Bugs, Issues and Concerns Reported by Soren Per Thygesen -- April 2018

These were originally reported by private emails to David Livingstone, and dealt with via email correspondence. For the record, they are reported under the Tickets/Bugs-DBMS part of the Sourceforge website.

1. **Bugs whose fixes were also provided by Soren Per Thygesen**

Bug	Status
The **RaquelTeachingTool** soft links need to be updated/created so that they point to the right libraries.	Fixed, via creation of bash script 'SoftlinkToTclTk.sh.
The "**-L ./**" linker path need to be added to the "**../RaquelGUI/Branches/AddMeta/Makefile.am**" in order for it to compile..	Fixed.
Compilation errors in the **InputStack.cpp** file. Wrong return values specified. *InputStack.cpp:352:12: warning: converting 'false' to pointer type 'CRaquelToken*' [-Wconversion-null] return false;* ^ *InputStack.cpp:366:12: warning: converting 'false' to pointer type 'CRaquelToken*' [-Wconversion-null] return false;* ^	Fixed.

2. **Issues raised by Soren Per Thygesen**

Issues	Comments
The **RaquelDBMS** Module handles an instance of itself, a **static** instance no less. Is this a proper implementation of the **Singleton** pattern found in Gamma et al.? Or is this a cheat to make it look OO, but really implementing it C-style?	The DBMS architecture calls for the DBMS to be implemented as a single 'procedure', formed as a stack of layers. This appears to be an OO-style approach to implementing it, as a single static instance to ensure just one DBMS 'object' permanently runs. In due course, re-factoring should replace the class by a procedure.
The place and name of the databases are hardcoded into the **RaquelDBMS** module. Perhaps an end user should be able to tell the system where it should store the data ?	This is entirely a limitation of the prototype, resulting from a lack of resources. The user should be able to create and destroy their own DBs, and store them where they like.

Concerns	Comments
The **sscanf** functions converting strings to various datatypes are not C++. They cause problems when trying to compile on Windows, and it's an ugly mix of C and C++ (without "extern 'C' {} blocks"). This is going to cause problems at some time in the future. The correct C++ way is using "std::stringstream".	It is agreed that it is preferable to use standard C++ facilities throughout. The location of the **sscanf** functions was not provided.
RaquelDBMS The '**make clean**' command is not working as expected. The library must be deleted manually.	Not checked.

3. **Other Concerns raised by Soren Per Thygesen**

Several other general concerns were also put forward. They were non-specific in that no code locations in the prototype were given. A few were misunderstandings of the code. After discussion, none of the comments were pursued or acted on.