Problem Statement: Diagnostic Center

XYZ diagnostic center conducts many diagnostic tests on patients, and the company needs a billing application to add diagnostic details, the application should also support viewing billing details of the tests conducted on a specified date and amount collected for a given patient.

The application does not store/capture the actual diagnosed result like Sugar level, blood pressure, etc.

Design and implement console based application program to automate the process of billing.
The main menu contains following options

1. **Add Diagnostic tests**
2. **Generate Report**
3. **Exit**

Above menu must be redisplayed after each operation is completed, except for the **'Exit'** option. The details for each option are specified in the rest of the document.

Your application program must follow the control flow as, start from class
DiagnosticSystemClient -> DiagnosticSystemManager->DAO class -> Database

Instructions:

I. DiagnosticSystemClient class contains main method, which would interact with user for receiving input and displaying the output

II. DiagnosticSystemManager class should contain all business logic and would interact with DAO class for any data from database

III. DAO class must implement an interface and contain all DB transaction logic

IV. DB script:

- Upload your table script, stored procedure and inserted values along with project solution.

V. Generics:

- Use Generics wherever applicable (parameters and return type of methods)

VI. Exception handling:

- All user defined exceptions must be created under com.mindtree.exceptions package
- Exceptions should be propagated from DAO layer.
- SQLException should be caught in DAO layer, wrap this exception as user defined exception and propagate to the caller code

- All exceptions must be handled in Manager Class
- The client class should display the appropriate message to the user based on type of exception threw by Manager Class.

VII. Loose coupling:

- Create a separate Dao interfaces and implementation classes. Database interaction code should be only in DAO classes.
- Dao classes should not have UI code.

1. Database design:

Create database **"Diagnostic_DB**" and create the following three tables used for the application: **"Patient", "Test" and "Patient_Test".** Listed below is the table design with constraints.

a) Table Name: Patient

Column Name	Data type	Constraint
P_ID	int	Primary key
Name	varchar(20)	NOT NULL
Email	varchar(20)	NOT NULL
Phone	varchar(10)	NOT NULL
DOB	Date	NOT NULL

Note: Create a **Stored Procedure** to insert records into **'Patient'** table. [No need to create Java code. Provide database scripts file for creating stored procedure and inserting records]. Below are the sample values of the table:

PID	Name	Email	Phone	DOB
3001	Ramesh	ramesh@gmail.com	9900012345	1984-12-1
3002	Suresh	suresh@gmail.com	9700011111	1990-05-23
3003	Mahesh	mahesh@gmail.com	8800883333	1968-10-15
3004	Rajesh	rajesh@gmail.com	3675211234	1972-11-03

b) Table Name:Test

Column Name	Data type	Constraint
T_ID	int	Primary key
Name	varchar(30)	NOT NULL
Cost	int	NOT NULL

Note: Create a **Stored Procedure** to insert records into **'Test'** table. [No need to create Java code. Provide database scripts file for creating stored procedure and inserting records]. Below are the sample values of the table:

Test		
T_ID	**Name**	**Cost**
1001	Blood Group	100
1002	ECG	1350
1003	HBA1c	1050
1004	Bone Densitometry	3000
1005	MRI Scanning	5000

c) Table Name:Patient_Test

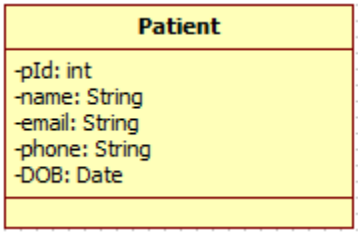
Column Name	Data type	Constraint
P_ID	int	Foreign key, references P_id column of Patient table
T_ID	int	Foreign key, references T_id column of Test table
Date	Date	NOT NULL

Note: The values for **'Patient_Test'** table must be inserted through your application.

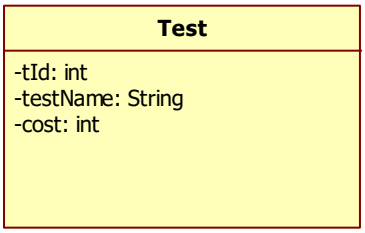
2. Entity classes:

Create following entity classes under package com.mindtree.entity

A. Patient



B. Test



DiagnosticSystemClient class displays the menu as below:

```
XYZ Diagnostic Center
---------------------
1.Add Diagnostic tests
2.Generate Report
4.Exit
Enter your choice:
```

3. Add Diagnostic tests

- When user selects option 1, it must allow user to enter test details of the patient.
- Select patient from existing patient list by using email or phone number of the patient
- It should allow user to enter multiple tests for a selected patient
- Sample output shown below.

```
Enter tests details
--------------------
1. Fetch patient-details by email
2. Fetch patient-details by phone
Enter your choice:1

Enter Email:rajesh@gmail.com
Enter Test Name: ECG
Want to add more Tests [YES/NO]: YES
Enter Test Name: HBA1c
Want to add more Tests [YES/NO]: NO
```

Note:

a) Save tests details entered for a selected patient along with Diagnostic date [use System Date, do not accept it from user].

b) Use collection class (ArrayList<Test>) to transfer tests details from client -> Manager->DAO

Write code to display the output in below format if tests details are successfully saved in the table '**Patient_Test'**, otherwise display an appropriate exception/error message.

```
Tests details are successfully saved!!
```

The **DiagnosticSystemManager class** code should check for the business rules given below and throw appropriate user defined exception, when exception thrown corresponding message should be displayed to user, which is as given in the table:

Rule No	Business constraint	User defined exceptions to be thrown	Message to user to be displayed
1	Email id entered by the user should exist in the database	If provided email id doesn't present then throw **InvalidEmailIdException**	"Invalid email id please check your input"
2	Phone number entered by the user should exist in the database	If provided phone number doesn't present then throw **InvalidPhoneNumberException**	"Invalid phone number please check your input"
3	Test name entered by the user should exist in the database	If provided test name doesn't not present then throw **InvalidTestNameException**	"Invalid test name please check your input"

4. Generate Report

- When user selects option 2, user must be able to generate report for a selected patient.
- Select patient from existing patient list by using email id or phone number of the patient
- Enter Diagnostic Test Date.
- The report should display the Test details sorted by test name (use collection framework)
- Sample output shown below.

```
Generate Report
-----------------------------------------------
1. Fetch Patient by Email
2. Fetch Patient by Phone
Enter your choice:1

Enter Email: rajesh@gmail.com
Enter Test Date [DD-MM-YYYY]:  03-12-2014

Output:
Name: Rajesh
Email: rajesh@gmail.com
Phone: 9000231345
Date: 03-12-2014

Test Details:
                Test            Cost
                ECG             1350.00
                HBA1c           1050.00
                Total:          2400.00
-----------------------------------------------
```

Note:

a) The output also must be saved in a text file (with patient id as file name) for future purpose.

b) Use group by/ joins/index based queries to fetch test details from database

The **DiagnosticSystemManager class** code should check for the business rule given below and throw appropriate user defined exception, when exception thrown corresponding message should be displayed to user, which is as given in the table:

Rule No	Business constraint	User defined exceptions to be thrown	Message to user to be displayed
1	Date entered by the user must be valid format (dd-mm-yyyy)	If date entered is not in valid format then throw **InvalidDateException**	"Invalid date please check your input"

5. Exit

- When user selects option 3, application should terminate