TITLE: Admission Entrance Test Automation System

Object Oriented Programming II [MCA 4221]

(FISAC Take Home Project)

By

Nihal Kiran Shetty

230970065

MCA

2nd SEMESTER

Section – A

Overview:

An educational institution aims to streamline its admission entrance test process by automating the creation of Multiple-Choice Questions (MCQs) question papers from a database of questions. Each question paper should contain a minimum of five (which can be user-defined) MCQs, with each MCQ having four options. Students are permitted to choose only one option per question. Upon submission, the system will automatically evaluate the answers, display the student's score on-screen, and store it in the database.

Create a GUI application in Java for conducting an admission entrance test consisting of MultipleChoice Questions (MCQs) with the following requirements:

1. Login Window:

- **The application should open a login window for user authentication.**
- **Users need to enter their credentials (username and password) for validation.**
- **After successful login, the application should navigate to a new page.**
- **This new page should contain a set of multiple-choice questions (MCQs).**

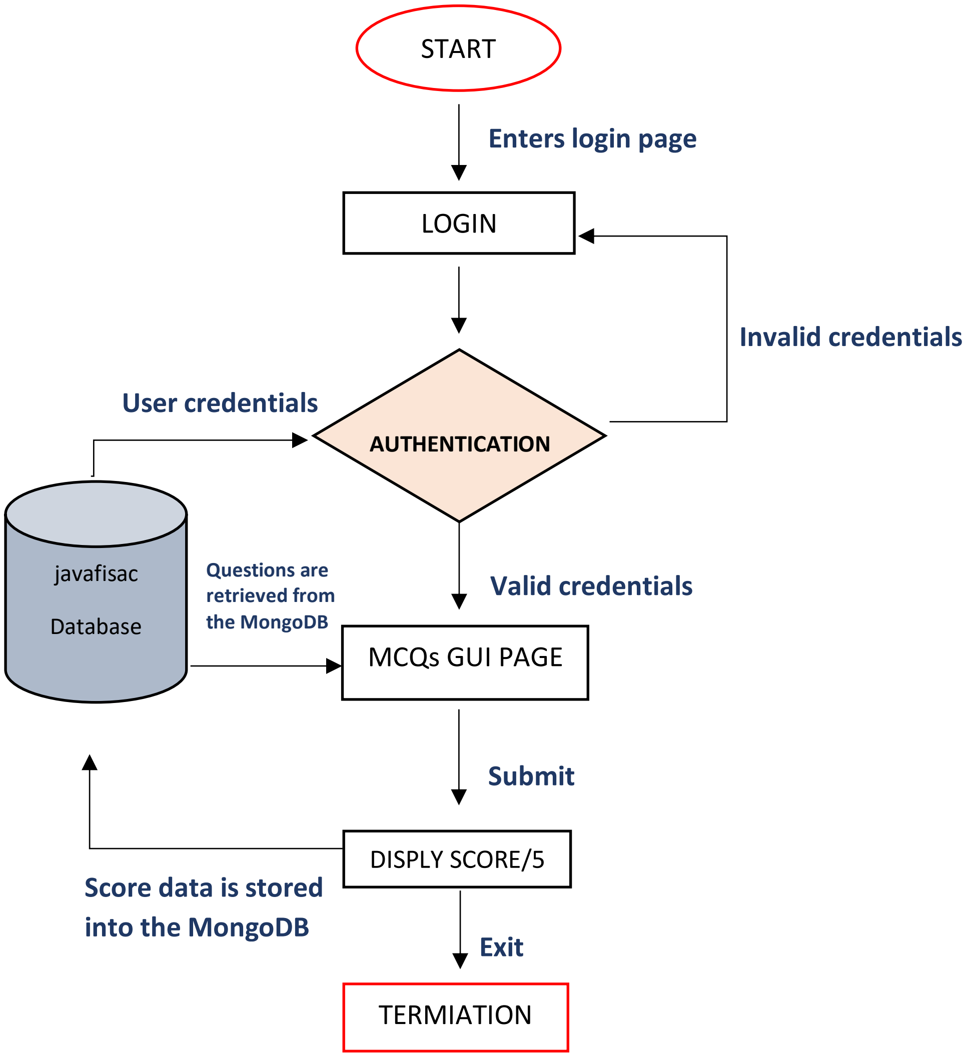
2. MCQ Page:

- **Display 5 MCQs on this page, each with a question and multiple radio buttons for choosing options.**
- **Users should be able to select one option per question.**
- **Provide a "Submit" button for users to submit their answers.**

3. Result Display:

- **After the user has attempted all the questions and submitted their answers, the application should calculate and display the total score and store the same in the database for each student.**

Flow diagram:



Swing Components Used:

Swing components used in the application:

- **JFrame:** Serves as the application's primary window
- **JPanel:** Used for arranging and grouping various components
- **JLabel:** Utilized for presenting text and labels for GUI
- **JTextField:** Allows text input (for username and password)
- **JPasswordField**: Facilitates password entry Field
- **JButton:** Triggers actions such as login, next and submit Buttons
- **JRadioButton:** Displays Radio button options for MCQs
- **ButtonGroup:** Groups radio buttons together
- **JOptionPane:** Shows pop-up messages after the score is displayed
- **MongoClient, MongoDatabase, MongoCollection:** These are part of the MongoDB Java driver and are used for database operations

Events, Actions:

- There is a Submit Button ('JButton'):
    - When clicked, it triggers an action listener that performs the following tasks:
        - Insert the user's responses (answers) into a MongoDB database.
        - Displays a message using 'JOptionPane' to show the user's score.
- Each answer option is represented by a 'JRadioButton':
    - Each 'JRadioButton' has an associated action listener.
    - The action listener checks if the selected 'JRadioButton' matches the correct answer.
    - If the user selects the correct answer, a 'correct answer' is matched from the MongoDB field named 'correctoption'.

Program Code with Comments:

Login.java:

```
//Nihal Kiran Shetty
//230970065
//MCA 2nd SEM section - A
package com;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import com.mongodb.client.*;
import org.bson.Document;

class Login extends JFrame {
   private final MongoClient mongoClient;
   private final MongoDatabase database;
```

```
    Login() {
        // Initialize MongoDB client and connect to the database
        mongoClient = MongoClients.create("mongodb://localhost:27017");
        database = mongoClient.getDatabase("javaFisac");

        loginGUI();
    }

    public void loginGUI() {
        setLayout(null);
        getContentPane().setBackground(Color.black);
        setTitle("ENTRENCE EXAM");
        setSize(380, 350);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //LOGIN GUI STRUCTURE
        JLabel lblhead = new JLabel("Login Form");
        JLabel lbluname = new JLabel("Username :");
        JLabel lblpasswd = new JLabel("Password :");
        final JTextField txtuname = new JTextField();
        final JPasswordField txtpwd = new JPasswordField(); // Use JPasswordField for
password
        JButton btnLogin = new JButton("Login");

        //SETING THE COORDINATES FOR THE GIU COMPONENTS
        lblhead.setBounds(70, 10, 250, 100);
        lbluname.setBounds(50, 110, 90, 30);
        txtuname.setBounds(160, 110, 130, 30);
        lblpasswd.setBounds(50, 150, 90, 30);
        txtpwd.setBounds(160, 150, 130, 30);

        btnLogin.setBounds(120, 220, 110, 30);

        lblhead.setFont(new Font("SansSerif", Font.BOLD, 40));
        lblhead.setForeground(Color.WHITE);
        lbluname.setForeground(Color.WHITE);
        lblpasswd.setForeground(Color.WHITE);
        txtuname.setBackground(Color.lightGray);
        txtpwd.setBackground(Color.lightGray);
        btnLogin.setBackground(Color.darkGray);
        btnLogin.setForeground(Color.WHITE);

        add(lblhead);
        add(lbluname);
        add(txtuname);
        add(lblpasswd);
        add(txtpwd);
        add(btnLogin);
        setVisible(true);

        final JLabel lblErr = new JLabel("Invalid username or password!");
        lblErr.setBounds(110, 170, 300, 50);
        lblErr.setForeground(Color.red);
        add(lblErr);
        lblErr.setVisible(false);

        btnLogin.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String uname = txtuname.getText();
                String pass = new String(txtpwd.getPassword()); // Get password as string

                // Check authentication against MongoDB
                MongoCollection<Document> collection = database.getCollection("users");
                Document query = new Document("username", uname).append("password", pass);
                Document user = collection.find(query).first();

                if (user != null) {
                    new Quiz(uname);
                } else {
                    lblErr.setVisible(true);
                }
            }
        });
    }

    public void quizGUI() {
        JLabel q1 = new JLabel("Question 1");
        q1.setBounds(50, 110, 90, 30);
        add(q1);
    }

    public static void main(String[] args) {
        new Login();
    }
}
```

Quiz.java:

```
//Nihal kiran Shetty
//230970065
//MCA 2nd SEM section - A

package com;

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import com.mongodb.client.*;
import org.bson.Document;
```

```
public class Quiz extends JFrame implements ActionListener {
    String selectedans[] = new String[10];
    String questions[][] = new String[10][5];
    String answers[] = new String[10];

    JLabel lblqno, lblqtn,lblhead;
    JRadioButton opt1, opt2, opt3, opt4;
    JButton btnNext, btnSubmit;
    ButtonGroup grpoptn;
    long qtcount;
    int count = 0, total = 0;

    String uname;

    Quiz(String uname)
    {
        this.uname = uname;

        getContentPane().setBackground(Color.lightGray);
        setBounds(50, 0, 1200, 650);
        setTitle("ENTRENCE EXAM");
        setLayout(null);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //MCQS GUI
        lblhead = new JLabel("WELCOME TO THE MCQs EXAM , WISH YOU ALL THE
BEST");
        lblqno = new JLabel(" ");
        lblqtn = new JLabel(" ");

        opt1 = new JRadioButton(" ");
        opt2 = new JRadioButton(" ");
        opt3 = new JRadioButton(" ");
        opt4 = new JRadioButton(" ");

        //FUNCTIONS TO FORMAT AND PLACE RADIO BUTTONS
        createOptions(opt1, 320);
        createOptions(opt2, 350);
        createOptions(opt3, 380);
        createOptions(opt4, 410);

        add(opt1);
        add(opt2);
        add(opt3);
        add(opt4);

        lblhead.setBounds(100, 0, 1000, 50);
        lblhead.setForeground(Color.BLACK);
        lblqno.setBounds(30, 275, 100, 30);
        lblqtn.setBounds(70, 275, 1200, 30);
        lblhead.setFont(new Font("Tahoma", Font.PLAIN, 35));
        lblqno.setFont(new Font("Tahoma", Font.PLAIN, 22));
        lblqtn.setFont(new Font("Tahoma", Font.PLAIN, 22));

        grpoptn = new ButtonGroup();
        grpoptn.add(opt1);
        grpoptn.add(opt2);
        grpoptn.add(opt3);
        grpoptn.add(opt4);

        btnNext = new JButton("Next");
        btnNext.setBounds(30, 450, 150, 100);
        btnNext.setFont(new Font("Tahoma", Font.PLAIN, 22));
        btnNext.setBackground(Color.BLACK);
        btnNext.setForeground(Color.white);

        btnSubmit = new JButton("Submit");
        btnSubmit.setBounds(980, 450, 150, 100);
        btnSubmit.setFont(new Font("Tahoma", Font.PLAIN, 22));
        btnSubmit.setBackground(Color.BLACK);
        btnSubmit.setForeground(Color.WHITE);

        btnNext.addActionListener(this);

        btnSubmit.addActionListener(this);

        add(btnNext);

        add(btnSubmit);

        add(lblhead);
        add(lblqno);
        add(lblqtn);

        for (int i = 0; i < 10; i++)
        {
            answers[i]="none";
        }

        setVisible(true);

        // Load questions from MongoDB
        loadQuestionsFromDB();

        quizFrames(count);
    }

    public void createOptions(JRadioButton opt, int y)
```

```
{
   opt.setBounds(30, y, 1100, 30);
   opt.setBackground(Color.white);
   opt.setFont(new Font("Dialog", Font.PLAIN, 20));
}

//FUNCTION TO LOAD QUESTION FROM MONGODB
public void loadQuestionsFromDB() {
   MongoClient mongoClient = null;
   try {
      //CONNECTION STRING TO MONGODB
      mongoClient = MongoClients.create("mongodb://localhost:27017");
      MongoDatabase database = mongoClient.getDatabase("javaFisac");
      MongoCollection<Document> collection = database.getCollection("questions");
      FindIterable<Document> questions = collection.find();

      qtcount=collection.countDocuments();

      int index = 0;
      for (Document question : questions) {
         this.questions[index][0] = question.getString("question");
         this.questions[index][1] = question.getString("option1");
         this.questions[index][2] = question.getString("option2");
         this.questions[index][3] = question.getString("option3");
         this.questions[index][4] = question.getString("option4");
         this.answers[index] = question.getString("correctOption");
         index++;

      }
   } catch (Exception e) {
      e.printStackTrace();
   } finally {
      if (mongoClient != null) {
         mongoClient.close();
      }
   }
}


public void quizFrames(int count) {
   lblqno.setText("" + (count + 1) + ".");
   lblqtn.setText(questions[count][0]);

   opt1.setText(questions[count][1]);
   opt2.setText(questions[count][2]);
   opt3.setText(questions[count][3]);
   opt4.setText(questions[count][4]);

   opt1.setActionCommand(questions[count][1]);
   opt2.setActionCommand(questions[count][2]);
```

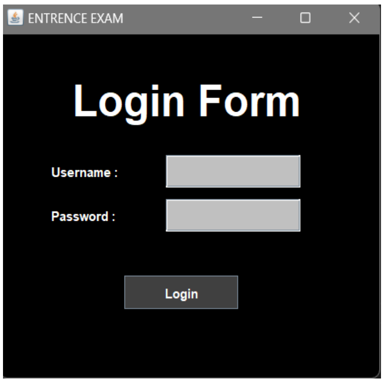
```
        opt3.setActionCommand(questions[count][3]);
        opt4.setActionCommand(questions[count][4]);

        grpoptn.clearSelection();
    }

    public void actionPerformed(ActionEvent e)
     {
        if (grpoptn.getSelection() != null) {
            //store the selected answer in userAns[]
            selectedans[count] = grpoptn.getSelection().getActionCommand();
    }

        if (e.getSource() == btnNext) {
            count++;
            if (count < 4)
                quizFrames(count);
            else {
                quizFrames(count);
                btnNext.setEnabled(false);
            }
        }

        if (e.getSource() == btnSubmit) {
            // Iterate over the selected answers and check against the correct answers
            for (int i = 0; i < answers.length; i++) {
                // Check if the selected answer is not null and is equal to the correct answer
                if (selectedans[i] != null && selectedans[i].equals(answers[i])) {
                    total++;
                }
            }

            // Display score
            new Score(uname, total);
        }

    }


    public static void main(String[] args) {
        new Quiz("uname");
    }
}
```

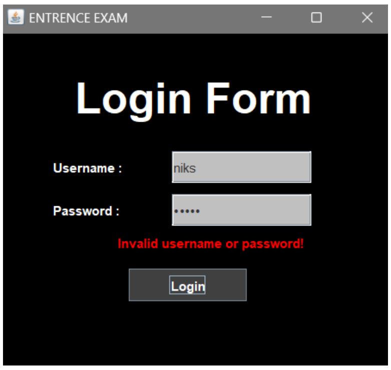
Score.java:

```
//Nihal kiran Shetty
//230970065
//MCA 2nd SEM section - A
```

```
package com;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import com.mongodb.client.*;
import com.mongodb.client.model.UpdateOptions; // Import required MongoDB classes

import org.bson.Document; // Import required MongoDB class for Document

public class Score extends JFrame implements ActionListener {
   String uname;
   int score;

   Score(String uname, int score) {
      this.uname = uname;
      this.score = score;

      setLayout(null);
      setTitle("ENTRENCE EXAM");
      setBackground(Color.WHITE);
      setSize(450, 350);
      setLocationRelativeTo(null);

      JLabel lblThank = new JLabel("Thank you " + uname + "!");
      lblThank.setBounds(90, 50, 300, 40);
      lblThank.setFont(new Font("Tahoma", Font.PLAIN, 30));
      add(lblThank);

      JLabel lblScore = new JLabel("Score: " + score);
      lblScore.setBounds(150, 150, 300, 40);
      lblScore.setFont(new Font("Tahoma", Font.PLAIN, 24));
      add(lblScore);

      JButton btnExit = new JButton("Exit");
      btnExit.setFont(new Font("Tahoma", Font.PLAIN, 22));
      btnExit.setBackground(Color.BLACK);
      btnExit.setForeground(Color.WHITE);
      btnExit.setBounds(150, 200, 100, 50);
      btnExit.addActionListener(this); // Add ActionListener to the exit button
      add(btnExit);

      setVisible(true);

      // Store score in MongoDB
      storeScoreInDB();
   }

   public void actionPerformed(ActionEvent e) {
      if (e.getActionCommand().equals("Exit")) {
            dispose(); // Close the JFrame
            System.exit(0); // Terminate the program
        }
    }

    public void storeScoreInDB() {
        MongoClient mongoClient = null;
        try {
            mongoClient = MongoClients.create("mongodb://localhost:27017"); // Create
MongoDB client
            MongoDatabase database = mongoClient.getDatabase("javaFisac"); // Access the
database
            MongoCollection<Document> collection = database.getCollection("users"); // Access
the collection

            // Create query document
            Document query = new Document("username", uname);
            Document update = new Document("$set", new Document("score", score));
            UpdateOptions options = new UpdateOptions().upsert(true);

            collection.updateOne(query, update, options); // Update or insert document
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (mongoClient != null) {
                mongoClient.close();
            }
        }
    }


    public static void main(String[] args) {
        new Score("", 0); // Create an instance of Score JFrame
    }
}
```

Screenshot of Output:

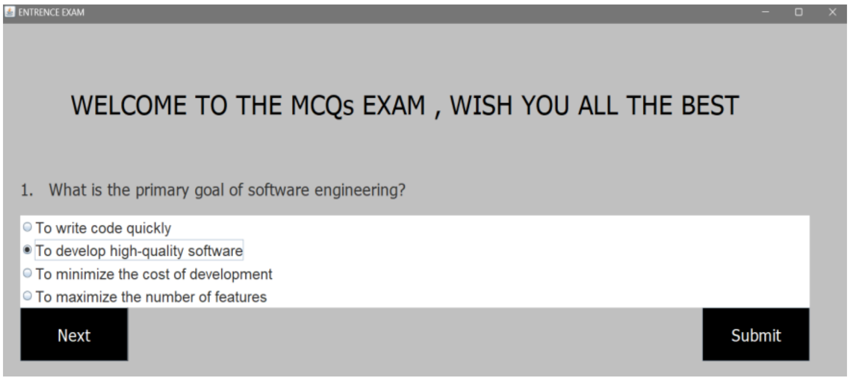




niks

•••••

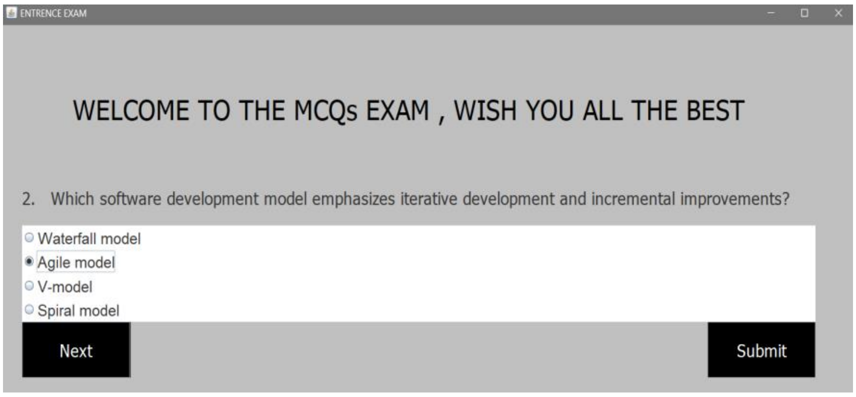
Invalid username or password!



WELCOME TO THE MCQs EXAM , WISH YOU ALL THE BEST

1. What is the primary goal of software engineering?

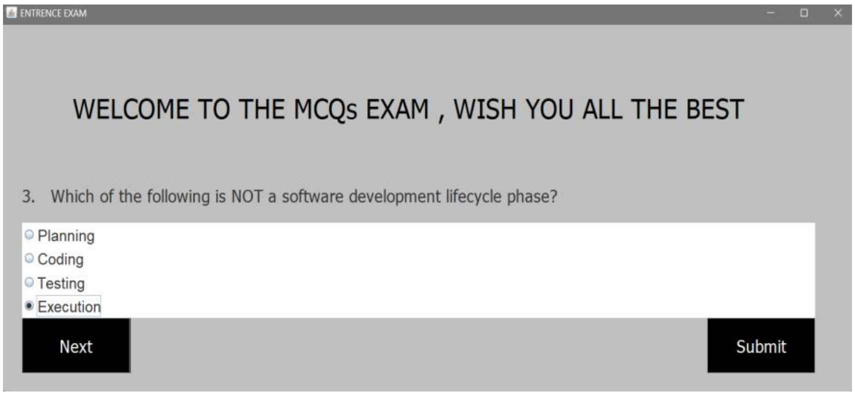
○ To write code quickly
● To develop high-quality software
○ To minimize the cost of development
○ To maximize the number of features



WELCOME TO THE MCQs EXAM , WISH YOU ALL THE BEST

2. Which software development model emphasizes iterative development and incremental improvements?

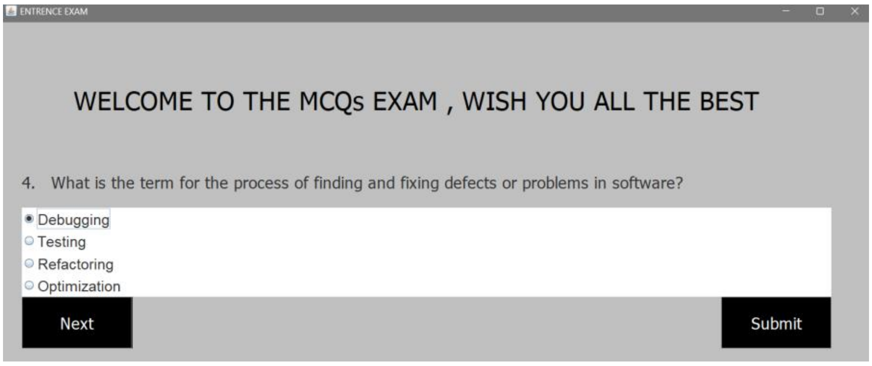
○ Waterfall model
● Agile model
○ V-model
○ Spiral model



WELCOME TO THE MCQs EXAM , WISH YOU ALL THE BEST

3. Which of the following is NOT a software development lifecycle phase?

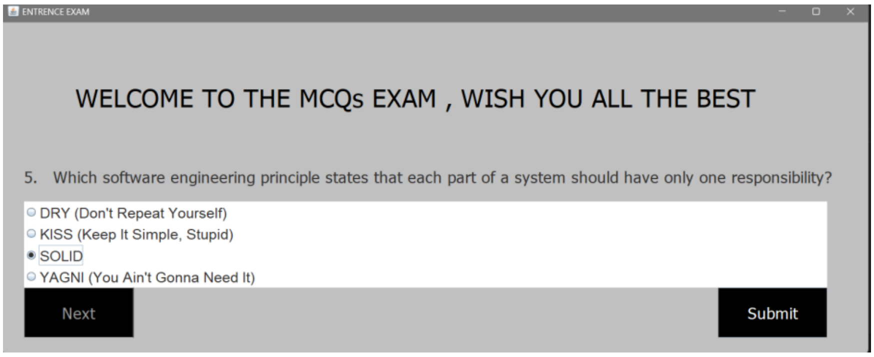
○ Planning
○ Coding
○ Testing
● Execution



WELCOME TO THE MCQs EXAM , WISH YOU ALL THE BEST

4. What is the term for the process of finding and fixing defects or problems in software?

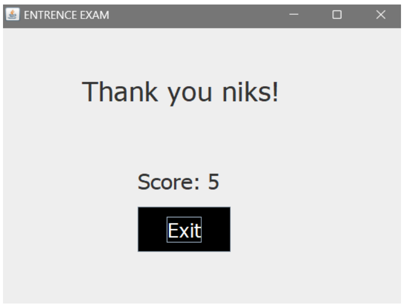
● Debugging
○ Testing
○ Refactoring
○ Optimization



WELCOME TO THE MCQs EXAM , WISH YOU ALL THE BEST

5. Which software engineering principle states that each part of a system should have only one responsibility?

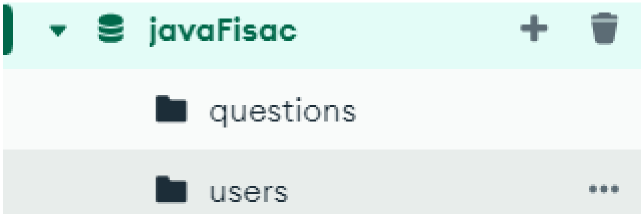
○ DRY (Don't Repeat Yourself)
○ KISS (Keep It Simple, Stupid)
● SOLID
○ YAGNI (You Ain't Gonna Need It)



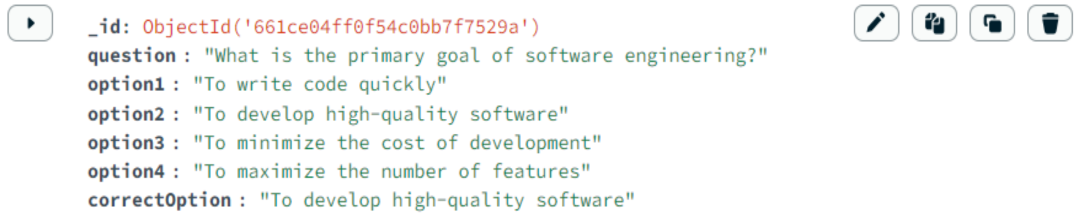
Thank you niks!

Score: 5

Database, Collections and Document Structure:

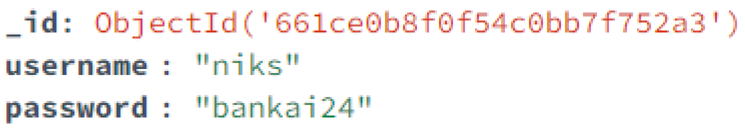


javaFisac(questions) collection sample records:



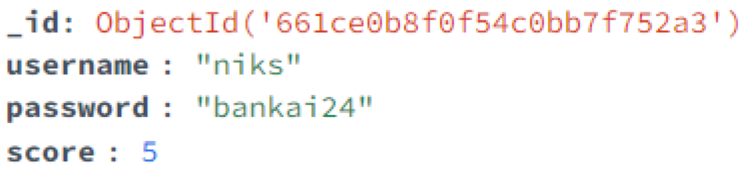
```
_id: ObjectId('661ce04ff0f54c0bb7f7529a')
question : "What is the primary goal of software engineering?"
option1 : "To write code quickly"
option2 : "To develop high-quality software"
option3 : "To minimize the cost of development"
option4 : "To maximize the number of features"
correctOption : "To develop high-quality software"
```

javaFisac(users) Collection samples records:



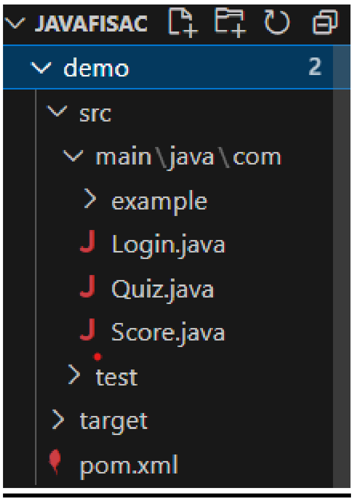
```
_id: ObjectId('661ce0b8f0f54c0bb7f752a3')
username : "niks"
password : "bankai24"
```

Score value is updated to javaFisac(users) collection sample records



```
_id: ObjectId('661ce0b8f0f54c0bb7f752a3')
username : "niks"
password : "bankai24"
score : 5
```

Project Structure:



```
JAVAFISAC
  demo 2
    src
      main\java\com
        example
        Login.java
        Quiz.java
        Score.java
      test
    target
    pom.xml
```

Reference:

- Oracle Java Documentation: Swing
- Schlidt, Herbert. *Java: the complete reference*, McGraw_Hill Education Group, 2014
- MongoDB Java Driver Documentation: MongoDB Java Driver
- Youtube.com