315-DBMS-Project Report

Team Members

- Jaiden Gerig
- Nabir Dinani
- Eric Cochrane
- Tristan Le

Partially revised documentation for DBMS

Using our library

- Include DBMS.h

```
#include "DBMS.h"
```

- Create a DBMS object

```
DBMS db;
```

- use the execute function to issue queries and commands(it will return 1 if an error occurs)

```
db.execute("SHOW test;");
```

- use the get_table function to retrieve a specfic table(more on tables later)

```
Table t = db.get_table("test");
```

Query Language

Our query language is almost identical to the one presented in the problem state with only 1 caveat, All key words are capitalized(even ones relating to queries). you can find examples for every query or command statement in the included test.cpp. it is also important to know that for any sort of name, you cannot start it with a number or 2 consecutive Capital letters (I.E. TAblename, 1_table), execute will reject any names that ignore these rules, note that this does not apply to literals such as "TAblename" and "1_table" as they are wrapped in parentheses. Please feel free to try any commands in the included shell.cpp file

Using the Table class

A table contains a vector of Attributes(which have a name and a type) and a vector of Records(each of which hold vectors of strings), and a name . it can be visualized like so,

```
|Attribute  |Attribute  |Attribute  |
|record_0[0]|record_0[1]|record_0[2]|
|record_1[0]|record_1[1]|record_1[2]|
|record_i[0]|record_i[1]|record_i[2]|
```

Every Table has getters for both single and of of the attributes and records it contains which can be seen in Engine/Table.h, every record and attribute also contains it's own getters which can be found in Engine/Record.h and Engine/Attribute.h respectively and can be used to extract data from the Table object.

Note that you can only get Tables from the database, you cannot save one back into it and thus you should avoid modifying them directly.

Development Log

2/6/2015

- Finished Documentation

2/9/2015 - 2/15/15

- Began creating Engine
- finished Table class(Jaiden)
- finished Insert in Database class(Nabir)
- finished Get/Set for Record.h(Tristan)
- finished Open function out of command functions(Eric)
- finished create in Database(Tristan)
- finished write function in command function(Jaiden)
- finished show function in command functions(Tristan)
- finished on set_union in Query function(Tristan)
- finished delete_records in command functions(Nabir)
- finished set_difference (Jaiden)
- finished update in command functions(Nabir)
- Working on set_select() (Eric)
- finished close (Jaiden)
- Finished set_difference in Query functions(Jaiden)
- finished set_rename query function(Nabir)
- finished set_product in Query functions (Tristan)
- finished set_project (Jaiden)
- cleaned up files (Jaiden)
- Added comments (Nabir, Jaiden, Eric, Tristan)
- Finished Engine

2/16/2015 - 2/22/15

- began creating parser
- finished Token Class (Jaiden)
- finished query part of parser (Jaiden)
- finished condition part of parser (Tristan)
- finished command functions (Tristan)
- finished Tokenizer (Eric)
- finished integrating Tokenizer with parser (Jaiden)
- cleaned up files + commenting (Nabir, Jaiden, Eric, Tristan)
- Finished Parser

2/22/2015 - 2/27/15

- Began integrating parser
- Contacted team whose DBMS we are recieving(Eric,Nabir)
- Started working on DB Application(Eric,Nabir)
- finished integrating queries(Jaiden)
- finished integrating conditions(Jaiden)
- finished integrating commands(Tristan)
- Finished integrating Parser

3/4/2015 - 3/8/2015

- working on the interface (Nabir)
- Finished on make post (Nabir)
- Working on integrating the given DBMS (Eric)
- Made Weblog skeleton .h and .cpp (Eric)
- Working on Edit functions (Nabir)
- Working on post class (Nabir, Eric)
- Finished view/show (Nabir, Eric)
- Finished search funtions (Nabir, Eric)
- Working on comment on post/comment (Nabir, Eric)
- Working on delete a post (Nabir, Eric)
- Finished menus (Nabir, Eric)
- Formatting the output and the control flow(Nabir, Eric)
- Testing different cases (Nabir, Eric)
- cleaned up files + commenting (Nabir, Jaiden, Eric, Tristan)

Test Session logs

DBMS Tests

```cpp
#include "DBMS.h"
#include <iostream>

using namespace std;

int main(){
    DBMS db;
    db.execute("CREATE TABLE test (id INTEGER,name VARCHAR(20),description VARCHAR(50),user_id INTEGER) PRI

    for (int i = 0; i < 20; ++i){
        Record r;
        //c++ 11 features
        string pid = to_string(i);
        string uid = to_string(i*20);
        string q = "INSERT INTO test VALUES FROM ("+ pid+",\"Dummy String\",\"Dummy Text\","+uid+");";
        db.execute(q);
    }
    //command tests
    db.execute("SHOW test;");
```

```
test
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
0|Dummy String|Dummy Text|0|
1|Dummy String|Dummy Text|20|
2|Dummy String|Dummy Text|40|
3|Dummy String|Dummy Text|60|
4|Dummy String|Dummy Text|80|
5|Dummy String|Dummy Text|100|
6|Dummy String|Dummy Text|120|
7|Dummy String|Dummy Text|140|
8|Dummy String|Dummy Text|160|
9|Dummy String|Dummy Text|180|
10|Dummy String|Dummy Text|200|
11|Dummy String|Dummy Text|220|
12|Dummy String|Dummy Text|240|
13|Dummy String|Dummy Text|260|
14|Dummy String|Dummy Text|280|
15|Dummy String|Dummy Text|300|
16|Dummy String|Dummy Text|320|
17|Dummy String|Dummy Text|340|
18|Dummy String|Dummy Text|360|
19|Dummy String|Dummy Text|380|
```

```cpp
    db.execute("SHOW test;");
    db.execute("WRITE test;");
    db.execute("CLOSE test;");
    db.execute("OPEN test;");
    db.execute("SHOW test;");
```

```
test
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
0|Dummy String|Dummy Text|0|
1|Dummy String|Dummy Text|20|
2|Dummy String|Dummy Text|40|
3|Dummy String|Dummy Text|60|
4|Dummy String|Dummy Text|80|
5|Dummy String|Dummy Text|100|
6|Dummy String|Dummy Text|120|
7|Dummy String|Dummy Text|140|
8|Dummy String|Dummy Text|160|
9|Dummy String|Dummy Text|180|
10|Dummy String|Dummy Text|200|
11|Dummy String|Dummy Text|220|
12|Dummy String|Dummy Text|240|
13|Dummy String|Dummy Text|260|
14|Dummy String|Dummy Text|280|
15|Dummy String|Dummy Text|300|
16|Dummy String|Dummy Text|320|
17|Dummy String|Dummy Text|340|
18|Dummy String|Dummy Text|360|
19|Dummy String|Dummy Text|380|
```

```
    db.execute("UPDATE test SET name = \"jaiden\" WHERE (name == \"Dummy String\"); ");
    db.execute("SHOW test;");
    db.execute("INSERT INTO test VALUES FROM (25,\"Eric\",\"Tristan\",3);" );
    db.execute("SHOW test;");
    db.execute("DELETE FROM test WHERE (name == \"jaiden\");" );
    db.execute("SHOW test;");
```

```
test
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
0|jaiden|Dummy Text|0|
1|jaiden|Dummy Text|20|
2|jaiden|Dummy Text|40|
3|jaiden|Dummy Text|60|
4|jaiden|Dummy Text|80|
5|jaiden|Dummy Text|100|
6|jaiden|Dummy Text|120|
7|jaiden|Dummy Text|140|
8|jaiden|Dummy Text|160|
9|jaiden|Dummy Text|180|
10|jaiden|Dummy Text|200|
11|jaiden|Dummy Text|220|
12|jaiden|Dummy Text|240|
13|jaiden|Dummy Text|260|
14|jaiden|Dummy Text|280|
15|jaiden|Dummy Text|300|
16|jaiden|Dummy Text|320|
17|jaiden|Dummy Text|340|
18|jaiden|Dummy Text|360|
19|jaiden|Dummy Text|380|
test
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
0|jaiden|Dummy Text|0|
1|jaiden|Dummy Text|20|
2|jaiden|Dummy Text|40|
3|jaiden|Dummy Text|60|
4|jaiden|Dummy Text|80|
5|jaiden|Dummy Text|100|
6|jaiden|Dummy Text|120|
7|jaiden|Dummy Text|140|
8|jaiden|Dummy Text|160|
9|jaiden|Dummy Text|180|
10|jaiden|Dummy Text|200|
11|jaiden|Dummy Text|220|
12|jaiden|Dummy Text|240|
13|jaiden|Dummy Text|260|
14|jaiden|Dummy Text|280|
15|jaiden|Dummy Text|300|
16|jaiden|Dummy Text|320|
17|jaiden|Dummy Text|340|
18|jaiden|Dummy Text|360|
19|jaiden|Dummy Text|380|
25|Eric|Tristan|3|
test
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
25|Eric|Tristan|3|
```

```
string q = "testing <- SELECT (id < 5 && (name == \"Eric\" || name != \"Tristan\")) test;";
db.execute(q);
db.execute("SHOW testing;");

q = "testing <- test - testing;";
db.execute(q);
db.execute("SHOW testing;");

q = "testing <- test + testing;";
db.execute(q);
db.execute("SHOW testing;");

q = "testing <- RENAME (new_id,new_name,new_descr,new_user_id) testing;";
db.execute(q);
db.execute("SHOW testing;");

q = "testing <- PROJECT (new_id,new_name) testing;";
db.execute(q);
db.execute("SHOW testing;");

q = "testing <- SELECT (id == new_id) (test * testing);";
db.execute(q);
db.execute("SHOW testing;");
```

```
testing
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
0|Dummy String|Dummy Text|0|
1|Dummy String|Dummy Text|20|
2|Dummy String|Dummy Text|40|
3|Dummy String|Dummy Text|60|
4|Dummy String|Dummy Text|80|
testing
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
5|Dummy String|Dummy Text|100|
6|Dummy String|Dummy Text|120|
7|Dummy String|Dummy Text|140|
8|Dummy String|Dummy Text|160|
9|Dummy String|Dummy Text|180|
10|Dummy String|Dummy Text|200|
11|Dummy String|Dummy Text|220|
12|Dummy String|Dummy Text|240|
13|Dummy String|Dummy Text|260|
14|Dummy String|Dummy Text|280|
15|Dummy String|Dummy Text|300|
16|Dummy String|Dummy Text|320|
17|Dummy String|Dummy Text|340|
18|Dummy String|Dummy Text|360|
19|Dummy String|Dummy Text|380|
testing
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|
0|Dummy String|Dummy Text|0|
1|Dummy String|Dummy Text|20|
2|Dummy String|Dummy Text|40|
3|Dummy String|Dummy Text|60|
4|Dummy String|Dummy Text|80|
5|Dummy String|Dummy Text|100|
6|Dummy String|Dummy Text|120|
7|Dummy String|Dummy Text|140|
8|Dummy String|Dummy Text|160|
9|Dummy String|Dummy Text|180|
10|Dummy String|Dummy Text|200|
11|Dummy String|Dummy Text|220|
12|Dummy String|Dummy Text|240|
13|Dummy String|Dummy Text|260|
14|Dummy String|Dummy Text|280|
15|Dummy String|Dummy Text|300|
16|Dummy String|Dummy Text|320|
17|Dummy String|Dummy Text|340|
18|Dummy String|Dummy Text|360|
19|Dummy String|Dummy Text|380|
```

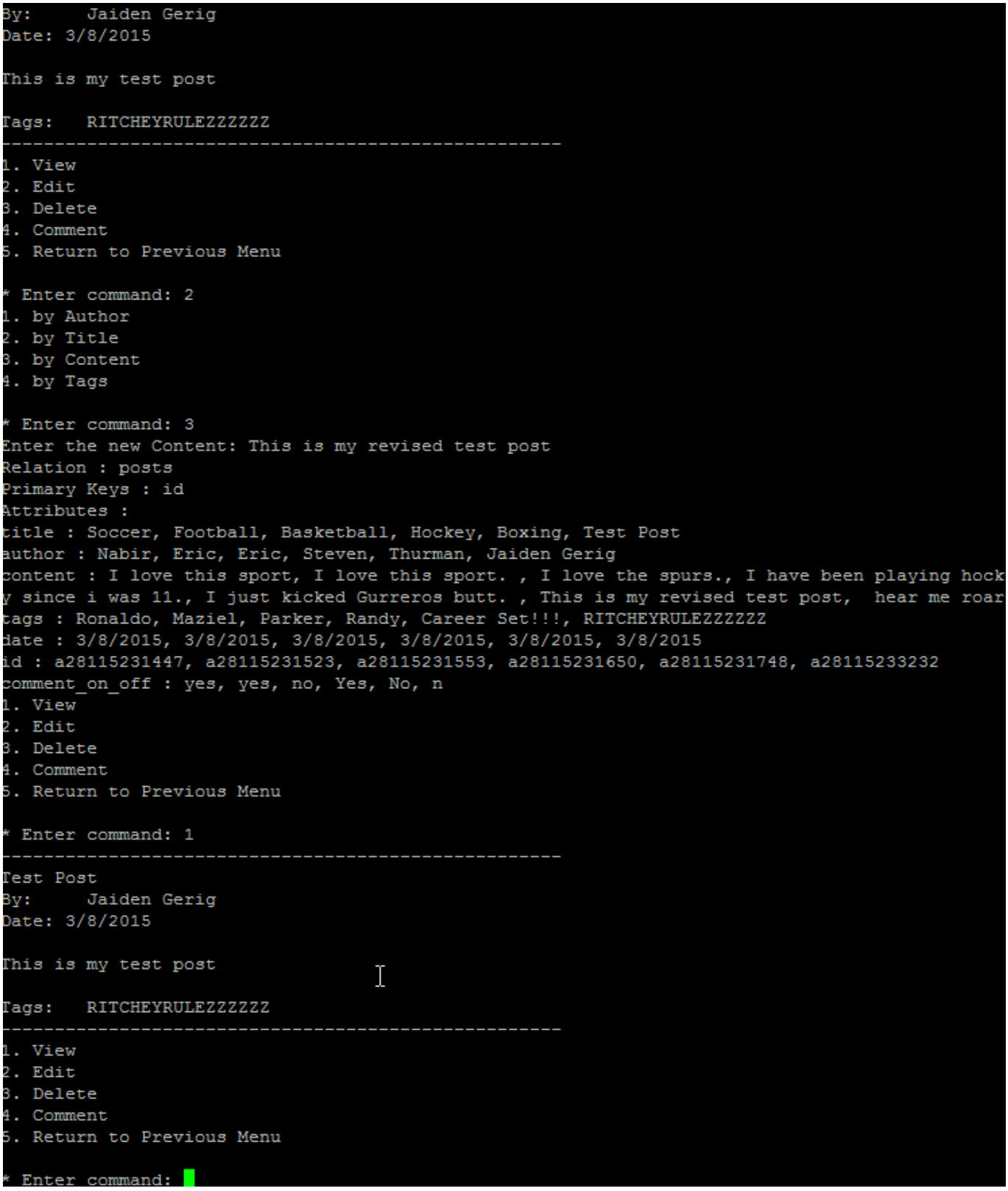
```
testing
new_id(INTEGER)|new_name(VARCHAR(20))|new_descr(VARCHAR(50))|new_user_id(INTEGER)|
0|Dummy String|Dummy Text|0|
1|Dummy String|Dummy Text|20|
2|Dummy String|Dummy Text|40|
3|Dummy String|Dummy Text|60|
4|Dummy String|Dummy Text|80|
5|Dummy String|Dummy Text|100|
6|Dummy String|Dummy Text|120|
7|Dummy String|Dummy Text|140|
8|Dummy String|Dummy Text|160|
9|Dummy String|Dummy Text|180|
10|Dummy String|Dummy Text|200|
11|Dummy String|Dummy Text|220|
12|Dummy String|Dummy Text|240|
13|Dummy String|Dummy Text|260|
14|Dummy String|Dummy Text|280|
15|Dummy String|Dummy Text|300|
16|Dummy String|Dummy Text|320|
17|Dummy String|Dummy Text|340|
18|Dummy String|Dummy Text|360|
19|Dummy String|Dummy Text|380|
testing
new_id(INTEGER)|new_name(VARCHAR(20))|
0|Dummy String|
1|Dummy String|
2|Dummy String|
3|Dummy String|
4|Dummy String|
5|Dummy String|
6|Dummy String|
7|Dummy String|
8|Dummy String|
9|Dummy String|
10|Dummy String|
11|Dummy String|
12|Dummy String|
13|Dummy String|
14|Dummy String|
15|Dummy String|
16|Dummy String|
17|Dummy String|
18|Dummy String|
19|Dummy String|
testing
id(INTEGER)|name(VARCHAR(20))|description(VARCHAR(50))|user_id(INTEGER)|new_id(INTEGER)|new_name(VARCHAR(20))|
0|Dummy String|Dummy Text|0|0|Dummy String|
1|Dummy String|Dummy Text|20|1|Dummy String|
2|Dummy String|Dummy Text|40|2|Dummy String|
3|Dummy String|Dummy Text|60|3|Dummy String|
4|Dummy String|Dummy Text|80|4|Dummy String|
5|Dummy String|Dummy Text|100|5|Dummy String|
6|Dummy String|Dummy Text|120|6|Dummy String|
7|Dummy String|Dummy Text|140|7|Dummy String|
8|Dummy String|Dummy Text|160|8|Dummy String|
9|Dummy String|Dummy Text|180|9|Dummy String|
10|Dummy String|Dummy Text|200|10|Dummy String|
11|Dummy String|Dummy Text|220|11|Dummy String|
12|Dummy String|Dummy Text|240|12|Dummy String|
13|Dummy String|Dummy Text|260|13|Dummy String|
14|Dummy String|Dummy Text|280|14|Dummy String|
15|Dummy String|Dummy Text|300|15|Dummy String|
16|Dummy String|Dummy Text|320|16|Dummy String|
17|Dummy String|Dummy Text|340|17|Dummy String|
18|Dummy String|Dummy Text|360|18|Dummy String|
19|Dummy String|Dummy Text|380|19|Dummy String|
```

Application Tests

```
Search by:
1. Author
2. Title
3. Tag(s)
4. Date
5. Return to Main Menu

* Enter command: 2
Enter the Title: Test Post
0: Test Post
Jaiden Gerig
1: Test Post
Jaiden Gerig
Enter the Record number: 1
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command: 1
--------------------------------------------------
Test Post
By:     Jaiden Gerig
Date: 3/8/2015

This is my test post, hear me roar!

Tags:   RITCHEYRULEZZZZZZ
--------------------------------------------------
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command: 5

[Search Menu]

Search by:
1. Author
2. Title
3. Tag(s)
4. Date
5. Return to Main Menu

* Enter command: 3
Enter the Tags: RITCHEYRULEZZZZZZ
0: Test Post
3/8/2015
1: Test Post
3/8/2015
2: Test Post
3/8/2015
Enter the Record number: 1
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command: 1
--------------------------------------------------
Test Post
By:     Jaiden Gerig
Date: 3/8/2015

This is my test post, hear me roar!

Tags:   RITCHEYRULEZZZZZZ
--------------------------------------------------
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command:
```

```
[Main Menu]

1. Make a new post
2. Search a post
3. Exit

* Enter command: 2

[Search Menu]

Search by:
1. Author
2. Title
3. Tag(s)
4. Date
5. Return to Main Menu

* Enter command: 2
Enter the Title: Test Post
0: Test Post
Jaiden Gerig
Enter the Record number: 0
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command: 3
Error in removeRelationFromDisk() :: a28115234734 is not on the Disk.
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command: 5

[Search Menu]

Search by:
1. Author
2. Title
3. Tag(s)
4. Date
5. Return to Main Menu

* Enter command: 4
Enter the Date(mm/dd/yyyy): 03/08/2015
No files found.

[Search Menu]

Search by:
1. Author
2. Title
3. Tag(s)
4. Date
5. Return to Main Menu

* Enter command:
```

```
[jaideng]@linux2 ~/DBMS/DB_APP> (23:31:40 03/08/15)
:: ./a.out
[Main Menu]

1. Make a new post
2. Search a post
3. Exit

* Enter command: 1

Title: Test Post

Author: Jaiden Gerig

Content: This is my test post, hear me roar!

Tags: RITCHEYRULEZZZZZZ

Commenting(Yes/no)? n
[Main Menu]

1. Make a new post
2. Search a post
3. Exit

* Enter command: 2

[Search Menu]

Search by:
1. Author
2. Title
3. Tag(s)
4. Date
5. Return to Main Menu

* Enter command: 1
Enter the author: Jaiden Gerig
0: Test Post
3/8/2015
Enter the Record number: 0
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command:
1
--------------------------------------------------
Test Post
By:     Jaiden Gerig
Date: 3/8/2015

This is my test post, hear me roar!

Tags:   RITCHEYRULEZZZZZZ
--------------------------------------------------
1. View
2. Edit
3. Delete
4. Comment
5. Return to Previous Menu

* Enter command:
```



Post Prodution Notes

DBMS

Our model of Tables consisting of Records and Attributes held up very well, and was very easy to implement and work with. For the parsing part of our DBMS, we originally thought of using a boost library named Spirit, but as we looked at it more we found the code required to write for it to be

too complex and the tool itself seemed to be a bit too sophisticated for our project, so we scrapped the idea and made a parser from scratch using a model where a string is turned into a vector of tokens via a tokenizer(lexer), then this vector of tokens is taken by the actual parser which performs the actions that it needs to and then exits. The trickiest part of this implementation was accommodating nested conditional and query statements, as it required a great a deal of code to integrate it. The parser itself proved to be an unwieldy piece of code as it eventually grew to about 800 lines making it the largest, most complex file in our DBMS. But the team that used our DBMS had very few problems with it so we would call it a success overall

DB Application

Our initial design for our Database application held up well, however the same cannot be said for the DBMS system we received. While not completely broken, There were quite a few bugs that we encountered. We tried to get the Team that gave it to us to fix them, but in the end we had to use a number of workarounds to get it to function properly. This led to some less than stellar code, but as a whole the Application works as intended and we had very few problems with our design.

Work load distribution

- Jaiden - 25 (DBMS Engine, Query + Condition Parser)
- Eric - 25 (DBMS Engine functions, Tokenizer, Application)
- Tristan - 25(DBMS Engine functions, Command Parser, Application)
- Nabir - 25(DBMS Engine functions, Application)