

Programming Assignment – 3

Simple XML DBMS



First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotitaw	28

PersonID	Address	Address	City	City	First Name	First Name	Last
1	1040 East Street	1040 East Street	Plateau City	Plateau City	Loretta	Loretta	Boy
2	154 Baltic Walk	154 Baltic Walk	Excelsior	Excelsior	Evelyn	Evelyn	Ellis
3	952 Tennessee Avenue	952 Tennessee Avenue	Embarcadero	Embarcadero	Harold	Harold	Mcc
4	780 Fourth Lane	780 Fourth Lane	Tenderloin	Tenderloin	Chad	Chad	Hart
5	1079 Beach Way	1079 Beach Way	Cow Hollow	Cow Hollow	Sandra	Sandra	San
6	758 North Lane	758 North Lane	North Beach	North Beach	Kathleen	Kathleen	Rho
7	978 Eighth Walk	722 Arrow Lane	Minutemen Park	Columbus	Pamela	Pamela	Nea
8	247 Fifth Place	247 Fifth Place	Western Addition	Western Addition	Emily	Emily	Dev
9	843 States Street	843 States Street	Noe Valley	Noe Valley	Vernon	Vernon	Carl
10	749 Washington Street	749 Washington Street	Civic Center	Civic Center	William	William	Gor
11	360 Tennessee Place	360 Tennessee Place	Fisherman's Wharf	Fisherman's Wharf	Gladys	Gladys	Lev
12	14 Oriental Place	14 Oriental Place	Buena Vista	Buena Vista	Margaret	Margaret	Cok
13	868 Lower Avenue	868 Lower Avenue	Diamond Heights	Diamond Heights	Kathleen	Kathleen	Guz
14	896 Third Street	896 Third Street	Civic Center	Civic Center	Bertha	Bertha	Pov
15	229 Kentucky Place	229 Kentucky Place	Ocean View	Ocean View	Kim	Kim	Org
16	1019 Marvin Gardens Place	1019 Marvin Gardens Place	Palmer Hill	Palmer Hill	Steve	Steve	Cun

Prepared by:

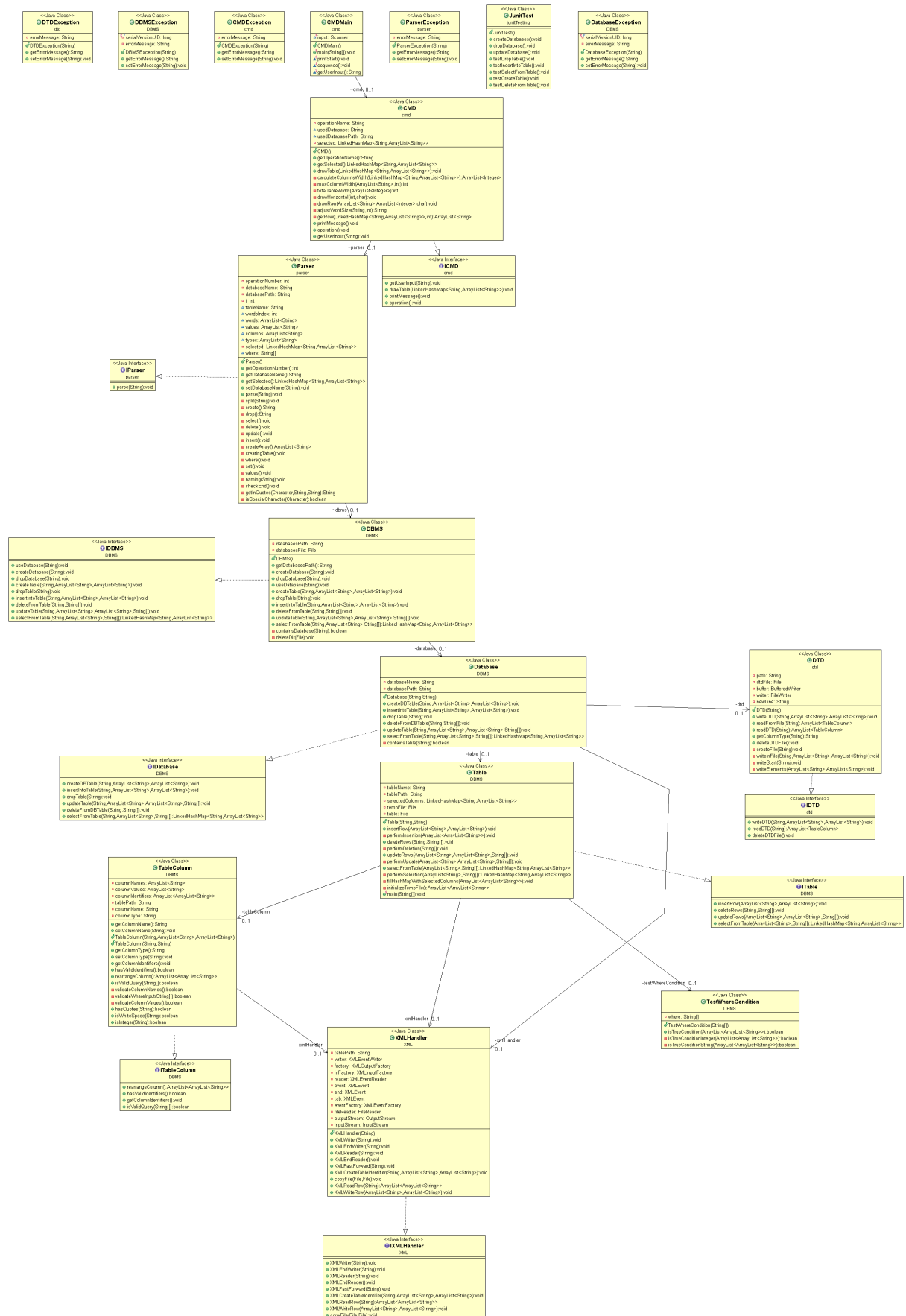
Aya Sameh Esmail (1)

Bassent Mostafa Saad Zaghloul (21)

Salma Hesham Mohammed Ragab (34)

Mira Samir Ragheb Tawfiles (78)

- **UML Diagram:**



- **User Guide:**

- The user should first select the required database through the “USE” command.
- The user should choose this required database from the Databases_ folder which is created on his drive on running the program.
- New database folder is automatically created in the Databases_ folder on using the “CREATE DATABASE” command.
- Naming is important!
A database or a table name cannot start with a number or a special character except ‘_’.
- Datatypes are only “varchar” and “int”.
- The columns size is supposed to be constant, so the user is not required to enter a size.
- On creating a table, an XML and a DTD files are automatically created in the database source folder with the name of the table.
- A user can mess up his XML file, but on reading it again it won’t match the DTD file, so the table won’t load!

Queries used:

Use: This query sets the database that the user wants to access.

```
>> USE DATABASE databaseName;
```

Create database: This query creates a new database.

```
>> CREATE DATABASE databaseName;
```

Drop database: This query deletes the database chosen.

```
>> DROP DATABASE databaseName;
```

Create table: This query creates a table in the used database.

```
>> CREATE TABLE tableName  
(columnName1 data_type,  
columnName2 data_type,  
columnName3 data_type  
);
```

Insert into table: This query inserts some values in the chosen table.

```
>> INSERT INTO tableName  
VALUES (value1,value2,value3,...);  
  
>> INSERT INTO tableName  
(column1,column2,column3,...)  
VALUES (value1,value2,value3,...);
```

Select from table: This query selects some values from the chosen table.

```
>> SELECT columnName,columnName  
FROM tableName;  
  
>> SELECT * FROM tableName;
```

Delete from table: This query deletes a row from the chosen table.

```
>> DELETE FROM tableName  
WHERE columnName = someValue;
```

Update table: This query updates some values in the chosen table.

```
>> UPDATE tableName  
SET column1=value1,column2=value2,...  
  
WHERE some_column=some_value;
```

Drop table: This query deletes the table chosen.

```
>> DROP TABLE tableName;
```

- **Design decisions and Assumptions:**

- This project follows the MVC design pattern.
- The CMD class is considered the handler of the main class and the Parser class.
- The Parser sends the query and the instructions to the DBMS which then do the required task.
- In the XML parsers, DOM is not efficient for large files; because DOM saves the file before reading it, then it starts reading so it consumes a lot of memory and is considered slow. SAX and StAX parsers, both have no memory, so less memory consumption, so they are faster than DOM parser.

StAX can write into the file and read from it while SAX only reads from files, so if SAX is used, DOM parser has to be used to write into the file.

SAX has a DTD validator, so on reading validation can be done, while StAX has no such function.

SAX cannot be stopped, while the StAX can stop reading at a specific line or tag through its handler.

From all these points, we used StAX parser and a DTD validator tool.