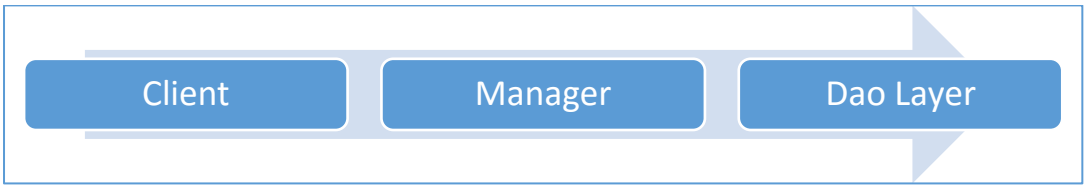
Problem Statement: Football matches

Football League wants an application to keep track of teams and matches.

Read the complete document before coding.

Your solution should follow below application Architecture:



Instructions:

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

II. **FootbalMatchManager** 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. Upload your table script with inserted values along with project solution.

V. Exception handling:

a. All user defined exceptions must be created under com.mindtree.exceptions package
b. Exceptions should be propagated from DAO layer.
c. SQLException should be caught in DAO layer, wrap this exception as user defined exception and propagate to the caller code
d. All exceptions must be handled in Manager Class
e. The client class should display the appropriate message to the user based on type of exception threw by Manager Class.

VI. **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 “FOOTBALL_MATCH_DB” and create the following two tables “TEAMS” and “MATCH”.

Table design details for “**TEAMS**” table.

COLUMN NAME	DATA TYPE	CONSTRAINTS
TEAM_NAME	VARCHAR(25)	PRIMARY KEY
TEAM_CITY	VARCHAR(30)	NOT NULL

Insert the following records manually into “**TEAMS**” table:

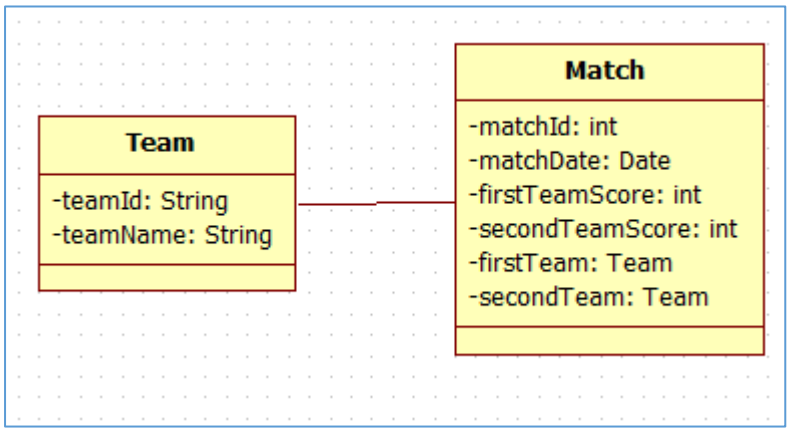
TEAM_NAME	TEAM_CITY
Delhi Dynamos	Delhi
Goa	Margoa
Kerala Blasters	Kochi
Chennaiyan	Chennai
Bengaluru FC	Bangalore

Table design details for “**MATCHES**” table.

COLUMN NAME	DATA TYPE	CONSTRAINTS	DETAILS
MATCH_ID	INT	PRIMARY KEY, AUTO INCREMENT	
MATCH_DATE	TIMESTAMP	NOT NULL	DATE-TIME when match was played
FIRST_TEAM_NAME	VARCHAR(25)	FOREIGN KEY	REFERENCES TEAM_ID OF TEAM TABLE
SECOND_TEAM_NAME	VARCHAR(25)	FOREIGN KEY	REFERENCES TEAM_ID OF TEAM TABLE
FIRST_TEAM_GOALS	INT	NOT NULL	**Goals scored by first team**
SECOND_TEAM_GOALS	INT	NOT NULL	**Goals scored by second team**

The values for “MATCHES” should be inserted through your application.

2. **Entity classes:**
   Create following entity classes under package "com.mindtree.entity”



3. Application:

Application client class displays the Menu as below have the following three options:

1. **Add Match details**
2. **List all matches played by a given team.**
3. **Exit**

3.1 Update Match details

- When user selects option 1, it must display the available teams and prompt the user to select "two teams", enter "match date" and "goals scored by both the teams"
- Sample output shown below.

```
Enter Choice [1/2/3]
1
Bengaluru FC    Bengaluru
Chennaiyan      Chennai
Delhi Dynamos   Delhi
Goa             Margoa
Kerala BlastersKochi
Select First Team Name:
Chennaiyan
Select Second Team Name:
Delhi Dynamos
Enter Match Date:
12-2-2013 4:30
Enter Goals scored by First Team:
2
Enter Goals scored by Second Team:
1
Match details added..
```

3.2 List all matches played by a given team

Accept team Name from user input and display Match date, opponent team, goals scored by both teams and output should be sorted by goal difference in descending order. Use Java Collection API for displaying "goal difference in descending order. If goal difference is same then sort based on MATCH_DATE.

Note: "goal difference in descending order" is not natural ordering.

```
Enter Choice [1/2/3]
2
Enter team Name:
Chennaiyan
*********
Team Name: Chennaiyan
MATCH_DATE           -              OPPONENT-                GOALS
15-02-2013 12:30     -                   Goa-                3 - 1
16-02-2013 04:15     -                   Goa-                2 - 2
13-02-2013 06:45     -     Kerala Blasters-                  1 - 1
12-02-2013 04:30     -       Delhi Dynamos-                  3 - 5
*********
```

3.2 Exit

When user selects option 3, application should terminate.

4. Business Rules

4.1 Manger class should check for the following business rules and throw appropriate user defined exceptions:

Rule	Business Constraint	User Defined Exception to be thrown	Message to be displayed to the user
1	Match Date should be of format DD-MM-YYYY HH:MM format	If rule is violated throw **InvalidInputException**	"Date format should be [DD-MM-YYYY HH:MM]"
2	Goals should not be negative value	If rule is violated throws **InvalidInputException**	"Invalid goal"
3	Selected TEAM NAME should be present in the database	If rule is violated throw **FetchException**	"Team XXXXX does not exist"