# GLOSSARY

**Project structure:** is a MATLAB structure containing all the project parameters.

**GUI**: java interface, create through the NetBeans IDE.

**Components**: each graphical elements (text, combo, list, etc…)

# Main Functions

1. Create a new project structure
2. Load an existing project structure
3. Save/Save As the loaded project structure
4. Do analysis (select analyses, set a name, [modify output folder], start)

# TODO

1. Create a JPanel for each Tab of the JTPMain panel. Remove the elements now in the JTPMain window and move to the newly create JPanel. The present “Import & Data” tab will contain two panels (find how to arrange one below the other). At the end , the JTPMain panel will be empty, as all the tabs will be load at runtime by code (as we did for the preprocessing tab)
2. Give a name to each components of each tab. The idea is that the GUI structure might match, as much as possible, the fields of the project structure (the latter can be modified according to the GUI needing, this is the last chance to modify them !!). use the following naming convention:

## Variables naming conventions

Each *component* will be named according to a specific nomenclature: it must start with a predetermined suffix and followed by the corresponding project structure variable:

e.g.; project.preproc.ff1\_eeg => txt\_ff1\_eeg in the class representing the preproc tab

Read only labels: lab\_

Input text: txt\_

Listbox: lb\_

Datagrid: dg\_

Combobox: cmb\_

Checkbox: chk\_

Tree: tr\_

Normal variables should start with the following prefix.

Integer variables: i\_

Other number: f\_

String variable: s\_

Don’t know if mapping matlab variables type to java, let’s see, in case we decide to do it,…..

Cell array: ca\_

Struct array: sa

Numeric vector/matrix : a\_

1. Create a java class that maps the project structure

The idea is to create a java class that maps the project structure. All the operation done on the GUI will reflect on this data. It will also deal the load and save operations from/to java and matlab workspaces (reading a MAT file and writing a MAT & M file) and will set data to the GUI elements and retrieve from them. It will contain instances of other classes, each representing the fields and subfields of the project structure.

to implement its interaction with the GUI (e.g. setting the values), each panel class will implement the function setData(project) and getData(project) receiving its own project field as parameter. In the set/getGUI methods of the class you will have to call those methods.

## Java Class Project

Methods:

loadMat()….load

saveMat()

saveM()

setGUI()

getGUI()

### Properties.

Mirror all the fields and subfield of a typical project structure. One instance for each field of the project structure….e.g. import, data, preproc, etc…..

Instance of the JTPMain

# Classes Architecture

Design

Study

Epoching

.mat/m files

Data

Preprocessing

JPImport

JPProcessing

JPData

Project

Import

JTPMain