2013

# Integrated Analysis Platform of Brain Wave Data
# Design Document


Team Dec13-17

EE/CprE/SE Senior Design


12/9/2013

## Table of Contents

# Project Overview

## Problem Statement

The TDAM lab at ISU in cooperation with William and Mary are working to develop a method to perform spatial PCA-Mass Univariate analyses of event-related potential data. The statistical algorithms underpinning the analysis are currently written in MATLAB scripts. For this project, a user friendly GUI is needed to allow users without knowledge of line command control of MATLAB to use this method. The GUI will need to read in data from multiple vendors, allow the user to select constraints and perform analyses in a user-friendly manner, generate grand-averaged graphs and topographies per subject, and have the ability to export these graphs as publication-ready figures.

## General Solution

We propose to build a GUI around the pre-existing MATLAB files in a standalone application. Our solution will use the MATLAB scripts to perform all of the intensive mathematics then display it in a manner that can be easily interpreted. Our project will require refactoring of these scripts in order to successfully integrate into our solution, as well as modifications of these scripts in order to produce higher-quality plots and graphs. The goal is to take these results and be able to produce publication ready figures. The project requires some knowledge of MATLAB, matrix algebra, and manipulation of large data structures (e.g., 50MB to 1GB).

## Goals/Deliverables

- A multi-platform GUI application that integrates sPCA and MUA MATLAB scripts and reads data from 2-4 vendors software.
  - User will be able to select the type of analysis to perform on the data.
  - User will be able review the results that MATLAB outputs within the GUI.
  - System will output results to the user in publication-ready figures.
- Website for the GUI application that includes Download links, Documentation, Users Guide and Instructions/Tutorials.

# System Design

## Functional Requirements

***Priority Key***
***High:*** *Required for this implementation, one of the first functionalities to be built.*
***Medium:*** *Required for this implementation, but only if all High priorities are met first.*
***Low:*** *Not required for this implementation, but may be desirable later.*

### Application Entrance Requirements

1 When the user first enters the application, the system shall open on a leading window that allows the user to begin at multiple entrance points into the application.
**Priority:** Medium

a. The system shall allow the user to begin the application by building a data set via a "Build Data Set" option.
**Priority:** Medium

b. The system shall allow the user to begin the application by plotting Grand-Averages of all electrodes via a "Plot Grand-Averages" option.
**Priority:** Medium

c. The system shall allow the user to begin the application identifying spatial principle components, running parallel analysis, and performing spatial PCA via a "Identify Components" option, given they have built a data set output file.
**Priority:** Medium

d. The system shall allow the user to begin the application by plotting Spatial Principal Component Analysis results via a "Plot Spatial PCA Results" option, given they have ran Spatial PCA and generated an output file.
**Priority:** Medium

e. The system shall allow the user to begin the application by running Mass Univariate Analysis via a "Run Mass Univariate Analysis" option, given they have a spatial PCA output file.
**Priority:** Medium

f. The system shall allow the user to begin the application by plotting Mass Univariate Analysis results via a "Plot Mass Univariate Analysis Results" option, given they have ran Mass Univariate analysis and generated an output file.
**Priority:** Medium

### Build Data Set Requirements

2 The system shall read input from major vendors and generic text files at the level of averaged files, and output the formatted data set as an output file.
**Priority:** High

a. The system shall allow the user to name conditions for input.
**Priority:** High

b. The system shall allow the user to select input files for subjects.
**Priority:** High

c. The system shall determine and display the number of subjects and conditions to the user.
**Priority:** High

d. The system shall allow the user to select the electrode file (vendor) of the data input.
**Priority:** High

e. The system shall allow the user to select electrodes in the data input.
**Priority:** High

f. The system shall allow the user to input sampling rate of the input, in Hertz.
**Priority:** High

g. The system shall allow the user to input the Baseline of the input, in milliseconds.
**Priority:** High

h. The system shall allow the user to select the name and directory of the output file.
**Priority:** High

i. The system shall take the input files along with the input parameters and build a data set of the input. This data set will be stored in the output file specified by the user.
**Priority:** High

### Data Set Checking and Plot Grand-Average Requirements

3 After reading input and writing to the output file, the system shall display data in the following ways for data checking purposes:
**Priority:** High

a. The system shall display the data as a 2D arrangement of electrodes to the user to ensure proper electrode selection.
**Priority:** High

b. The system shall display this data as full montage data to the user to ensure correct data.
**Priority:** Medium

c. The system shall display this data at the grand-averaged level (of all electrodes) for supplementary data checking.
**Priority:** Medium

d. The system shall also be able to display this data at subject level for supplementary data checking.
**Priority:** Low

4 After data checking, the system shall ask the user via dialog if they are satisfied with their data.
**Priority:** Medium

a. If the user is not satisfied with their data input, the system shall redirect the user back to the data input screen.
**Priority:** Medium

b. If the user is satisfied with their data input, the system shall continue the user to the Leading Window screen.
**Priority:** Medium

### Identify Components Requirements

5 After output file and data checking, the system shall allow the user to select a dataset in which to Identify Components, requiring user input to where the dataset is located via a file browser.
**Priority:** High

6 The system shall allow users to input epoch range for Parallel Analysis and sPCA. The epoch will have a start and end range to input.
**Priority:** High

7 The system shall allow users to select electrodes for Parallel Analysis and sPCA.
**Priority:** High

8 After the user selects epoch and electrodes, the system shall perform Spatial PCA by calling MATLAB scripts.
**Priority:** High

a. The system shall provide a "Run Parallel Test" option for the user to run parallel test on the data. This will return an integer of the number of spatial components present to the user.
**Priority:** High

b. The system shall provide a "Run sPCA" option for the user to perform the Spatial PCA Analysis. This will take the data from the user and perform the Spatial PCA by calling MATLAB scripts.
**Priority:** High

9 After running Spatial PCA Analysis, the system shall allow the user to view the results.
**Priority:** High

a. The system shall allow the user to view the output as Raw, Promax rotation, and Varimax rotation pattern matrices.
**Priority:** High

b. The results shall appear as individual virtual ERP figures of each topography.

**Priority:** High

c. The system shall output topographies for source analyses and inform the user which file the results have been output to.

**Priority:** High

### Mass Univariate Analysis Requirements

10 The system shall have the ability perform Massive Univariate analysis

**Priority:** High

a. The system shall require the user to input the sPCA results from a file browser.

**Priority:** High

c. The system shall allow the user to input Independent Values (1-2 IV's only), as well as number of levels (2-3 levels for each IV) and a label for each IV.

**Priority:** High

d. The system shall allow the user to select threshold values for p and false discovery rate.

**Priority:** High

e. The system shall allow the user to include group as a factor in the analyses.

**Priority:** Low

11 After inputting parameters for Mass Univariate Analysis, the system shall allow the user to run the analysis via a "Run Mass Univariate Analysis" button.

**Priority:** High

12 After running the Mass Univariate Analysis, the system shall allow the user to plot the Mass Univariate Analysis results.

**Priority:** High

a. The system shall allow the user to plot the Mass Univariate results as a virtual ERP.

**Priority:** High

b. The system shall allow the user to plot the Mass Univariate results as Contrasts of Virtual ERP's.

**Priority:** Medium

## Non-Functional Requirements

***Priority Key***
***High:*** *Required for this implementation, one of the first functionalities to be built.*
***Medium:*** *Required for this implementation, but only if all High priorities are met first.*
***Low:*** *Not required for this implementation, but may be desirable later.*

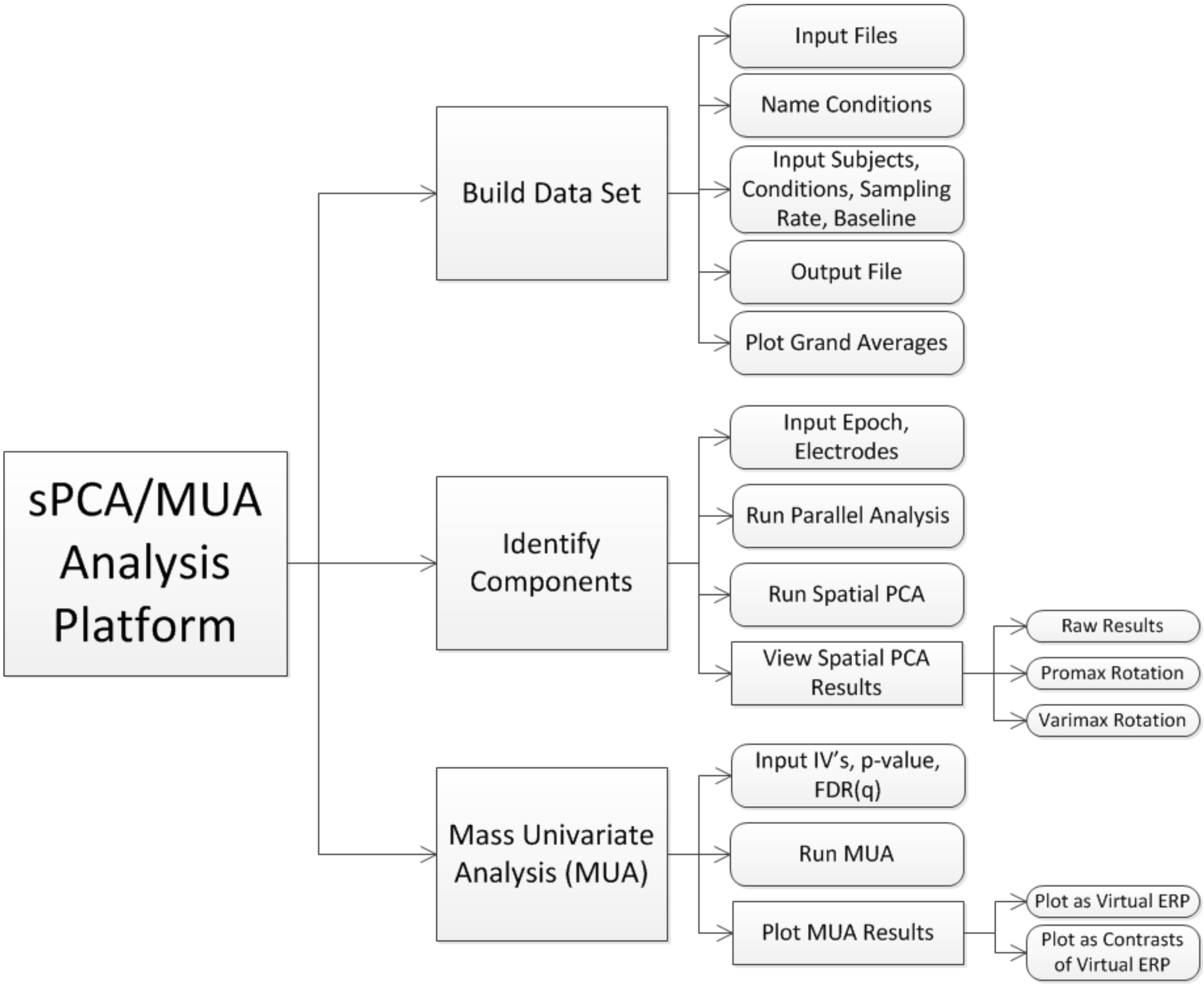
1 The system must be able to support data input from 2-4 vendors and provide a generic text file option for other users.
**Priority:** High

2 The input read from vendors must be represented as ASCII files or headered binary files.
**Priority:** High

3 The system shall be able to support data input up to 1 GB.
**Priority:** High

4 The GUI should be easy-to-use for a non-technical user.
**Priority:** High

## Functional Decomposition




## System Analysis

Our system will be based off of the functions described in the Functional Decomposition. Our Functional Decomposition represents a 2-dimensional design, it represents a horizontal top-down layered design as well as a vertical hierarchy of dependencies. From left to right, the abstraction of our design breaks down into subsections that grow significantly more concrete, until the rightmost components are as low as class-level components. From top to bottom represents the hierarchy of dependencies. In this system, a higher functionality must always be implemented before a lower functionality, because lower functionalities are dependent on higher ones.

The Functional Decomposition of the sPCA/MUA Analysis Platform can be broken down into 3 sections: the Build Data Set section, the Identify Components section, and the Mass Univariate Analysis Section. These three sections all have multiple functionalities and sub-functionalities, which are described as follows:

### Build Data Set Section

The Build Data Set section of this system is where the Data Set for the system is initially built to be run for sPCA analysis and MUA analysis. The functionalities needed to implement the Build Data Set section are as follows:

**Input Files:** This is the functionality that will allow the user will input the subject/condition text files used for analysis.
**Name Conditions:** After subject/condition files been input, this is the functionality that will name each condition, which is interactive with the user.
**Input Subjects, Conditions, Sampling Rate, Baseline:** This is the functionality that will allow the user to input number of subjects, number of conditions, sampling rate, and baseline.
**Output File:** This is the functionality associated with the output file. After the Dataset has been built, the output will be saved in an output .MAT file and the user will be notified of this file.
**Plot Grand Averages:** This is the functionality in which the system will generate grand average values of all electrodes. This plot will appear as a 2D arrangement of the electrodes.

After the dataset is built, the user will continue to Identify Components.

### Identify Components Section

The Identify Components section of this system is where the user will prompt the system to run Parallel Analysis and spatial Principal Component Analysis (PCA), as well as viewing the spatial PCA results to ensure proper component identification. The functionalities needed to implement the Identify functionality are as follows:

**Input Epoch, Electrodes:** This is the functionality that will allow the user will input the epoch and electrodes needed for sPCA Analysis.
**Run Parallel Analysis:** This is the functionality that will run the parallel analysis for the user. The system will return the number of spatial components after completing parallel analysis.
**Run Spatial PCA:** After parallel analysis has been run, this functionality will perform the spatial Principal Component Analysis for the user.
**View Spatial PCA Results:** After the Spatial PCA has been completed, the system shall allow the user to view the sPCA results as virtual ERP topographies. The sPCA results can be represented in 3 various forms, requiring the system to implement 3 separate sub-functionalities:

- **Raw:** The raw pattern matrix that can be plotted as a topography.

- **Varimax:** The pattern matrix after it has undergone Varimax Rotation, which can be plotted as a topography.
- **Promax:** The pattern matrix after it has undergone Promax Rotation, which can be plotted as a topography.

After the parallel analysis and spatial PCA, the user will continue to Mass Univariate Analysis.

#### Mass Univariate Analysis (MUA) Section

The MUA section of this system is where the user will prompt the system to run the Mass Univariate analysis, plot the result of the analysis, and generate publication-ready figures of grand averages. The functionalities needed to implement the Mass Univariate Analysis functionality are as follows:

**Input IV's, p-value, FDR(q):** This is the functionality that will allow the user will input the conditions necessary for Mass Univariate Analysis.
**Run MUA:** This is the functionality in which the system will perform the Mass Univariate Analysis for the user.
**Plot MUA Results:** After the MUA has been completed, the system shall allow the user to plot theMUA results as virtual ERP's. The MUA results can be represented in 2 forms, requiring the system to implement 2 separate sub-functionalities:

- **Virual ERP:** The results will be plotted as a virtual ERP.
- **Contrasts of virtual ERP:** The results will be plotted as contrasts of virtual ERP.

## Implementation and Design Decisions

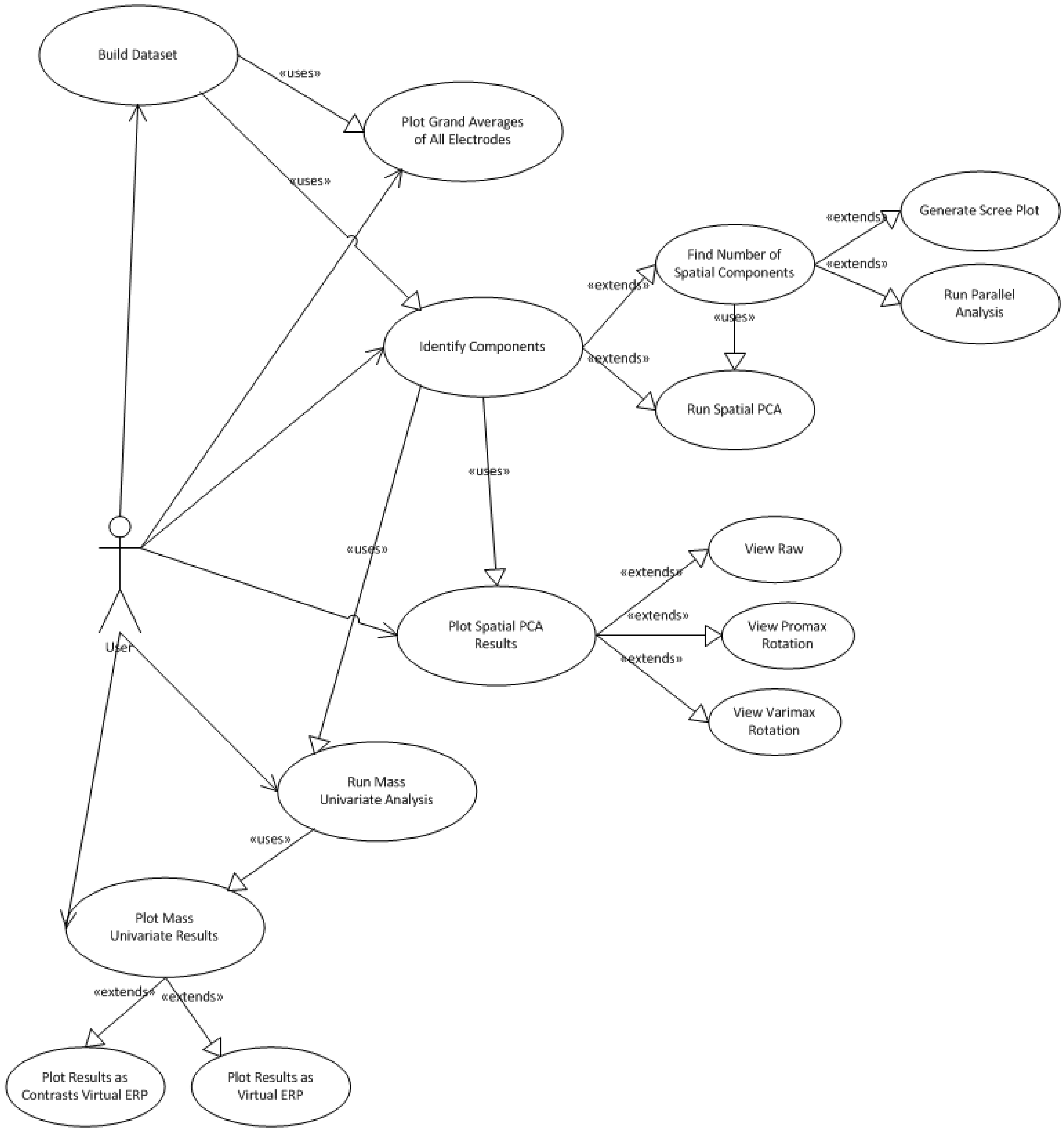
This section outlines some of the major implementation decisions and design decisions:

- This system is entirely a software system. There is no hardware to be designed whatsoever.
- The system is designed to work exclusively as a MATLAB application, as constrained by the client. All code will be written using MATLAB scripts, libraries, structures, and figures. In order for a user to run the system, they must have MATLAB installed on their computer.
- The Screen Flow and User Interface were developed by some high-priority user constraints. The client required a "Leading Window" with multiple entry points into the application, because there is a need to do each step one at a time, and frequently revert back to previous steps. The layout and flow of the GUI was developed over numerous usability discussions with our clients.

- As specified by the client, there are no networking-related requirements. There are requirements for the user to publish their graphical results. From there, the user may use the published results as they wish.
- This system is intended to be cross-platform. Any Operating System shall be able to use this application, given they have MATLAB installed on their computer. Any issues with cross-platform compatibility are addressed in the MATLAB Multi-Platform Compatibility module of our Modular Design (see *Modular Design* and *Module Description* sections of this document for more information).

# Detailed Design

## Use Case Diagram



## Use Case Scenarios

# 1 Use Case: Build Dataset

## 1.1 Description

The user builds a dataset from raw text files containing subject/condition data.

## 1.2 Actors

User

## 1.3 Triggers

User clicks "Build Dataset" from the Leading Window.

## 1.4 Flow of events

### 1.4.1 Basic Flow

1. The user clicks "Name Conditions", and names the conditions they wish to input.
2. The user clicks "Input Files" and inputs the corresponding text files using a file browser.
3. The user inputs sample rate (in Hertz) by text box.
4. The user inputs baseline (in ms) by text box.
5. The user clicks "Output Files" and selects where to write the Dataset output file via a file writer.

### 1.4.2 Alternative Flows

None

### 1.4.3 Preconditions

The user has subject/condition data stored on their computer as raw text files.

### 1.4.4 Postconditions

The Dataset is successfully written, as a ".MAT" file.
User is redirected to "Plot Grand Averages".

### 1.4.5 Exceptional Conditions

2.a. The number of input files does not coordinate with expected number of subjects and conditions.

2.b. The system prompts a dialog to warn the user to re-select the input files.

## 1.5 Issues

None

# 2 Use Case: Plot Grand Averages of All Electrodes

## 2.1 Description

The user plots the 2D arrangement of electrodes within the Dataset as grand-averaged values.

## 2.2 Actors

User

## 2.3 Triggers

User clicks "Plot Grand Averages All Electrodes" from the Leading Window, or the user is redirected here by the system after "Build Dataset".

## 2.4 Flow of events

### 2.4.1 Basic Flow

1 The user clicks "Input Files" and inputs the dataset they wish to plot.
2 The plot of the grand-averaged value of each electrode in the Dataset is generated for the user as a 2D arrangement of all electrodes.

### 2.4.2 Alternative Flows

None

### 2.4.3 Preconditions

The user has built a Dataset of the data they wish to plot.

### 2.4.4 Postconditions

User is redirected to "Plot Spatial PCA Results".

### 2.4.5 Exceptional Conditions

1.a. The provided input file is not a valid Dataset.
1.b. The system prompts a dialog to warn the user to re-select the input files.

## 2.5 Issues

None

# 3 Use Case: Identify Components

## 3.1 Description

The user runs Spatial PCA analysis from a Dataset, and outputs the Spatial PCA Results file.

## 3.2 Actors

User

## 3.3 Triggers

User clicks “Identify Components” from the Leading Window.

## 3.4 Flow of events

### 3.4.1 Basic Flow

1. The user clicks “Input Files” and inputs the dataset they wish to analyze.
2. The user inputs the Epoch Start and End times, which specifies desired time frame to be analyzed.
3. The user selects which Electrodes they wish to analyze.
4. The user clicks “Run the Parallel Analysis”, which returns an Integer of the estimated number of spatial components.
5. The system autofills the “Number of Spatial Components to Retain” text box as the integer returned in step 4. The user may edit this value if desired.
6. The user clicks “Run the Spatial PCA”, and the system runs the Spatial Principal Component Analysis.
7. After the analysis is complete, the system prompts the user to determine where to write the Spatial PCA Results File via a file writer.

### 3.4.2 Alternative Flows

4. The user clicks “Run the Scree Plot” which generates the Scree Plot.
5. The user then determines the Spatial Component value from the graph, and input their desired value into the “Number of Spatial Components to Retain” text box.

### 3.4.3 Preconditions

The user has built a Dataset of the data they wish to analyze.

### 3.4.4 Postconditions

The Spatial PCA Results file is successfully written, as a ".MAT" file.
User is redirected to "Plot Spatial PCA Results".

### 3.4.5 Exceptional Conditions

2.a. The provided input file is not a valid Dataset.
2.b. The system prompts a dialog to warn the user to re-select the input file.

## 3.5 Issues

None

# 4 Use Case: Plot Spatial PCA Results

## 4.1 Description

The user plots the Spatial PCA results.

## 4.2 Actors

User

## 4.3 Triggers

User clicks "Plot the Spatial PCA results" from the Leading Window, or the user is redirected here by the system after "Identify Components".

## 4.4 Flow of events

### 4.4.1 Basic Flow

1. The user clicks "Input Files" and inputs the Spatial PCA Results file to be plotted.
2. The user clicks "Raw Results".
3. The plot of the Raw Results from the Spatial PCA Results file are generated for the user as a topography.

### 4.4.2 Alternative Flows

**Promax Rotation Alternative**

2. The the user clicks on “Promax Rotation”.
3. The plot of the results from the Spatial PCA Results file after Promax rotation are generated for the user as a topography.

**Promax Rotation Alternative**
2. The the user clicks on “Varimax Rotation”.
3. The plot of the results from the Spatial PCA Results file after Varimax rotation are generated for the user as a topography.

#### 4.4.3 Preconditions

User has created the Spatial PCA Results file they wish to plot.

#### 4.4.4 Postconditions

User is redirected to the Leading Window.

#### 4.4.5 Exceptional Conditions

1.a. The provided input file is not a valid Spatial PCA Results file.
1.b. The system prompts a dialog to warn the user to re-select the input files.

### 4.5 Issues

None

## 5 Use Case: Run Mass Univariate Analysis

### 5.1 Description

The user runs the Mass Univariate Analysis from a Spatial PCA Results file, and outputs the Mass Univariate Results file.

### 5.2 Actors

User

### 5.3 Triggers

The user clicks “Run Mass Univariate Analysis” from the Leading Window.

### 5.4 Flow of events

#### 5.4.1 Basic Flow

1 The user clicks “Input File” and inputs a Spatial PCA Results file
2 The user provides the Number of Independent Variables (IV) via text box.
3 The user inputs the number of levels for each IV via text box.
4 The user inputs a label for each IV via text box.
5 The user inputs the p-value via text box.
6 The user inputs the False Discovery Rate (FDR) via text box.
7 The user inputs the Threshold width (in ms) via text box.
8 The user click “Run the Massive Univariate Analysis”, and the system runs the Mass Univariate Analysis.
9 After the analysis is complete, the system prompts the user to determine where to write the Mass Univariate Results File via a file writer.

#### 5.4.2 Alternative Flows

None

#### 5.4.3 Preconditions

User has created the Spatial PCA Results file they wish to analyze.

#### 5.4.4 Postconditions

The Mass Univariate Results file is successfully written, as a “.MAT” file.
User is redirected to “Plot Mass Univariate Results”.

#### 5.4.5 Exceptional Conditions

**Invalid Spatial PCA Results File**

1.a. The provided input file is not a valid Spatial PCA Results file.
1.b. The system prompts a dialog to warn the user to re-select the input files.

**Invalid Number of Levels for each Independent Variable**

3.a. The user inputs an invalid number of levels for each IV (constrained to 2-3).
3.b. The system warns the user to re-input the number of levels.

### 5.5 Issues

None

# 6 Use Case: Plot Mass Univariate Results

### 6.1 Description

The user plots the Mass Univariate Analysis results.

## 6.2 Actors

User

## 6.3 Triggers

User clicks “Plot Mass Univariate Results” from the Leading Window, or the user is redirected here by the system after “Run Mass Univariate Analysis”.

## 6.4 Flow of events

### 6.4.1 Basic Flow

1. The user clicks “Input Files” and inputs the Mass Univariate Results file to be plotted.
2. The user clicks “Plot Results as Virtual ERP”.
3. The plot of the results from the Mass Univariate Results file is generated for the user as a Virtual ERP.

### 6.4.2 Alternative Flows

2. The user clicks “Plot Results as Contrasts Virtual ERPs”.
3. The plot of the results from the Mass Univariate Results file is generated for the user as contrasts of multiple Virtual ERPs.

### 6.4.3 Preconditions

User has created the Mass Univariate Results file they wish to plot.

### 6.4.4 Postconditions

User is redirected to the Leading Window.

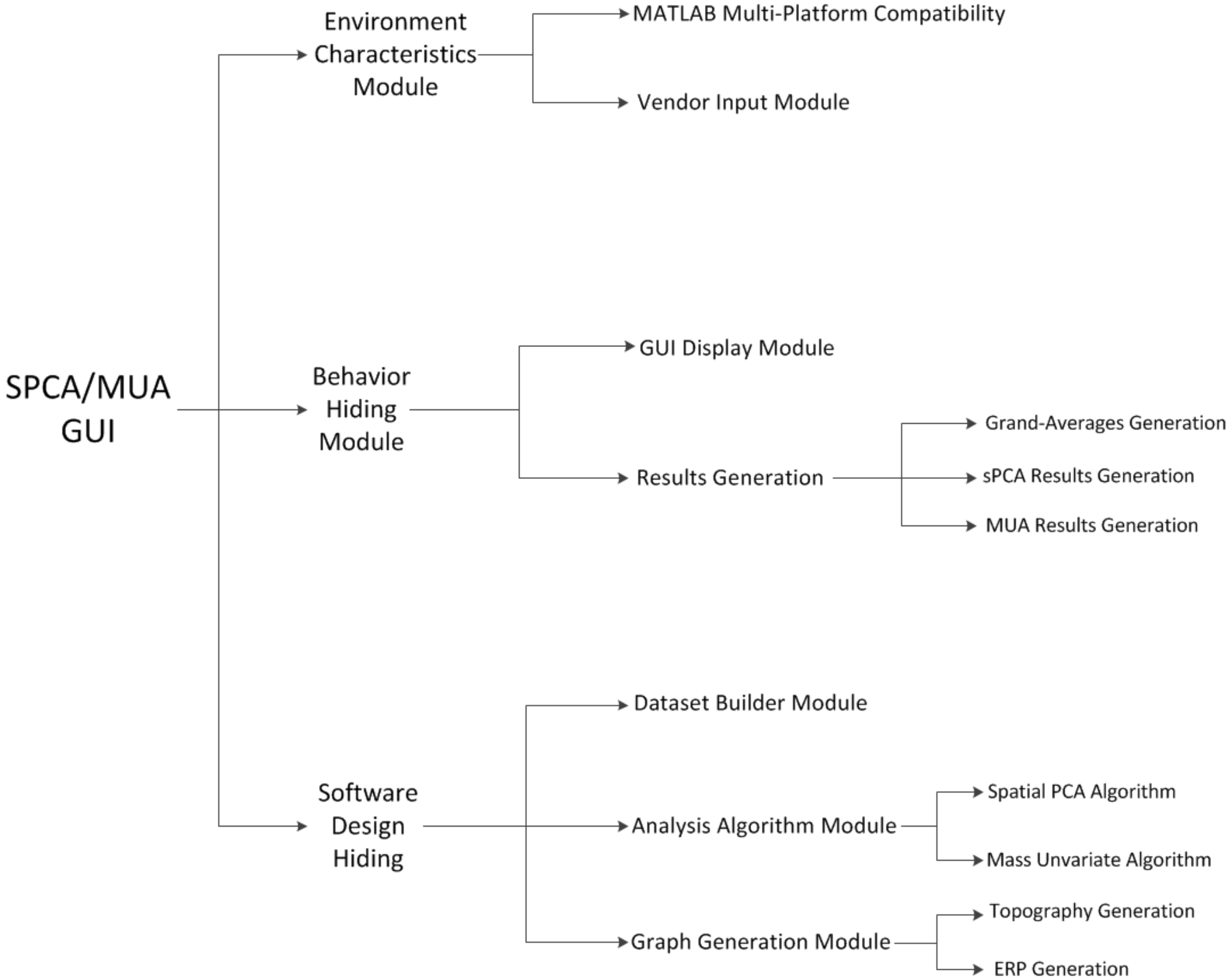
### 6.4.5 Exceptional Conditions

1.a. The provided input file is not a valid Mass Univariate Results file.
1.b. The system prompts a dialog to warn the user to re-select the input file.

## 6.5 Issues

None

## Modular Design




## Module Description

This document describes the module structure by characterizing each module's secrets. It also provides a brief description of the services provided by the module.

### 1 Environment Characteristics Module

The environment characteristics modules consist of those programs that need to be changed if the multi-platform requirements for MATLAB or the vendor input into the application change. The secret of the environment characteristics modules is the interfaces between the SPCA/MUA GUI, the platform it runs on, and the vendors whose input will be supported.

### 1.1. MATLAB Multi-Platform Compatibility

*Service:* Provides the same execution environment regardless of OS platform.

*Secret:* How the system adapts to perform on different platforms.

### 1.2. Vendor Input Module

*Service:* Provides a method for input of vendors' data.

*Secret:* The details of vendors' data.

## 2. Behavior Hiding Module

The Behavior Hiding Modules include programs that need to be changed if the required outputs from the system and the conditions under which they are produced are changed. Its secret is when (under what conditions) to produce which outputs. Programs in the Behavior Hiding Module use programs in the Software Design Module to Read Datasets, Run Analysis Algorithms, and Generate Graphs.

### 2.1. GUI Display Module

*Service:* Provides screen transitions that are intuitive to the user.

*Secret:* How, and under what conditions, the system transitions between screens.

### 2.2. Results Generation

*Service:* Provides results of analyses for the user.

*Secret:* How the analyses are performed, and how they are output to the user.

The Results Generation Module is decomposed into 3 functional submodules, one for each type of result:

#### 2.2.1. Grand-Averages Generation

*Service:* Provides grand-averaged results to the user.

*Secret:* How the grand-averages are computed, and how the results are generated.

#### 2.2.2. SPCA Results Generation

*Service:* Provides results of the Spatial PCA to the user.

*Secret:* How the SPCA is performed, and how the results are generated.

#### 2.2.3. MUA Results Generation

*Service:* Provides results of the Mass Univariate Analysis to the user.

*Secret:* How MUA is performed, and how the results are generated.

## 3. Software Design Hiding

The Software Design Hiding modules hide software design decisions based upon programming considerations such as algorithmic efficiency. Both the secrets and the interfaces to this module are determined by software designers. Changes in these modules are more likely to be motivated by a desire to improve performance than by externally imposed changes. Programs in the Software Design Hiding Module use programs in the Environment Characteristics Module to determine optimization based on platform, and to input Vendor Data into the system.

### 3.1. Dataset Builder Module

*Service:* Build and store a Dataset from vendor input for use in analyses.

*Secret:* The data structures and algorithms used to build the Dataset.

### 3.2. Analysis Algorithm Module

*Service:* Perform the analysis and store the results.

*Secret:* The algorithms and data structures used for the analysis.

The Analysis Algorithm Module is decomposed into 2 functional submodules, one for each type of analysis:

#### 3.2.1. Spatial PCA Algorithm

*Service:* Perform the Spatial PCA and store the results.

*Secret:* The algorithms and data structures used for the Spatial PCA.

#### 3.2.2. Mass Univariate Algorithm

*Service:* Perform the Mass Univariate Analysis and store the results.

*Secret:* The algorithms and data structures used for Mass Univariate Analysis.

### 3.3. Graph Generation Module

*Service:* Generate a graph and display to the user.

*Secret:* The algorithms and data structures used to generate the graphs.

The Graph Generation Module is decomposed into 2 functional submodules, one for each type of graph:

#### 3.3.1. Topography Generation

*Service:* Generate a brain wave topography and display to the user.

*Secret:* The algorithms and data structures used to generate the topography.

#### 3.3.2. ERP Generation

*Service:* Generate an Event Related Potential of the brain wave data and display to the user.

*Secret:* The algorithms and data structures used to generate the ERP.

## Input/Output Specification

There various inputs and outputs of this system are specified below. They are grouped by high-level use case. Each use case has its own inputs and outputs:

### Build Dataset Use Case I/O

Build Dataset is the use case in which the user inputs their subject/condition data, and the system generates and outputs a *Dataset* that is used for further analysis in the application.

**Input:** Raw text files (.txt) containing subject/condition information, which contains information about electrode responses over time. These text files are input by the user via a file browser.

**Output:** MATLAB structure array (.MAT) that contains a vector containing all subjects'/conditions' data, vectorized. It also holds metadata including electrode selection (as an array), sample rate (in Hz), and Baseline (in ms). This .MAT file is known as the ***Dataset***. This name and directory of the dataset is specified by the user via a file writer.

### Plot Grand Averages Use Case I/O

Plot Grand Averages is the use case in which the user inputs the *Dataset*, and the system generates and outputs a plot of the 2D electrode arrangement as grand-averaged values.

**Input:** The *Dataset*, .MAT file containing MATLAB vectorized data for multiple subjects/conditions. The dataset is either input automatically, if continuing straight from Build Dataset, or it is input by the user via a file browser.

**Output:** Plot of the grand-averaged 2D arrangement of electrodes, in the form of a MATLAB-generated plot.

### Identify Components Use Case I/O

Identify Components is the use case in which the user inputs the *Dataset*, runs parallel analysis to discover the number of spatial components, and runs the spatial PCA (spatial Principal Components Analysis). The system computes the spatial PCA and outputs the results as an *sPCA Results file* for further analysis in the application.

**Input:** The *Dataset*, .MAT file containing MATLAB vectorized data for multiple subjects/conditions. The dataset is input by the user via a file browser.

**Output:** MATLAB structure array (.MAT) that contains the results from running spatial Principal Component Analysis. This .MAT file is known as the ***sPCA Results file***. This name and directory of the sPCA Results file is specified by the user via a file writer.

### Plot Spatial PCA Results Use Case I/O

Plot Spatial PCA Results is the use case in which the user inputs the *sPCA Results file*, and plots the results using either the raw data, varimax rotation, or promax rotation.

**Input:** The *sPCA Results file*, .MAT file containing MATLAB data from computing the spatial Principal Component Analysis. The sPCA Results file is either input automatically, if continuing straight from Identify Components, or it is input by the user via a file browser.

**Output:** Plot of the sPCA Results (in raw form, varimax rotation, or promax rotation, as specified by the user), in the form of a MATLAB-generated plot.

### Run Mass Univariate Analysis Use Case I/O

Run Mass Univariate Analysis is the use case in which the user inputs the *sPCA Results file* and runs Mass Univariate Analysis on this data. The system computes the Mass Univariate Analysis and outputs the results as a *Mass Univariate Results file* for further analysis in the application.

**Input:** The *sPCA Results file*, .MAT file containing MATLAB data from computing the spatial Principal Component Analysis. The sPCA Results file is input by the user via a file browser.

**Output:** MATLAB structure array (.MAT) that contains the results from computing Mass Univariate Analysis. This .MAT file is known as the ***Mass Univariate Results file***.

This name and directory of the Mass Univariate Results file is specified by the user via a file writer.

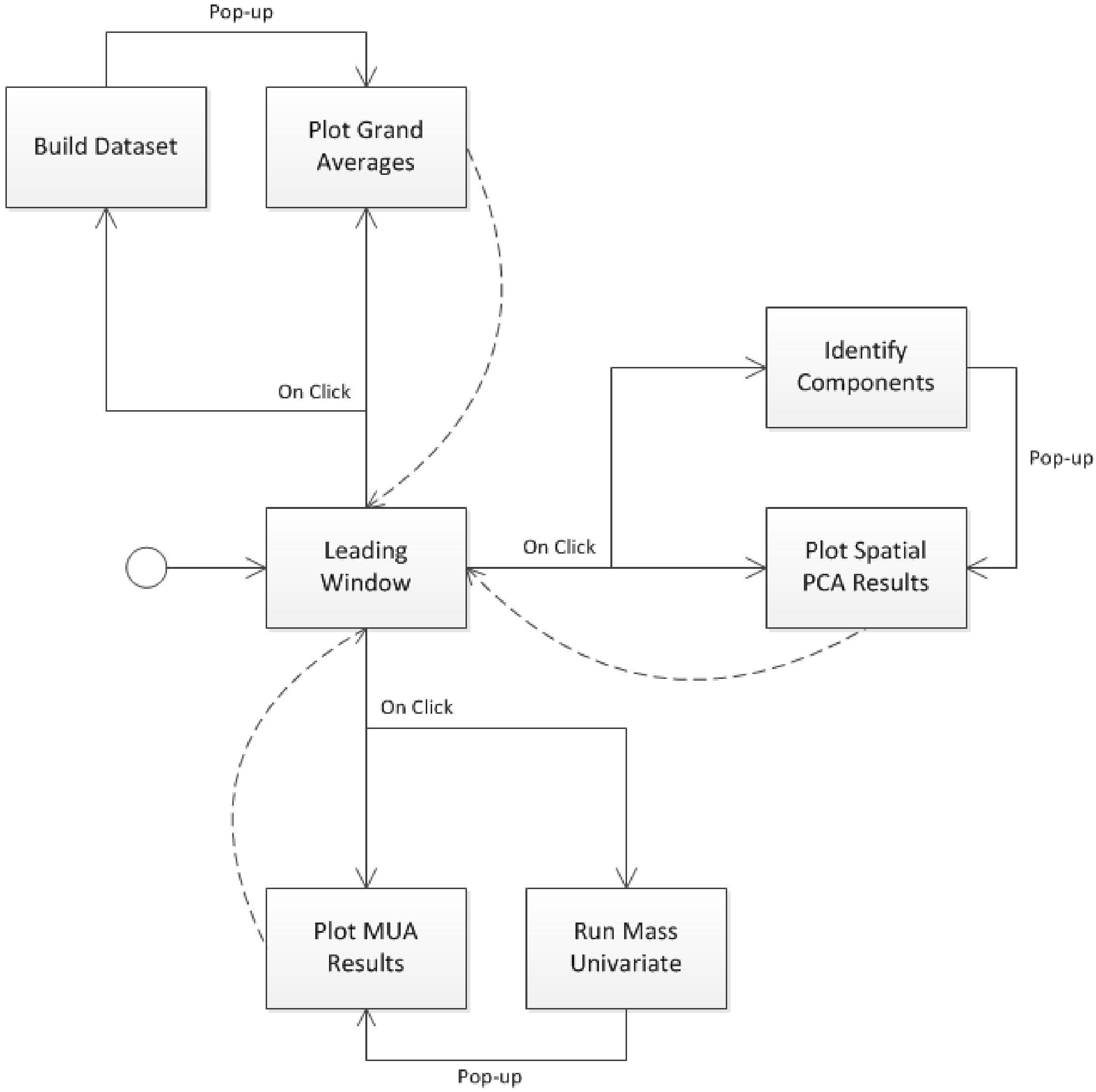
### Plot Mass Univariate Results Use Case I/O

Plot Mass Univariate Results is the use case in which the user inputs the *Mass Univariate Results file*, and plots the results as either Virtual ERP's or contrasts of Virtual ERP's.

**Input:** The *Mass Univariate Results file*, .MAT file containing MATLAB data from computing the Mass Univariate Analysis. The Mass Univariate Results file is either input automatically, if continuing straight from Run Mass Univariate Analysis, or it is input by the user via a file browser.

**Output:** Plot of the Mass Univariate Results (as Virtual ERP's or contrasts of Virtual ERP's, as specified by the user), in the form of a MATLAB-generated plot.

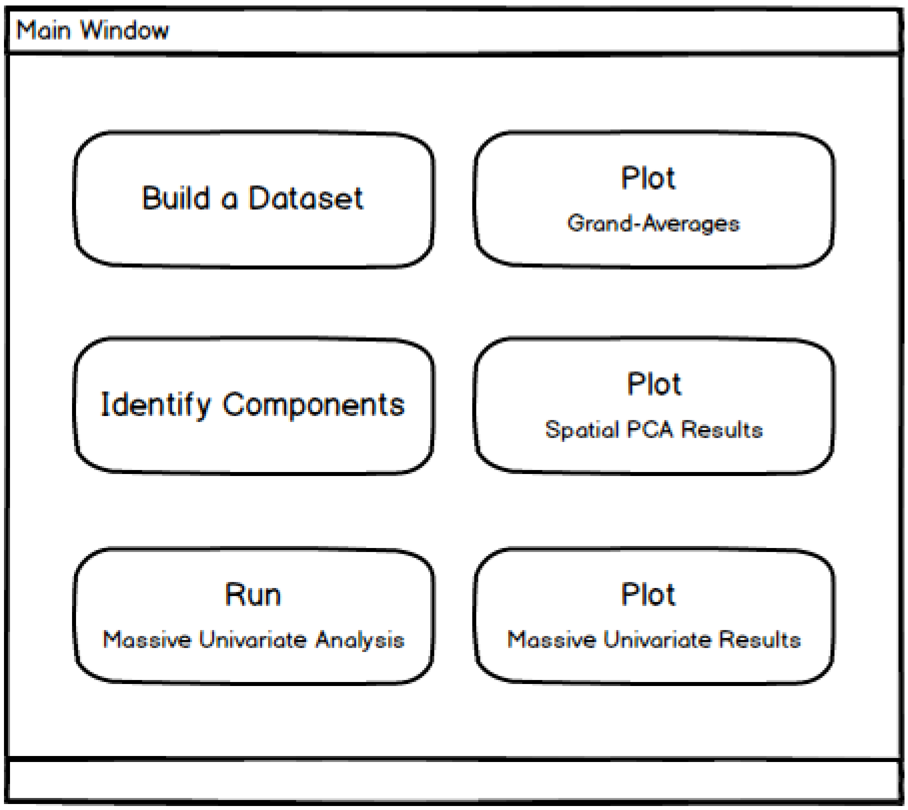
## Screen Flow Diagram



## User Interface Specification

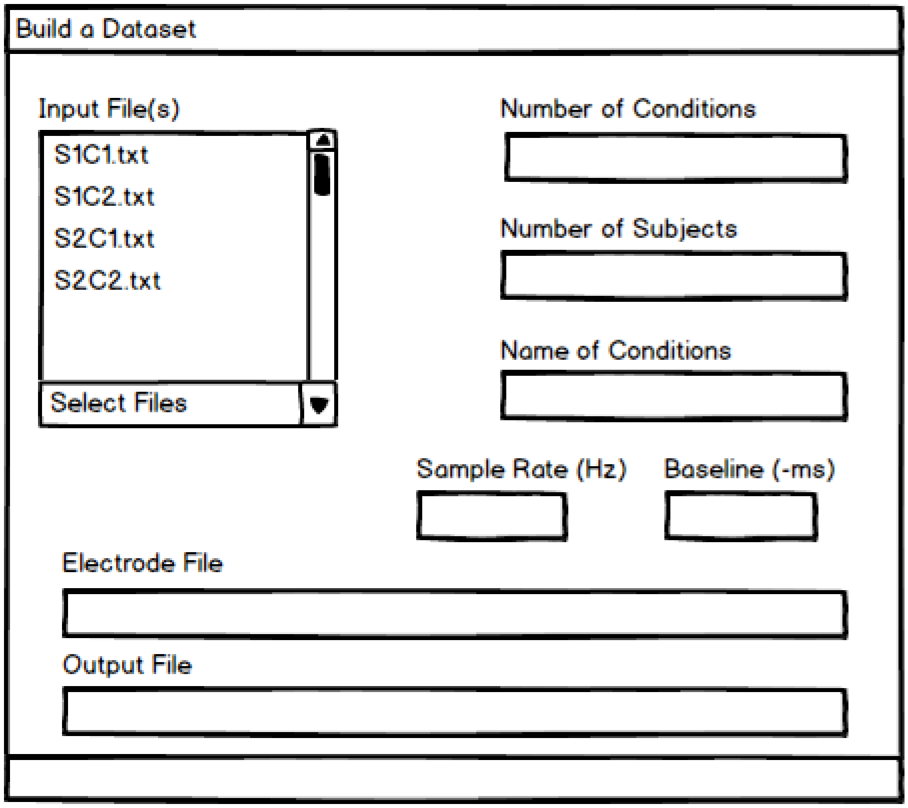
Listed below are screen sketches that are prototypes of the various screens our solution will feature. Our User Interface will be a standard two-dimensional MATLAB GUI featuring various buttons, links, and text.

### Leading Window




The leading window is the main entrance into our application. From here, you may enter the application from various starting points.

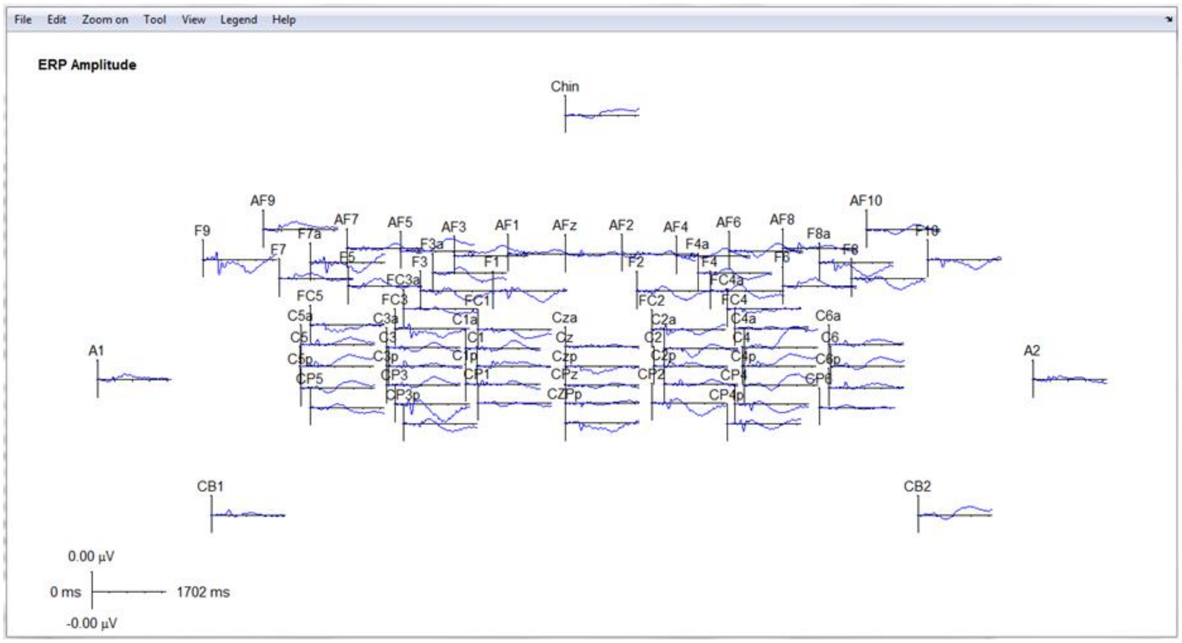
**Build Dataset**




The Build Dataset screen is where you input the data from various vendors. This data will be represented as ".txt" files, and will be accompanied by various parameters, including number of subjects/conditions, sampling rate, etc. When the user is ready to build the data set, they will specify the output file they wish the data set to be written to. This output Dataset (in ".MAT" format) can then be input in other parts of the application to perform analyses on. After building a Dataset, the user is redirected to the Plot Grand Averages screen.

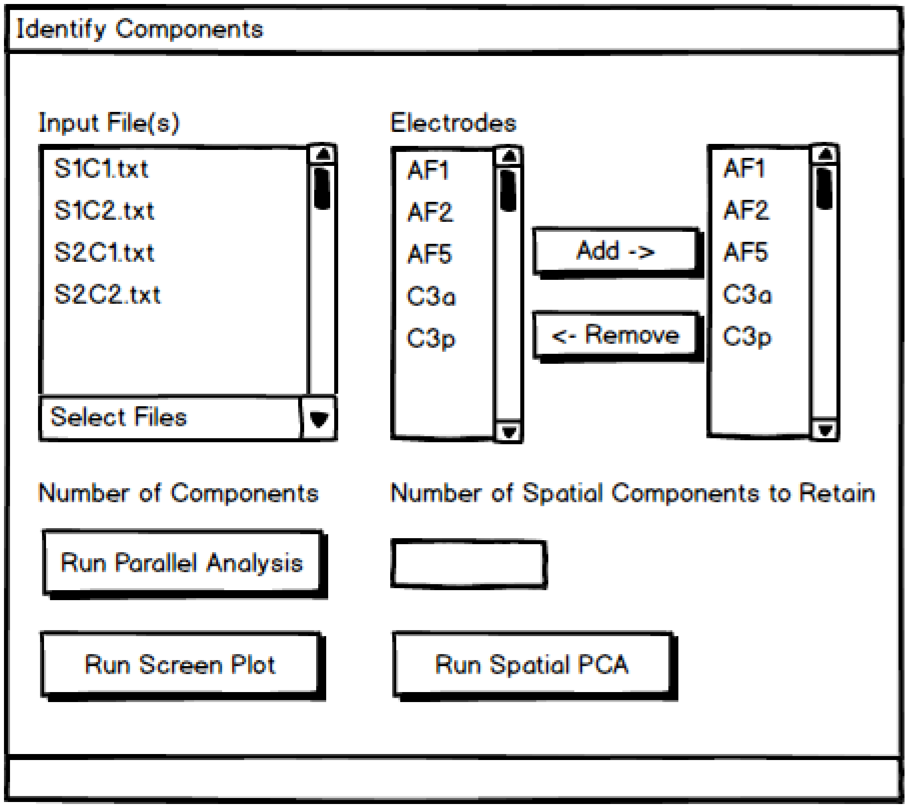
## Plot Grand Averages

Plot Grand-Averages
All Electrodes



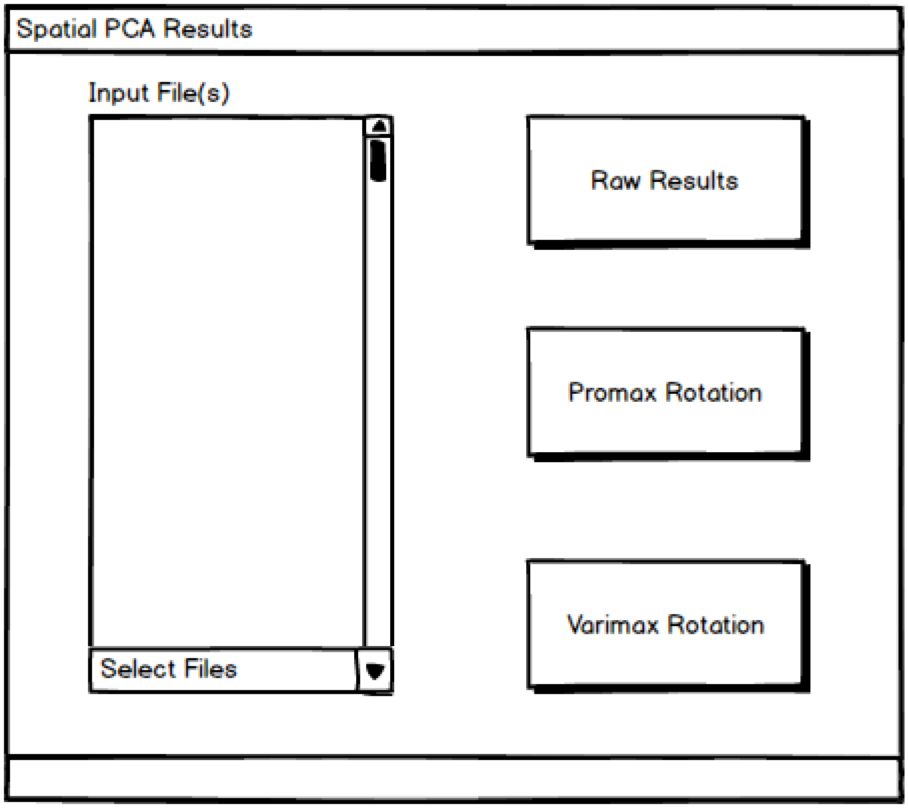

The user will have the ability to plot Grand-Averaged Data via the Plot Grand Averages screen. This screen will take data input as a dataset, and plot a 2D arrangement of all of these electrodes, as grand-averaged values of the subjects and conditions. After the user is finished with this screen, the user will be returned to the Leading Window.

**Identify Components**



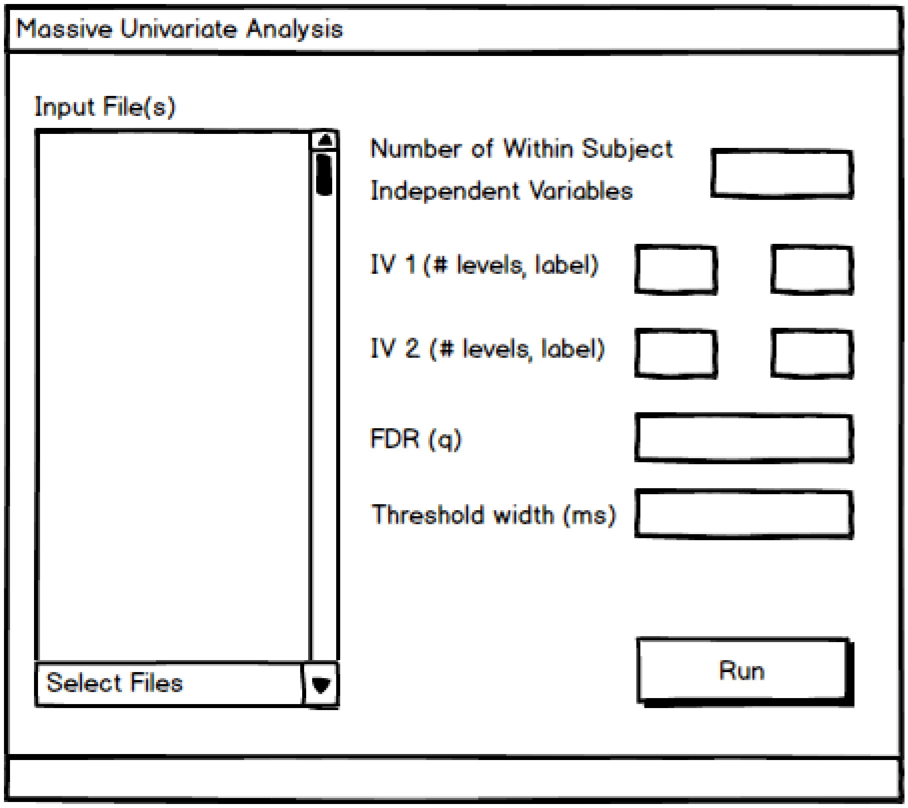

The Identify Components screen allows you to input a Data Set, and perform Parallel Analysis (or Scree Plot) and run Spatial PCA (Principal Component Analysis) on the Dataset. Before running these analyses, you must select epoch range and electrodes to run the analyses on. The results from running Spatial PCA will also be stored as a Spatial PCA Result file. After running spatial PCA, the user may view the results under the Plot Spatial PCA Results screen.

**Plot Spatial PCA Results**



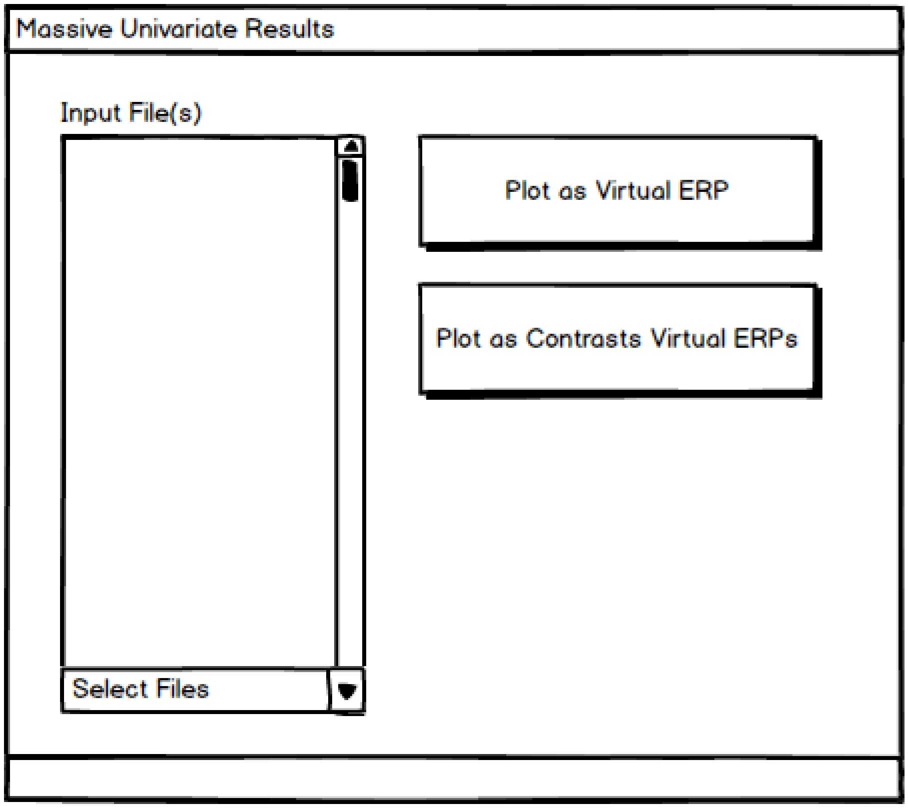

After running Spatial Principal Component Analysis, the user will be redirected to this Plot Spatial PCA Results screen. If the user starts at this screen from the Leading Window, they will be prompted with a file browser for the Spatial PCA Results file in which they wish to plot the Spatial PCA Results. If they are continuing from Identify Components, this will be pre-populated. The results can be viewed as topographies in Raw format, or with Promax and Varimax rotation.

**Run Mass Univariate Analysis**



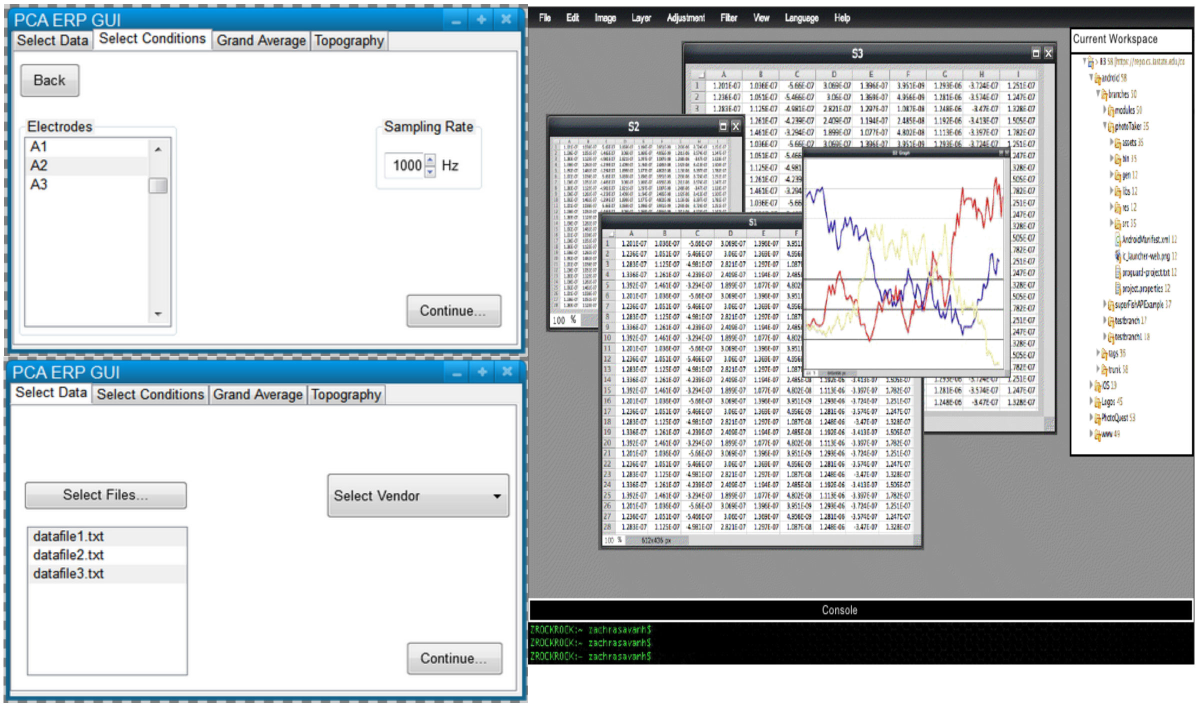

The Run Mass Univariate Analysis screen allows you to input the Spatial PCA Results file and run Mass Univariate Analysis on it. Before running this analysis, you must input the Independent Variables (only 1 or 2 allowed) as well as the number of levels and labels for each level. You must also input p-value, threshold width, and False Discovery Rate for proper analysis. After running the Mass Univariate Analysis, the application will redirect you to the Plot Mass Univariate Results screen. The results from running Mass Univariate Analysis will also be stored as a Mass Univariate Results file.

**Plot Mass Univariate Results**




After running Mass Univariate Analysis, the user will be redirected to this Plot Mass Univariate Results screen. If the user starts at this screen from the Leading Window, they will be prompted with a file browser for the Mass Univariate Analysis Results file in which they wish to plot the Mass Univariate Results. If they are continuing from Run Mass Univariate Analysis, this will be pre-populated. The results can be viewed as Virtual ERPs.

## Prototypes




We have been using multiple prototypes for this project. The majority of our prototypes thus far have been User Interface mockups/screenshots. We are also anticipating, by the end of this semester, an interactive wireframing prototype that will simulate transitions from screen-to-screen by the end of the semester. For this wireframe prototype we will be utilizing the prototyping tool Balsamiq® Our timetable for prototypes are as follows:

| | | |
|---|---|---|
| 1 | Initial screenshots/mockups: | **2/25/2013** |
| 2 | Refined screenshots: | **3/11/2013** |
| 3 | Initial Balsamiq® Wireframe Prototype: | **4/01/2013** |
| 4 | Refined Balsamiq® Wireframe Prototype: | **4/22/2013** |

# Test Specification

## Usability Testing Plan

After completing our Refined Balsamiq® Wireframe Prototype, we will present and distribute this prototype to members of the Department of Psychology for Usability Testing. We plan on performing this Usability Testing between May and August of 2013, and use the results to further refine our User Interface. Our Usability Testing will measure the following factors:

- Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- Efficiency: Once users have learned the design, how quickly can they perform tasks?
- Memorability: When users return to the design after a period of not using it, how easily can they're establish proficiency?
- Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction: How pleasant is it to use the design?

## Verification and Validation Plan

### I. Purpose

This document describes the verification and validation plan for the software developed by the Senior Design Team Dec13-17's SPCA/MUA GUI Project. *Validation* means "Did we build the right system?", (i.e., are we building the system that the customer wants, as expressed in the requirements?). *Verification* means "Are we building the system right?", (i.e., does the system conform to its specifications?).

### II. Methods

We will use a variety of validation and verification techniques for the software, as shown in the following list:

**1. Requirements review by stakeholders**, including Team Dec13-17, Dr. Weiss, Dr. West, and other potential customers as approved by Dr. West. Requirements reviews will be performed in the following ways:

a. *Review of Use Cases* by Dr. West and potential customers approved by Dr. West.

b. *Review of Use Cases, Functional Requirements, Non-Functional Requirements* by Team Dec13-17 and Dr. Weiss.

c. *Review of Prototypes* by Team Dec13-17 and Dr. Weiss to verify design is implemented correctly.

d. *Trial Use of Prototypes* by Dr. West and approved potential customers to validate performance and usability.

**2. Design reviews** by Team Dec13-17, and Dr. Weiss upon request.

a. *Modular Design Reviews*

b. *Functional Decomposition Reviews*

c. *Input/Output Specification Reviews*

d. *User Interface and Screen Flow Specification Reviews*

**3. Code reviews**, by Team Dec13-17, and Dr. Weiss upon request.

a. All code must be reviewed. For small commits to SVN (small fixes within a large module), only one other developer must review the code informally. For large commits (such as completion of a module, or integration between two modules), a formal code review will be held, with at least three other developers present.

**4. Testing**, by Team Dec13-17.

a. *Black-Box Testing* will be performed and documented through Test Cases, showing that the test passes the Requirements that are traceable to the test.

b. *White-Box Testing* will be performed by a developer before committing to SVN, assuring that the developer has tested all functionalities associated with the commit.

c. *Integration Testing* will be performed when integrating two or more modules, and will be traced against any Requirements and other Architectural Features in the Design Document.

d. *Stress Testing* will be performed after Black-Box testing, to ensure the same functionality is achieved when performing analyses on large sets of data (at least 1GB of data must pass stress tests).

## III. Schedule and Resources

These periodic reviews should be incorporated periodically during the design and development process as appropriate. As each module is completed it should immediately go under review, being scheduled and conducted by those responsible for that module.

## IV. Measures

Goals are to assure that there are no critical or major defects within the system created. In testing at least 70% of the code has been covered by tests in some fashion, with at most 10 minor errors after the first release of the application. Also there should be no more than 10 minor typographical errors per documentation, including the Project Plan, Design Document, Code Comments, and Test Cases.

## V. Acceptance Criteria

Consistent with the goals stated in section IV, *Measures*, our acceptance criteria for the first, and following, releases are as follows:

a. *Zero Critical Defects*. Critical Defects include inconsistencies between this release and all traceable High Priority Requirements.
b. *Zero major defects*. Major Defects include inconsistencies between this release and all traceable Medium Priority Requirements.
c. *At most 10 minor defects*. Minor Defects include inconsistencies between this release and all traceable Low Priority Requirements. The only exception is for the Final Release, which may have at most 5 minor defects.
d. *At most 10 typographical errors* in the Project Plan, and no other errors.
e. *At most 10 typographical errors* in the Design Document, and no other errors.
f. *At most 10 typographical errors* in the Code Comments, and no other errors.
g. *At most 10 typographical errors* in the Test Cases, and no other errors.
h. *Zero inconsistencies* among Project Plan, Design Document, Code Comments, and Test Cases.
i. *Code Coverage in Testing* should be at least 70%

## VI. Responsibilities

The entire Team Dec13-17 is responsible for ensuring that all acceptance criteria have been met. Upon a new release, Team Dec13-17 will have an Acceptance Criteria Review to decide whether or not all acceptance criteria have been met for this release, and will document their findings of the review, and plan their next release accordingly. The Acceptance Criteria Report will be distributed among Dr. Weiss, Dr. West, after it has been published.

## Standards

### Project Terminology

| Term | Definition |
|---|---|
| .m file(s) | MATLAB function, script, or class |
| .mat file(s) | MATLAB binary file for storing variables |
| Binary Files | A computer file that is not a text file; it may contain any type of data, encoded in binary form for computer storage and processing purposes. |
| Data Set | A collection of data, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. |
| Electrodes | Electrical conductor used to measure voltage fluctuations to measure neuronal activity. |
| Electroencephalography (EEG) | The recording of electrical activity along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain. |
| Electroencephalography Lab (EEGLAB) | An interactive MATLAB toolbox for processing continuous and event-related EEG, MEG and other electrophysiological data incorporating independent component analysis (ICA), time/frequency analysis, artifact rejection, event-related statistics, and several useful modes of visualization of the averaged and single-trial data. |
| Epoch | Moment in time, used when describing which time segments to chunk off of data sets (e.g. data has time from -200 to 500, but only want time from 0-200). |
| Event-Related Potential (ERP) | The measured brain response that is the direct result of a specific sensory, cognitive, or motor event. More formally, it is any stereotyped electrophysiological response to a stimulus. |

| | |
|---|---|
| Event-Related Potential Lab (ERPLAB) | A free, open-source MATLAB package for analyzing ERP data. |
| Graphical User Interface (GUI) | A type of user interface that allows users to interact with electronic devices using images rather than text commands. |
| Graphical User Interface Development Environment. (GUIDE) | The user interface development studio for MATLAB. |
| Independent Variables | The inputs or causes, or are tested to see if they are the cause. |
| Large Data Structures | Data structure greater than 50 MB? |
| Levels of Independent Variables | Independent Variables may have various levels, typically 2 or 3, which are essentially the number of dimensions within the IV itself. |
| Mass Univariate Analysis (MUA) | The analysis of a massive number of simultaneously measured dependent variables via the performance of univariate hypothesis tests (e.g., t-tests). |
| Matrix Laboratory (MATLAB) | A numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. |
| Partial Least Squares Graphical User Interface (PLSGUI) | GUI for performing brain data analyses utilizing Partial Least Squares. |
| Principal Component Analysis (PCA) | A mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. |
| Publication Ready | Data prepared to be presented in either a chart, diagram, and flow that can be used in other professional resources. |

| | |
|---|---|
| Spatial Principal Component Analysis (sPCA) | PCA performed on data with spatial components. |
| Subject | Person that the experiment was conducted on. |
| Temporal Dynamics of Attention and Memory (TDAM) | Neural basis of cognitive control and prospective memory using ERP methods. |