1.AdmissionUserInterface

```
package com.cts.collegeadmission.client;

import com.cts.collegeadmission.exception.MarkEligibilityException;
import com.cts.collegeadmission.service.AdmissionService;
import com.cts.collegeadmission.skeleton.SkeletonValidator;

public class AdmissionUserInterface {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		//VALIDATION STARTS
		new SkeletonValidator();
		//DO NOT CHANGE THIS CODE
		//VALIDATION ENDS

		//FILL THE CODE HERE
		AdmissionService addmissionService = new AdmissionService();
		try {
			addmissionService.addAdmissionList("inputfeed.txt");
		} catch (MarkEligibilityException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
	}

}
```

2.AdmissionDAO

```
package com.cts.collegeadmission.dao;

import java.sql.Connection;
import java.util.List;
import com.cts.collegeadmission.exception.MarkEligibilityException;
import com.cts.collegeadmission.model.Applicant;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import com.cts.collegeadmission.util.*;

public class AdmissionDAO {

	// FILL THE CODE HERE
	public boolean insertAdmissionList(List<Applicant> Admission_List) throws
MarkEligibilityException {
		boolean recordsAdded = false;

		// FILL THE CODE HERE
		Connection con = null;
		try {
			int status[];
			DBConnectionManager dbManager = DBConnectionManager.getInstance();
```

```
                con = dbManager.getConnection();
                PreparedStatement pstmt = con.prepareStatement(
                        "INSERT                                                                 INTO
ENGGADMISSION_LIST('ADMISSION_NO','APPLICANT_NAME','DATEOFBIRTH','PHYSICS_MAR
K','CHEMISTRY_MARK','MATHS_MARK','PREFERRED_STREAM','ADMISSION_FEE')
VALUES(?,?,?,?,?,?,?,?)");
                for (Applicant student : Admission_List) {
                    try {

                        pstmt.setString(1, student.getAdmissionNo());
                        pstmt.setString(2, student.getApplicantName());
                        pstmt.setDate(3,
ApplicationUtil.utilToSqlDateConverter(student.getDateOfBirth()));
                        pstmt.setDouble(4, student.getGender());
                        pstmt.setDouble(5, student.getPhyMark());
                        pstmt.setDouble(6, student.getChemMark());
                        pstmt.setDouble(7, student.getMathsMark());
                        pstmt.setString(8, student.getChoiceOfStream());
                        pstmt.setDouble(9, student.getAdmissionFee());
                        pstmt.addBatch();
                    } catch (SQLException e) {
                        con.rollback();
                        con.commit();
                        // TODO Auto-generated catch block
                        e.printStackTrace();

                    }
                }
                status = pstmt.executeBatch();
                recordsAdded = true;
                for (int s : status) {
                    if (s < 0) {
                        recordsAdded = false;
                    }
                }
            }

            catch (SQLException e) {
                e.printStackTrace();

            }

            finally {
                if (con != null) {
                    try {
                        con.close();
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }
            return recordsAdded;

        }

}
```

3.DBConnectionManager

```
package com.cts.collegeadmission.dao;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.sql.Connection;
import java.util.Properties;
import java.sql.DriverManager;
import java.sql.SQLException;

import com.cts.collegeadmission.exception.MarkEligibilityException;

public class DBConnectionManager {

	private static Connection con = null;
	private static DBConnectionManager instance;

	public DBConnectionManager() throws MarkEligibilityException {

	}

	public static DBConnectionManager getInstance() throws MarkEligibilityException {

		// FILL THE CODE HERE
		if (instance == null) {
			instance = new DBConnectionManager();
			try {

				Properties prop = new Properties();
				InputStream fis = new FileInputStream("database.properties");
				prop.load(fis);
				Class.forName(prop.getProperty("DB_DRIVER_CLASS"));
				con = DriverManager.getConnection(prop.getProperty("DB_URL"),
prop.getProperty("DB_USERNAME"),
					prop.getProperty("DB_PASSWORD"));

			} catch (IOException | ClassNotFoundException | SQLException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
		}
		return instance;
	}

	public Connection getConnection() {
		return con;
	}
}
```

==========
==========

4.MarkEligibilityException

```
package com.cts.collegeadmission.exception;

public class MarkEligibilityException extends Exception {
	String strMsg1;
	Throwable strMsg2;

	public MarkEligibilityException() {
		super();
	}

	public MarkEligibilityException(String strMsg1) {
		super(strMsg1);
	}

	public MarkEligibilityException(String strMsg1, Throwable strMsg2) {
		super();
		this.strMsg1 = strMsg1;
		this.strMsg2 = strMsg2;
	}
}
```

==========
==========

5.Applicant

```
package com.cts.collegeadmission.model;

import java.util.Date;

public class Applicant {
	private String admissionNo;
	private String ApplicantName;
	private Date dateOfBirth;
	private char gender;
	private double phyMark;
	private double chemMark;
	private double mathsMark;
	private String choiceOfStream;
	private double admissionFee;

	public Applicant() {
		super();
		// TODO Auto-generated constructor stub
	}

	public Applicant(String admissionNo, String applicantName, Date dateOfBirth, char
```

```
gender, double phyMark,
                double chemMark, double mathsMark, String choiceOfStream, double
admissionFee) {
        super();
        this.admissionNo = admissionNo;
        ApplicantName = applicantName;
        this.dateOfBirth = dateOfBirth;
        this.gender = gender;
        this.phyMark = phyMark;
        this.chemMark = chemMark;
        this.mathsMark = mathsMark;
        this.choiceOfStream = choiceOfStream;
        this.admissionFee = admissionFee;
    }

    public String getAdmissionNo() {
        return admissionNo;
    }

    public void setAdmissionNo(String admissionNo) {
        this.admissionNo = admissionNo;
    }

    public String getApplicantName() {
        return ApplicantName;
    }

    public void setApplicantName(String applicantName) {
        ApplicantName = applicantName;
    }

    public Date getDateOfBirth() {
        return dateOfBirth;
    }

    public void setDateOfBirth(Date dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }

    public char getGender() {
        return gender;
    }

    public void setGender(char gender) {
        this.gender = gender;
    }

    public double getPhyMark() {
        return phyMark;
    }

    public void setPhyMark(double phyMark) {
        this.phyMark = phyMark;
    }

    public double getChemMark() {
        return chemMark;
```

```
	}

	public void setChemMark(double chemMark) {
		this.chemMark = chemMark;
	}

	public double getMathsMark() {
		return mathsMark;
	}

	public void setMathsMark(double mathsMark) {
		this.mathsMark = mathsMark;
	}

	public String getChoiceOfStream() {
		return choiceOfStream;
	}

	public void setChoiceOfStream(String choiceOfStream) {
		this.choiceOfStream = choiceOfStream;
	}

	public double getAdmissionFee() {
		return admissionFee;
	}

	public void setAdmissionFee(double admissionFee) {
		this.admissionFee = admissionFee;
	}

	@Override
	public String toString() {
		return "Applicant [admissionNo=" + admissionNo + ", ApplicantName=" +
ApplicantName + ", dateOfBirth="
				+ dateOfBirth + ", gender=" + gender + ", phyMark=" + phyMark + ",
chemMark=" + chemMark
				+ ", mathsMark=" + mathsMark + ", choiceOfStream=" + choiceOfStream + ",
admissionFee=" + admissionFee
				+ "]";
	}

}
```

==========

6.AdmissionService

```
package com.cts.collegeadmission.service;

import java.io.FileNotFoundException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Date;
```

```
import java.util.List;

import com.cts.collegeadmission.dao.AdmissionDAO;
import com.cts.collegeadmission.exception.MarkEligibilityException;
import com.cts.collegeadmission.model.Applicant;
import com.cts.collegeadmission.util.*;

public class AdmissionService {

	public static ArrayList<Applicant> buildAdmissionList(List<String> AdmissionRecords) {

		final String COMMADELIMITER = ",";
		ArrayList<Applicant> admissionList = new ArrayList<Applicant>();
		for (String student : AdmissionRecords) {
			String details[] = student.split(COMMADELIMITER);
			Applicant applicant = new Applicant(details[0], details[1],
					ApplicationUtil.stringToDateConverter(details[2]),
details[3].toCharArray()[0],
					Double.parseDouble(details[4]),		Double.parseDouble(details[5]),
Double.parseDouble(details[6]),
					details[7], 0);

applicant.setAdmissionFee(calculateAdmissionFee(applicant.getChoiceOfStream()));
			System.out.println(applicant.toString());
			admissionList.add(applicant);
		}
		return admissionList;
	}

	public boolean addAdmissionList(String inputFeed) throws MarkEligibilityException {
		//FILL THE CODE HERE
		try {
			List<String> admissionList = ApplicationUtil.readFile(inputFeed);
			ArrayList<Applicant> admissionArrayList = buildAdmissionList(admissionList);
			AdmissionDAO admissionDAO = new AdmissionDAO();
			try {
				admissionDAO.insertAdmissionList(admissionArrayList);
				return true;
			} catch (MarkEligibilityException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
		} catch (FileNotFoundException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		return false;

	}

	public static double calculateAdmissionFee(String streamChoice) {

		double fees = 0;
		// FILL THE CODE HERE
```

```
		switch (streamChoice.toUpperCase()) {
		case "CSE":
		case "IT":
		case "AERO":
			fees = 100000;
			break;
		case "MECH":
		case "CIVIL":
			fees = 75000;
			break;
		case "EIE":
		case "EEE":
		case "ECE":
			fees = 50000;
			break;
		default:
			fees = 0;
			break;

		}
		return fees;
	}

}
```

==========

7.ApplicationUtil

```
package com.cts.collegeadmission.util;

import java.io.File;
import java.io.FileNotFoundException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Scanner;

import com.cts.collegeadmission.exception.MarkEligibilityException;

public class ApplicationUtil {

	public static List<String> readFile(String inputfeed) throws FileNotFoundException {
		final String COMMADELIMITER = ",";
		List<String> admissionList = new ArrayList<String>();
		// FILL THE CODE HERE
		Scanner myReader = null;
		try {
			File myObj = new File(inputfeed);
			myReader = new Scanner(myObj);
```

```
                while (myReader.hasNextLine()) {
                        String line = myReader.nextLine();
                        String inputLine[] = line.split(COMMADELIMITER);
                        try {
                                if                      (checkFor_Eligibility(Double.parseDouble(inputLine[4]),
Double.parseDouble(inputLine[5]),
                                                Double.parseDouble(inputLine[6]))) {
                                        admissionList.add(line);
                                } else {
                                        throw new MarkEligibilityException("Not Eligible");
                                }
                        } catch (NumberFormatException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        } catch (MarkEligibilityException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                }
        } catch (FileNotFoundException e) {
                e.printStackTrace();
        } finally {
                myReader.close();
        }
        return admissionList;
    }

    public static boolean checkFor_Eligibility(double phy, double chem, double maths) throws
MarkEligibilityException {
        boolean eligibility = false;
        double totalMarks = 0;
        if (phy >= 50 && chem >= 50 && maths >= 50) {
                totalMarks = phy + chem + maths;
                if (totalMarks > 224) {
                        eligibility = true;
                }
        }
        return eligibility;
    }

    public static Date stringToDateConverter(String stringDate) {
        // FILL THE CODE HERE
        SimpleDateFormat formatter2 = new SimpleDateFormat("dd-MM-yyyy");
        Date bithdate = null;
        try {
                bithdate = formatter2.parse(stringDate);
        } catch (ParseException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
        return bithdate;

    }

    public static java.sql.Date utilToSqlDateConverter(java.util.Date utDate) {
        java.sql.Date sqlPackageDate = new java.sql.Date(utDate.getDate());
        return sqlPackageDate;
```

```
    }
}
```