Here each line represents a complete databank. “Delete many” requires {Cntrl-left mouse button] while clicking over the databanks.