Name :- Rushikesh Karbhari Palve
Roll No. 31258

Date : / /20 Page No.

Assignment No. B4

DOP :- 24-11-2021 DOS :- 30-11-2021

Title :- Database Connectivity.

Problem Statement :-

Write a program to implement MongoDB database connectivity with any frontend language to implement database navigation operations (add, delete, edit, etc).

Objectives :-

(i) To learn about database connectivity.
(ii) To implement MongoDB database connectivity with basic operations.

Learning Outcomes :-

After completion of the assignment, students will be able to -
(i) Implement MongoDB database connectivity with basic operations.

THEORY :-

One can establish connectivity between Java

and MongoDB.
The steps to perform this are very easy to follow.

Prerequisites -

(i) Install the required JAR file to establish connectivity.
(ii) Import the required libraries in the Java program.

MongoClient :-

It helps to access the database to be used. You can create a database or use the existing one.

getCollection() :-

You can use a collection in the given database or create a new collection.
You can perform various operations on the collection.

Operations on the collection :-

(i) Insertion -

In order to insert a document into the collection, you have to create a database object and append the fields to the document.
This document is then inserted into the collection using the insert() method to do same.

Eg -

```
Document doc = new Document("field1", value1)
                         .append("field2", Dalue2)
                           :
                         .append("fieldn", valuen)

collection.insertOne(doc);
```

(ii) Searching :-

In order to look for a record you have to create a document with the required value to be matched -

Suppose you want to search for someone with a roll no - 1

```
Document doc = new Document("rollNo", 1)

collection.findOne(doc);
```

the result will have the document which is returned by the findOne operation -

(iii) Updation :-

Create a document with the updated values of the old document -

Create a query to look for the document to be updated and use the update() method to perform the updation -

collection.updateOne(searchDoc, updateDoc);

ⓘⓥ Deletion :-

The remove() method can be used to delete a document from a collection.

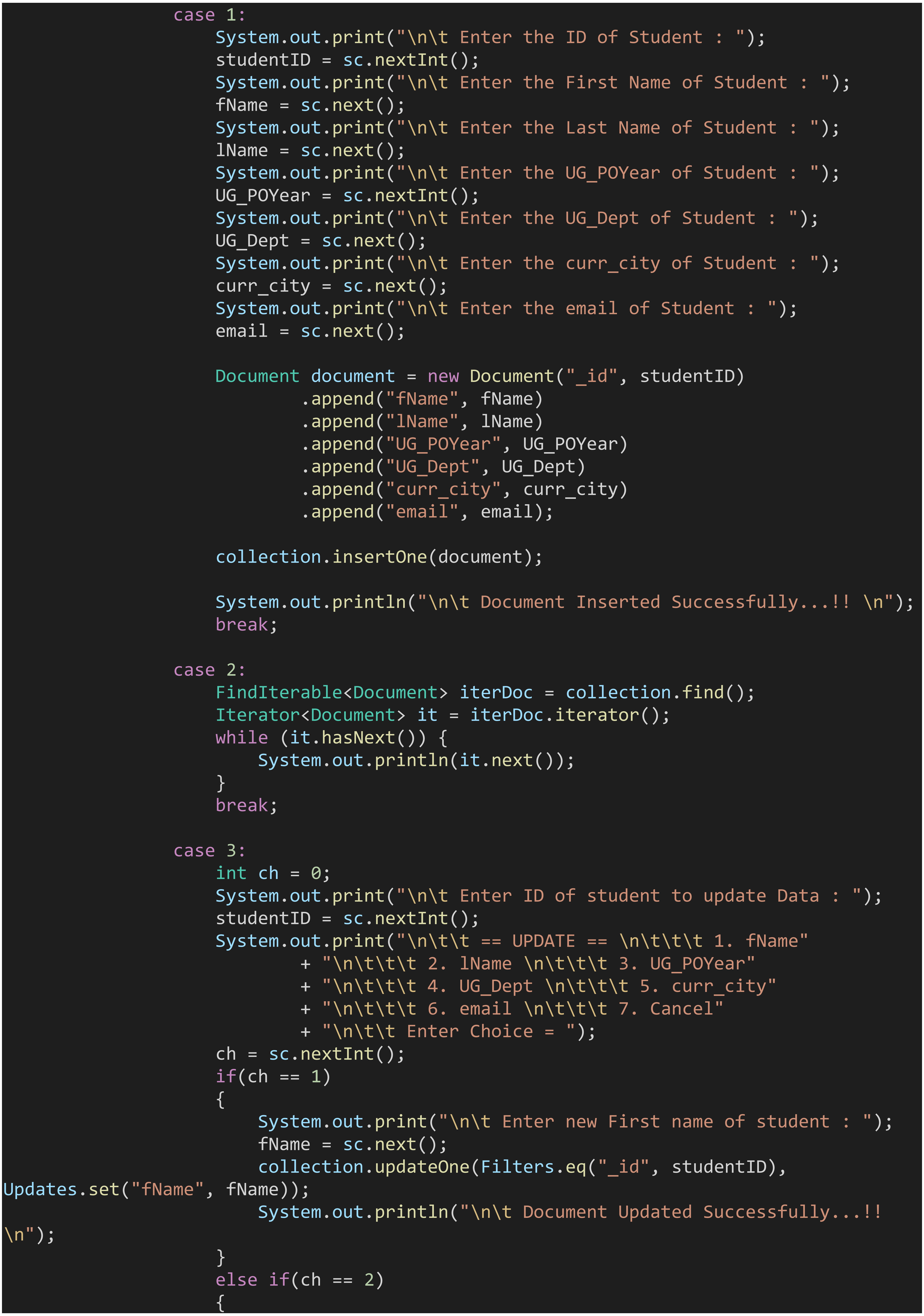
collection.remove(doc);

CONCLUSION :-

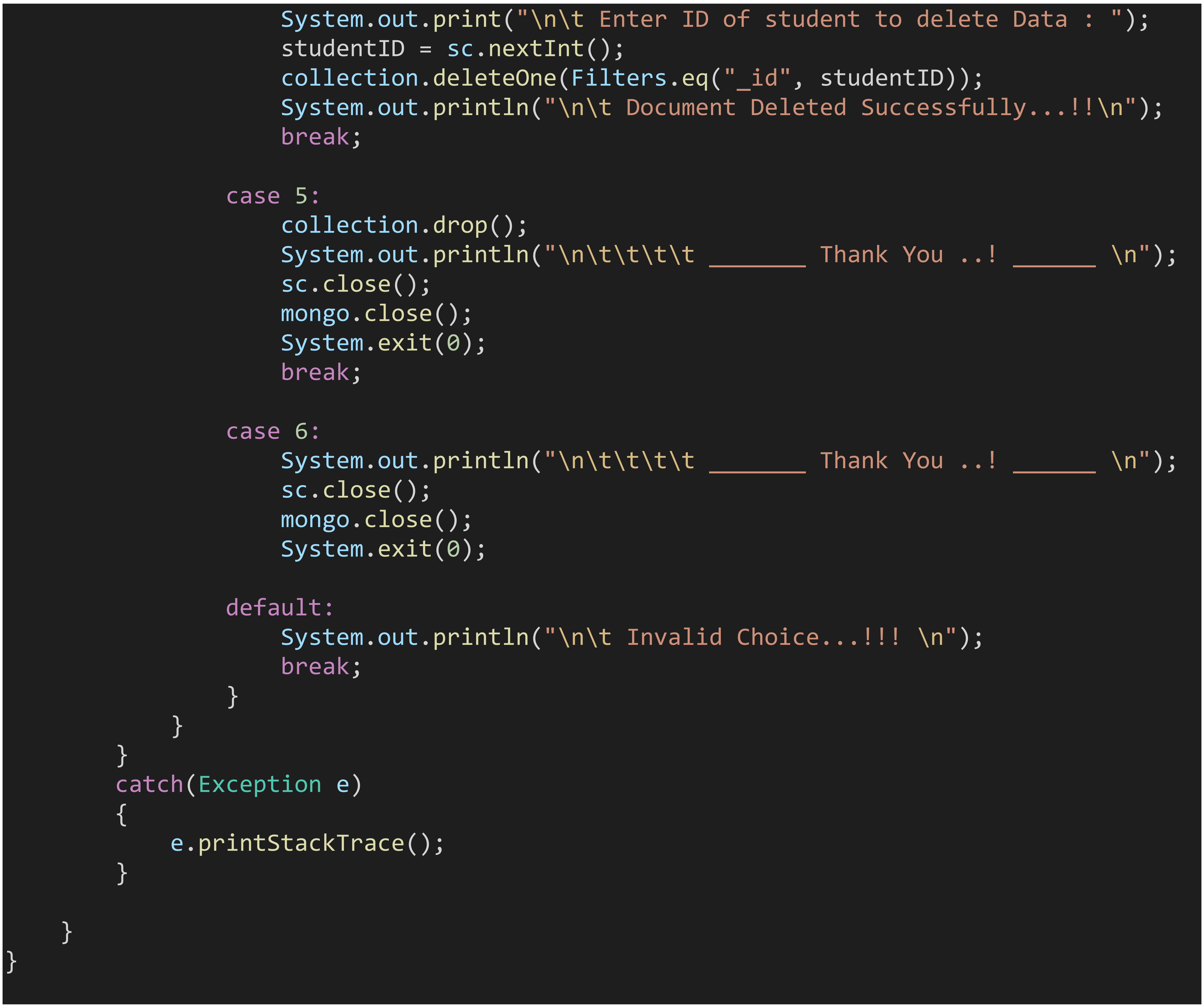
In this assignment, I understood how to connect MongoDB database with a java application and thus implemented a Java program to perform basic CRUD operations.

CODE :-

```
/*
 * Problem Statement :-
 *  Database Connectivity:
 *      Write a program to implement Mongo DB database connectivity with any front end
 *      language to implement Database navigation operations(add, delete, edit etc.)
*/


package assignmentNo_B4;

import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;
import com.mongodb.client.model.Updates;
import java.util.Scanner;
import java.util.Iterator;
import org.bson.Document;
import com.mongodb.MongoClient;

public class AssignmentNo_B4
{
    public static void main( String args[] )
    {
        try
        {
            Scanner sc = new Scanner(System.in);

            int studentID = 0, UG_POYear = 0;
            String fName = "", lName = "", UG_Dept = "", curr_city = "", email = "";

            MongoClient mongo = new MongoClient( "localhost" , 27017 );

            System.out.println("\n\t Connected to the database successfully...!!");

            MongoDatabase database = mongo.getDatabase("assignment_no_12");

            MongoCollection<Document> collection = database.getCollection("Student");

            System.out.println("\n\t Collection Student selected successfully...!!");

            int choice;
            while(true)
            {
                System.out.println("\n\t === Main Menu === \n\t\t 1. Insert Data"
                                        + "\n\t\t 2. Display Data"
                                        + "\n\t\t 3. Update Document"
                                        + "\n\t\t 4. Delete Document"
                                        + "\n\t\t 5. Drop Collection"
                                        + "\n\t\t 6. Exit");

                System.out.print("\n\t Enter Choice : ");
                choice = sc.nextInt();

                switch(choice)
                {
```



```
                case 1:
                    System.out.print("\n\t Enter the ID of Student : ");
                    studentID = sc.nextInt();
                    System.out.print("\n\t Enter the First Name of Student : ");
                    fName = sc.next();
                    System.out.print("\n\t Enter the Last Name of Student : ");
                    lName = sc.next();
                    System.out.print("\n\t Enter the UG_POYear of Student : ");
                    UG_POYear = sc.nextInt();
                    System.out.print("\n\t Enter the UG_Dept of Student : ");
                    UG_Dept = sc.next();
                    System.out.print("\n\t Enter the curr_city of Student : ");
                    curr_city = sc.next();
                    System.out.print("\n\t Enter the email of Student : ");
                    email = sc.next();

                    Document document = new Document("_id", studentID)
                            .append("fName", fName)
                            .append("lName", lName)
                            .append("UG_POYear", UG_POYear)
                            .append("UG_Dept", UG_Dept)
                            .append("curr_city", curr_city)
                            .append("email", email);

                    collection.insertOne(document);

                    System.out.println("\n\t Document Inserted Successfully...!! \n");
                    break;

                case 2:
                    FindIterable<Document> iterDoc = collection.find();
                    Iterator<Document> it = iterDoc.iterator();
                    while (it.hasNext()) {
                        System.out.println(it.next());
                    }
                    break;

                case 3:
                    int ch = 0;
                    System.out.print("\n\t Enter ID of student to update Data : ");
                    studentID = sc.nextInt();
                    System.out.print("\n\t\t == UPDATE == \n\t\t\t 1. fName"
                            + "\n\t\t\t 2. lName \n\t\t\t 3. UG_POYear"
                            + "\n\t\t\t 4. UG_Dept \n\t\t\t 5. curr_city"
                            + "\n\t\t\t 6. email \n\t\t\t 7. Cancel"
                            + "\n\t\t Enter Choice = ");
                    ch = sc.nextInt();
                    if(ch == 1)
                    {
                        System.out.print("\n\t Enter new First name of student : ");
                        fName = sc.next();
                        collection.updateOne(Filters.eq("_id", studentID),
Updates.set("fName", fName));
                        System.out.println("\n\t Document Updated Successfully...!!
\n");
                    }
                    else if(ch == 2)
                    {
                        System.out.print("\n\t Enter new Last name of student : ");
                        lName = sc.next();
                        collection.updateOne(Filters.eq("_id", studentID),
Updates.set("lName", lName));
                        System.out.println("\n\t Document Updated Successfully...!!
\n");
                    }
                    else if(ch == 3)
                    {
                        System.out.print("\n\t Enter new UG PassOut Year of student :
");
                        UG_POYear = sc.nextInt();
                        collection.updateOne(Filters.eq("_id", studentID),
Updates.set("UG_POYear", UG_POYear));
                        System.out.println("\n\t Document Updated Successfully...!!
\n");
                    }
                    else if(ch == 4)
                    {
                        System.out.print("\n\t Enter new UG Department of student :
");
                        UG_Dept = sc.next();
                        collection.updateOne(Filters.eq("_id", studentID),
Updates.set("UG_Dept", UG_Dept));
                        System.out.println("\n\t Document Updated Successfully...!!
\n");
                    }
                    else if(ch == 5)
                    {
                        System.out.print("\n\t Enter new Current City of student : ");
                        curr_city = sc.next();
                        collection.updateOne(Filters.eq("_id", studentID),
Updates.set("curr_city", curr_city));
                        System.out.println("\n\t Document Updated Successfully...!!
\n");
                    }
                    else if(ch == 6)
                    {
                        System.out.print("\n\t Enter new email of student : ");
                        email = sc.next();
                        collection.updateOne(Filters.eq("_id", studentID),
Updates.set("email", email));
                        System.out.println("\n\t Document Updated Successfully...!!");
                    }
                    else if(ch == 7)
                    {
                        System.out.println("\n\t Operation Cancelled
Successfully...!!");
                        break;
                    }
                    else
                    {
                        System.out.println("\n\t Invalid Choice...!!! \n");
                        break;
                    }
                    break;

                case 4:
```



```
                    System.out.print("\n\t Enter ID of student to delete Data : ");
                    studentID = sc.nextInt();
                    collection.deleteOne(Filters.eq("_id", studentID));
                    System.out.println("\n\t Document Deleted Successfully...!!\n");
                    break;

                case 5:
                    collection.drop();
                    System.out.println("\n\t\t\t\t _______ Thank You ..! ______ \n");
                    sc.close();
                    mongo.close();
                    System.exit(0);
                    break;

                case 6:
                    System.out.println("\n\t\t\t\t _______ Thank You ..! ______ \n");
                    sc.close();
                    mongo.close();
                    System.exit(0);

                default:
                    System.out.println("\n\t Invalid Choice...!!! \n");
                    break;
                }
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }

    }
}
```

OUTPUT :-

```
Connected to the database successfully...!!

Collection Student selected successfully...!!

=== Main Menu ===
   1. Insert Data
   2. Display Data
   3. Update Document
   4. Delete Document
   5. Drop Collection
   6. Exit

Enter Choice : 1

Enter the ID of Student : 1

Enter the First Name of Student : Rushi

Enter the Last Name of Student : Palve

Enter the UG_POYear of Student : 2023

Enter the UG_Dept of Student : Computer

Enter the curr_city of Student : Pune

Enter the email of Student : rushikeshkpalve@gmail.com

Document Inserted Successfully...!!


=== Main Menu ===
   1. Insert Data
   2. Display Data
   3. Update Document
   4. Delete Document
   5. Drop Collection
   6. Exit

Enter Choice : 1

Enter the ID of Student : 2

Enter the First Name of Student : Mayur

Enter the Last Name of Student : Mote

Enter the UG_POYear of Student : 2023

Enter the UG_Dept of Student : Computer

Enter the curr_city of Student : Pune

Enter the email of Student : mayurmote@gmail.com
```

```
    Document Inserted Successfully...!!


    === Main Menu ===
       1. Insert Data
       2. Display Data
       3. Update Document
       4. Delete Document
       5. Drop Collection
       6. Exit

    Enter Choice : 2
Document{{_id=1, fName=Rushi, lName=Palve, UG_POYear=2023, UG_Dept=Computer,
curr_city=Pune, email=rushikeshkpalve@gmail.com}}
Document{{_id=2, fName=Mayur, lName=Mote, UG_POYear=2023, UG_Dept=Computer,
curr_city=Pune, email=mayurmote@gmail.com}}

    === Main Menu ===
       1. Insert Data
       2. Display Data
       3. Update Document
       4. Delete Document
       5. Drop Collection
       6. Exit

    Enter Choice : 3

    Enter ID of student to update Data : 1

       == UPDATE ==
          1. fName
          2. lName
          3. UG_POYear
          4. UG_Dept
          5. curr_city
          6. email
          7. Cancel
       Enter Choice = 1

    Enter new First name of student : Rushikesh

    Document Updated Successfully...!!


    === Main Menu ===
       1. Insert Data
       2. Display Data
       3. Update Document
       4. Delete Document
       5. Drop Collection
       6. Exit

    Enter Choice : 2
Document{{_id=1, fName=Rushikesh, lName=Palve, UG_POYear=2023, UG_Dept=Computer,
curr_city=Pune, email=rushikeshkpalve@gmail.com}}
Document{{_id=2, fName=Mayur, lName=Mote, UG_POYear=2023, UG_Dept=Computer,
curr_city=Pune, email=mayurmote@gmail.com}}
```

```
    === Main Menu ===
       1. Insert Data
       2. Display Data
       3. Update Document
       4. Delete Document
       5. Drop Collection
       6. Exit

    Enter Choice : 4

    Enter ID of student to delete Data : 1

    Document Deleted Successfully...!!


    === Main Menu ===
       1. Insert Data
       2. Display Data
       3. Update Document
       4. Delete Document
       5. Drop Collection
       6. Exit

    Enter Choice : 2
Document{{_id=2, fName=Mayur, lName=Mote, UG_POYear=2023, UG_Dept=Computer,
curr_city=Pune, email=mayurmote@gmail.com}}

    === Main Menu ===
       1. Insert Data
       2. Display Data
       3. Update Document
       4. Delete Document
       5. Drop Collection
       6. Exit

    Enter Choice : 5

             _______ Thank You ..! ______
```