**COMPONENT SETTING MANAGEMENT**

**4.3 DESCRIPTION COMPONENT**

**4.1.1.PROCESSING NARRATIVE (PSPEC) FOR COMPONENT DATABASE MANAGEMENT**

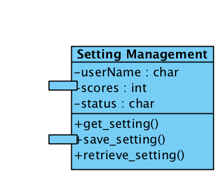
The intent of this component is to get the new settings and/or receive request on setting management and manage the settings of the system. The management of the settings includes getting the new settings and saving the new settings and retrieve the settings.

Start Declare variables of the basic data; Get input from data management component; If (input == setting input)  Set values into the database; If (input == retrieving settings)  {  Access database;  Load data and return;  }

End

**4.1.2 COMPONENT 1 PROCESSING DETAIL**

**4.1.2.1 Design class**

****

**4.1.2.2 Restriction/limitations**

This component can only be called after accessing to data management.

**4.1.2.3 Performance issues**

* New settings have to be updated into the settings database all the time.
* Ensure the uniqueness of each separated setting.
* The settings in the database have to be uniform.

**4.1.2.4 Design constraints**

* Restrict recording mal-interactions that can take place. Mal-interactions could be the attempt to access to a setting that doesn’t exist.
* Reduce the chance of errors. Errors can be:
  + - * + New setting is not updated into DB
        + New setting is updated into the DB but the old setting is still there
* Can also work to focus user’s attention to needed task.

**4.1.2.5 Processing details   4.1.2.5.1 Processing narrative for each operation**

**4.1.2.5.1. a get\_setting()**

The responsibility of this component is to get the input from user’s activities and decide whether it’s an input for setting. Then parse the input into the declared variables.

**4.1.2.5.1 . b save\_setting()**

The responsibility of this component is save the input in the declared variables into the database.

**4.1.2.5.1 . c retrieve\_setting()**

The responsibility of this component is to load the settings from the DB to the caller.  **4.1.2.5.2 Algorithmic model for each operation  4.1.2.5.2. a get\_setting()**

get\_input() {

declare variables;

while (input) { get input; validate input; if (input == setting input)   put into declared variables; }

}

**4.1.2.5.2. b save\_setting()**

pass\_input() {

declare variables;

while (input) { retrieve data from pre-declared variables;  save settings to DB;}

}

**4.1.2.5.2. c retrieve\_setting()**

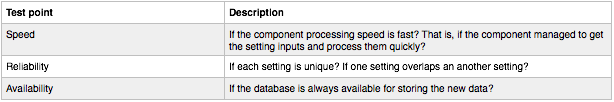
load\_input() {

declare variables;

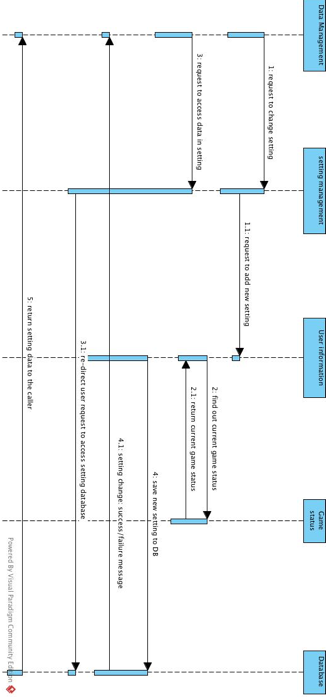
while (input) { get input; validate input; if (input == retrieve setting) {   access DB;  retrieve DB; }

}

**4.1.3 COMPONENT TEST POINTS LIST AND DESCRIPTION**



**4.1.4 DYNAMIC BEHAVIORS**

****

**4.1.5 COMPONENTS INTERFACE**

The component Setting management has the interface with the classes Data management. It takes request from data management and return the desired output to the data management. Setting management the employee inherits the attributes and functions from the class Data management. The Setting management has the responsibility of getting the new settings and/or receive request on setting management and manage the settings of the system.