**Mini Project 1**

**Project Name - Applicant Management System**

**Objective:-** Create a console based Java application that would allow the Admin of a training Institute to add and view Applicant information details as per the design specifications given below.  The data received from the user (Admin) will be stored in database and retrieved when required.

**Design:**

**1.      Create a new user in database**

Then the oracle user should be **abc** and the password should be **abc**

* For JDBC connection, only use **orcl** as service name and **1521** as port number

**2.      Create new user**

* Open command prompt
* Type Sqlplus / as sysdba
* Create user <user> identified by <pswd>;     [ For example to create a user named “test” with password “test” : create user test identified by test; ]
* Grant connect,resource to <username>;  [ E.g: grant connect,resource to test;]
* Commit;
* Exit;

**3.      Create Table using SQL Commands**

**Name of Table: APPLICANT\_TBL**

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** |  |
| **applicant\_ID** | Varchar2(6) (Primary key) |  |
| **applicant\_name** | Varchar2(15) |  |
| **marks1** | Number(3) |  |
| **marks2** | Number(3) |  |
| **marks3** | Number(3) |  |
| **applicant\_result** | Varchar2(15) |  |
| **applicant\_grade** | Varchar2(15) |  |

**4.       Sequence:**

**Sequence Name : APPLICANT\_SEQ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Min Value** | **Max Values** | **Increment by** | **Starting** |
| 6000 | 7000 | 1 | 6000 |

**B. System Design:**

|  |  |
| --- | --- |
| **Name of the package** | **Usage** |
| com.company.applicant.service | This package will contain the class which displays the console menu and take user input. |
| com.company.applicant.bean | This package will contain the entity class named ApplicantBean. |
| com.company.applicant.dao | This package will contain the class that will do the database related JDBC code. |
| com.company.applicant.util | This package will contain the class to establish database connection and also the class that handles the user defined exception. |

**Package:**com.company.applicant.util

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **DBUtil** |  | DB connection class |
|  | public static Connection **getConn()** | Establish a connection to the database and return the java.sql.Connection reference |
| **WrongDataException** |  | User defined exception class.  Override the toString() method of the Object class and return a String **“Data is Wrong”**. This exception will be thrown in the ApplicantMain class methods whenever invalid input is given. The details about when it has to be thrown is given in the respective methods of the ApplicantMain class |

**Package:**com.company.applicant**.**bean

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **ApplicantBean** |  | Bean class |
|  | private String id | Student Id |
|  | private String name | Student name |
|  | private  int marks1 | Mark in First Subject |
|  | private  int marks2 | Mark in Second Subject |
|  | private  int marks3 | Mark in Third Subject |
|  | private String applicant\_result | applicant\_result |
|  | private String applicant\_grade | applicant\_grade |
|  | setters & getters | Should create the getter and setter methods for all the attributes mentioned in the class |

**Package:**com.company.applicant**.dao**

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **ApplicantDAO** |  | DAO class |
|  | public String **addApplicant**(ApplicantBean ApplicantBean) | 1. This method should take the values from the ApplicantBean object and insert it into the database. 2. If the insertion is successful, then a String “SUCCESS” should be returned, else a String “FAIL” should be returned. 3. If any JDBC exception such as SQLException occur, this function should return “FAIL” |
|  | public ArrayList<Applicant Bean> **getByResult**(String criteria) | 1. This method should use the JDBC select statement to retrieve the records based on the criteria given. 2. If the criteria String contains **“PASS”** then the **getByResult(String criteria)**function should return an ArrayList of all Applicants who have passed 3. If the criteria String contains **“FAIL”** then the **getByResult(String criteria)**function should return an  ArrayList of all Applicants who have failed 4. If the criteria String contains **“ALL”** then the **getByResult(String criteria)**function should return an ArrayList of all the Applicants 5. In any of the criteria’s “PASS/FAIL/ALL” if there are no matching records then the function should return **null** 6. In case of any JDBCExceptions in the database then a **null** value needs to be returned |
|  | public String **generateApplicantId**(String name) | 1. This method should contain the necessary code to create a new Applicant id. 2. ApplicantID is a combination of **first 2 letters of name** in **uppercase** followed by **4 digit number that will be generated by the oracle sequence APPLICANT\_SEQ.** 3. For eg, the Applicant id for a Applicant Jacob could be ORA3456 4. The function should return the generated Applicant id. |

|  |  |  |
| --- | --- | --- |
| **Class** | **Method and Variables** | **Description** |
| **Applicant Main** |  | Main class |
|  | public static void **main**(String[] args) | The code that is needed to test your program goes here. A sample code is shown at the end of the document. |
|  | public String **addApplicant**(ApplicantBean ApplicBean) |          This method should add a ApplicantBean Object to the database.           The following are the conditions under which a user defined exception **WrongDataException** (found in  com.company.applicant **.util** package) should be thrown.     ApplicBean  is null     ApplicBean ’s name is empty String     ApplicBean ’s name contains less than 2 characters      ApplicBean ’s mark1,mark2,mark3 contains marks which are  not within 0 to 100 range.           This exception should be handled within the **addApplicant(ApplicantBean** ApplicBean**)** function itself.           If this exception is caught, then the function is expected to return a String “**incorrect data**”.  **NOTE:** Do **NOT**use **System.exit(0)** while handling the exception.           **Compute Applicant ID**    If the candBean object is valid, this function should call the **generateApplicantId (String name)** function of the **ApplicantDAO**  class to obtain the Applicant id. The ApplicBean’s name should be passed as parameter to the **generateApplicantId (String name)** function           **The ApplicBean’s id should be initialized using the Applicant Id that is received in the previous step**           **Compute Applicant\_Result and Applicant\_Grade**    The Applicant\_Result and Applicant\_Grade are computed using the following logic    MARKS1=mark1 of ApplicBean    MARKS2=mark2 of  ApplicBean    MARKS3=mark3 of  ApplicBean   |  |  |  | | --- | --- | --- | | **Total Marks** | **Applicant\_Result** | **Applicant\_Grade** | | (MARKS1+MARKS2+MARKS3)>= 240 | PASS | Distinction | | (MARKS1+MARKS2+MARKS3)>=  180 and (MARKS1+MARKS2+MARKS3)<240 | PASS | First Class | | (MARKS1+MARKS2+MARKS3)>=  150 and (MARKS1+MARKS2+MARKS3)<180 | PASS | Second Class | | (MARKS1+MARKS2+MARKS3)>= 105 and (MARKS1+MARKS2+MARKS3) <150 | PASS | Third Class | | (MARKS1+MARKS2+MARKS3)  <105 | FAIL | No Applicant\_Grade |            **Initialise the applicBean’s Applicant\_Result and Applicant\_Grade with the computed values**           Invoke **addApplicant(ApplicantBean ApplicantBean)** of the ApplicantDAO class to insert the candBean into the database.           On successful storage of the Applicant Details, to the table,  the function should return the ApplicantID and Applicant\_Result of the particular ApplicantBean           [ E.g if the Applicant ID generated is   SA1001, and the Applicant\_Result is PASS then the success message should be **SA1001:PASS**           If by any reason, the record is not stored, then the function should  return the String **Error** |
|  | public ArrayList<ApplicantBean> displayAll(String criteria) |          This method should return the collection of the Applicants from the Applicant table who are matching the given criteria           The criteria string can have values such as “PASS/FAIL/ALL”.           If the criteria contains either “PASS/FAIL/ALL” then invoke **getByResult(String criteria) of ApplicantDAO class and receive the collection**           If the criteria String contains any other values then the WrongDataException need to be thrown, and the function should return a null value.           **NOTE:** Do **NOT**use **System.exit(0)** while handling the exception. |

**Package:**com.company.applicant**.service**

**Test Cases:**

Test 1: Test for Null Value for ApplicantBean

Test 2: Test for Applicant Name to be Empty

Test 3: Test for Applicant Name less than 2 characters

Test 4: Test for Invalid range of Marks

Test 5: Test for correct ApplicantID generation

Test 6: Test for checking correct computation of PASS, FAIL and Applicant\_Grade

Test 7: Test for null value generation for different criteria (PASS/FAIL/ALL)

Test 8: Test the number of records retrieved for different criteria

**Main Method:**

**You can write code in the main method and test all the above test cases. A sample code of the main function to test the first test case is shown below for your reference.**

public static void main(String[] args) {

ApplicantMain applicantmain = new ApplicantMain();

}