PROJECT REPORT

Software Requirements Specification & Detailed Design Document

for

SPORTS EVENTS MANAGEMENT SYSTEM

Version 1.0

Prepared By

G.Supriya B121286

M.Tulasi Renuka B121050

S.Sudharani B121289

D.Pallavi B121007

Ch.Shirisha B121110

AB-1 007 E3 CSE

RGUKT BASAR

22-01-2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1 Introduction

1.1 Purpose

The main purpose of our project is to simplify the process of handling each sports event by providing a web interface for student and teacher. This project brings the entire process of sports event management online which is built for students to easily get the information about any sports in campus. It overcomes the dependency of a single person handling all the sports activity.

1.2 Document Conventions

IEEE Format. Size of main headings are Times 18 and sub-headings are Times New Roman 14 respectively. The font size of the remaining description is Arial 11.

1.3 Intended Audience and Reading Suggestions

We are developing Sports Events Management System. This SRS Document is developed by the project development team 3. Here we have project manager , developer, coder, tester, and users of the product who are reading our document.

1.4 Product Scope

This document covers all the requirements specification for the Sports Events. This system will be designed to minimize the user efforts of searching for the updates about sports events happening in campus,which would otherwise have to be performed manually. This will create interaction between students and teachers for queries and comments.

2.0 Overall Description

2.1 Product Perspective

In our campus, all the list of games and the information of about various games .The software being designed as a web based application mainly aimed for the use of university. The software works independently regardless of the rgukt hub and website. Here we provide log in with student id and password .The overall perspective of our website is to easily update students with event information and easy results generation along with scoreboard..

2.2 Product Functions

- Admin should be able to insert, update and delete records of users and status.
- Provide updates about all the games/events conducted in campus.
- Students and teachers can use the web interface to login and perform the desired task.
- Admin has to initiate with the sports events by adding type of sport(indoor/outdoor).
- Students will register for the interested sports activity.
- The System would provide the facility of viewing information of the companies. The

system would used to store the experiences, feedbacks.

- The System would also used to allow students communicate with others.

2.3 User Classes and Characteristics

The major User classes in the System would be :

1) **Student :**

- Student need to login with his/her username and password if they want to register for respective sport .
- Students can see winners list and can register for sports. Students can ask queries regarding any sports to respective sport's faculty.

2) **Administrators :**

- The Admin has the supreme power of the application.
- Admin provides assigning students to all facuties.
- Admin is responsible for maintaining and updating the whole system.
- Admin has the responsibility to give current notifications news.

3)**Teacher:**

- The faculty will select the students for particular sport and they will be incharge of the events regarding that sports.
- The faculty will answer the queries posted by students.
- The faculty will give the winners list to the admin and they will post the winners.

2.4 Operating Environment

- This web application can be deployed on linux or window machine with Apache Server and MySQL server.

- This application can be accessed by user through a machine having any web browser with html javascript support. The client devices must preferably have browsers like IE9 or above, Mozilla firefox (version 3.5 or above) or Opera 10 or chrome (version 29 or above) or safari installed in their OS. Specified versions are preferred to get HTML 5 output.

2.5 Design and Implementation Constraints

- User system should be connected to LAN.

- User should install any one of the above specified web browsers.

- Admin need to maintain database with security and need to follow constraints.

- The users who are accessing this website from any system or mobile must need internet connection.

2.6 Assumptions and Dependencies

- We are assuming that the user should have some basic knowledge of computer.
- Students should be from any branch or any year.

3.0 External Interface Requirements

3.1 User Interfaces

The user interface of the application will be user-friendly and good looking, intuitive and easy to use, implementing the well standards. We provide better GUI(Graphical User Interface) for easy access.

3.2 Hardware Interfaces

The hardware interface required for both users and admins are:

a. Processor (Intel i3)
b. RAM (2 GB)
c. Hard Disk (320 GB)

3.3 Software Interfaces

Operating Systems: Linux, Windows etc
Database : Mysql
Languages required: HTML, CSS, Java script, PHP

3.4 Communications Interfaces

Web browser, network servers communications protocols , HTTP Protocols.

4.0 System Features

- *Our application will provide security for users by providing log in and registration. It consists of username, password and they can register for required sports.*
- *It is centralized and maintained by admin. Stakeholders at every level of the organization can access data/reports assigned to them with permissions.*
- *If the user credentials are valid and if they follow all the constraints of our applications then the user can get good service from this application .*

- *MAIN MODULES:*

1. *Sports events registration*
2. *Assigning teachers for events*
3. *Selection procedure for registered students*
4. *Declaration of results*

5. *Asking queries*
6. *Providing updates*

5.0 Other Nonfunctional Requirements

5.1 Performance Requirements

a. The system shall function in real-time any operation on the stored information, shall complete within few seconds.
b. The system shall allow simultaneous use by several users , without data corruption and data interruption.
c. The system would exhibit high performance to users because it would be well optimized.
d. The system will reduce the use of manual process of conducting events.

5.2 Safety Requirements

The database may get crashed and corrupted at any certain point time because of virus or operating system failure. Hence, it is required to take the database backup.

5.3 Security Requirements

- *The users who are login into the web page and the user submitted their details should belongs to the one user only.*
- *Admin can be able to update , create, delete the records as required.*
- *We are developing secured database for users and create a user-friendly environments.*
- *We are having different stakeholders.*

5.4 Software Quality Attributes

- *The software system could provide automatically generated backup (on external hard drives) containing all the stored information at the time the backup is taken.*
- *The quality of the database is maintained in such a way that it will be user- friendly to all the users*

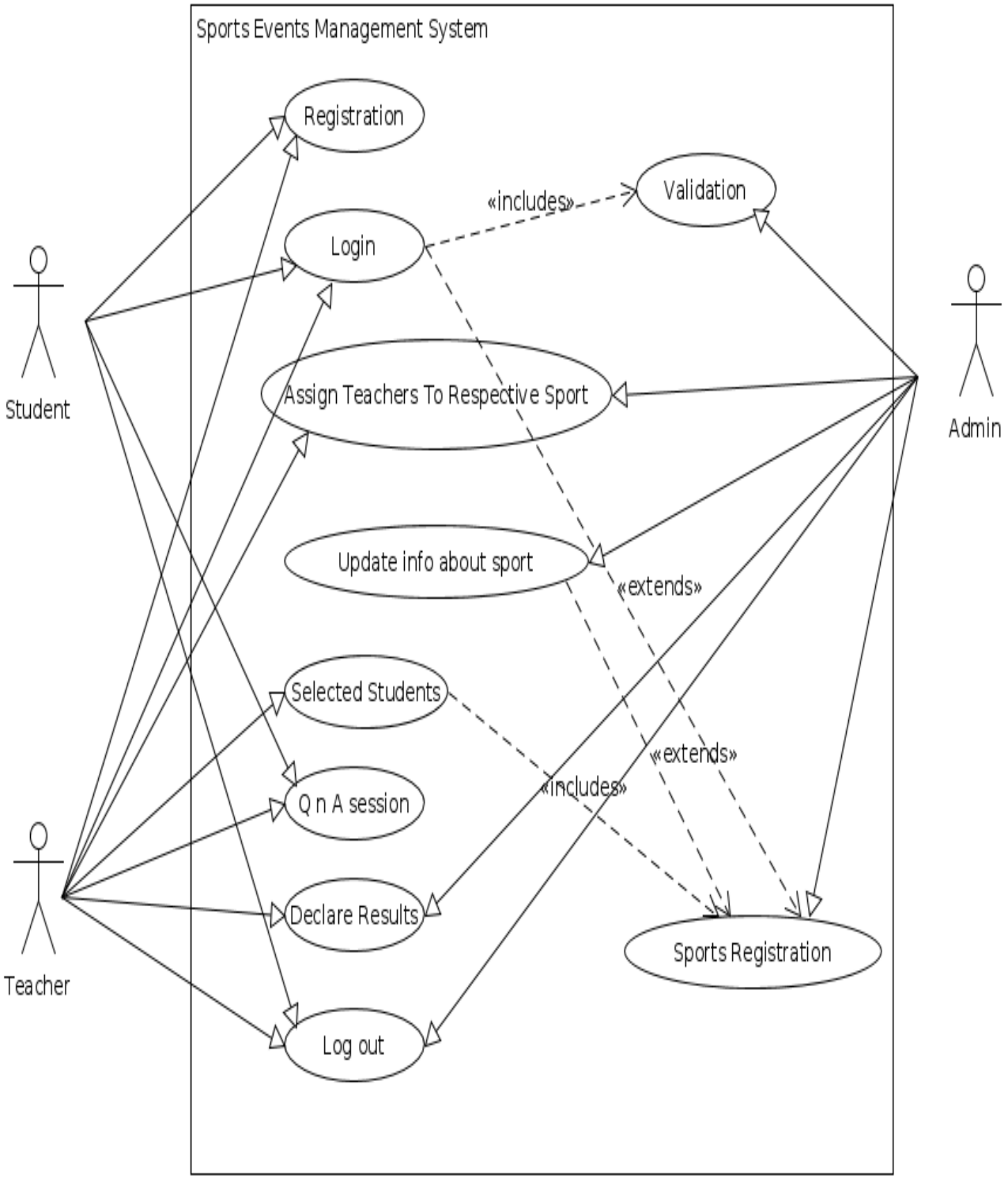
5.5 Business Rules

A business rule is anything that captures and implements business policies and practices. It includes the cost of the project. The users can only access the data and admin will provide the required credentials and constraints for the users . The user will avoid illegal rules and protocols.
No one should cross their authenticated rules for users and admins.

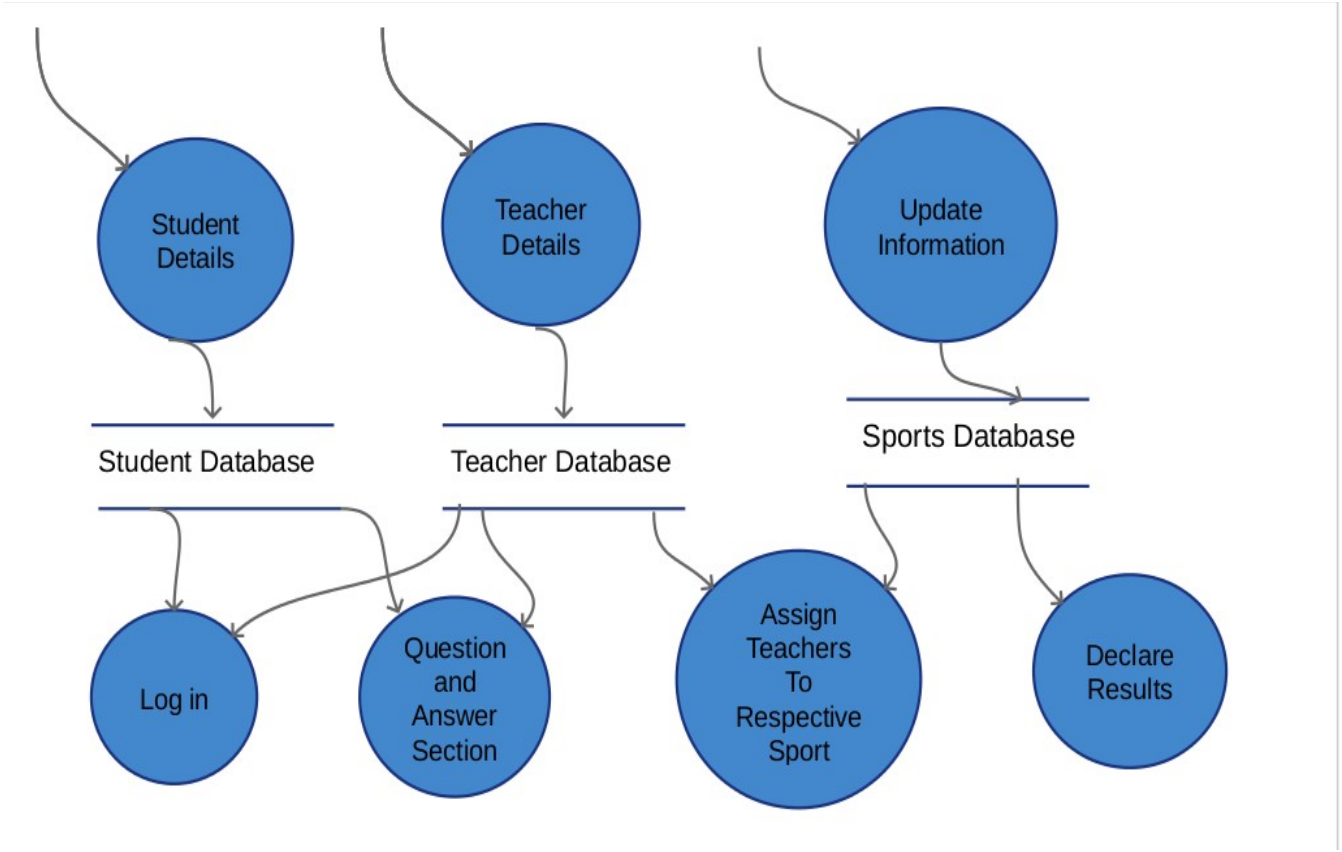
6.0 Other Requirements

Other requirements includes -maintaining website
-sports related database
-maintaining user information
-update the upcoming events
-solving the problems raised in queries

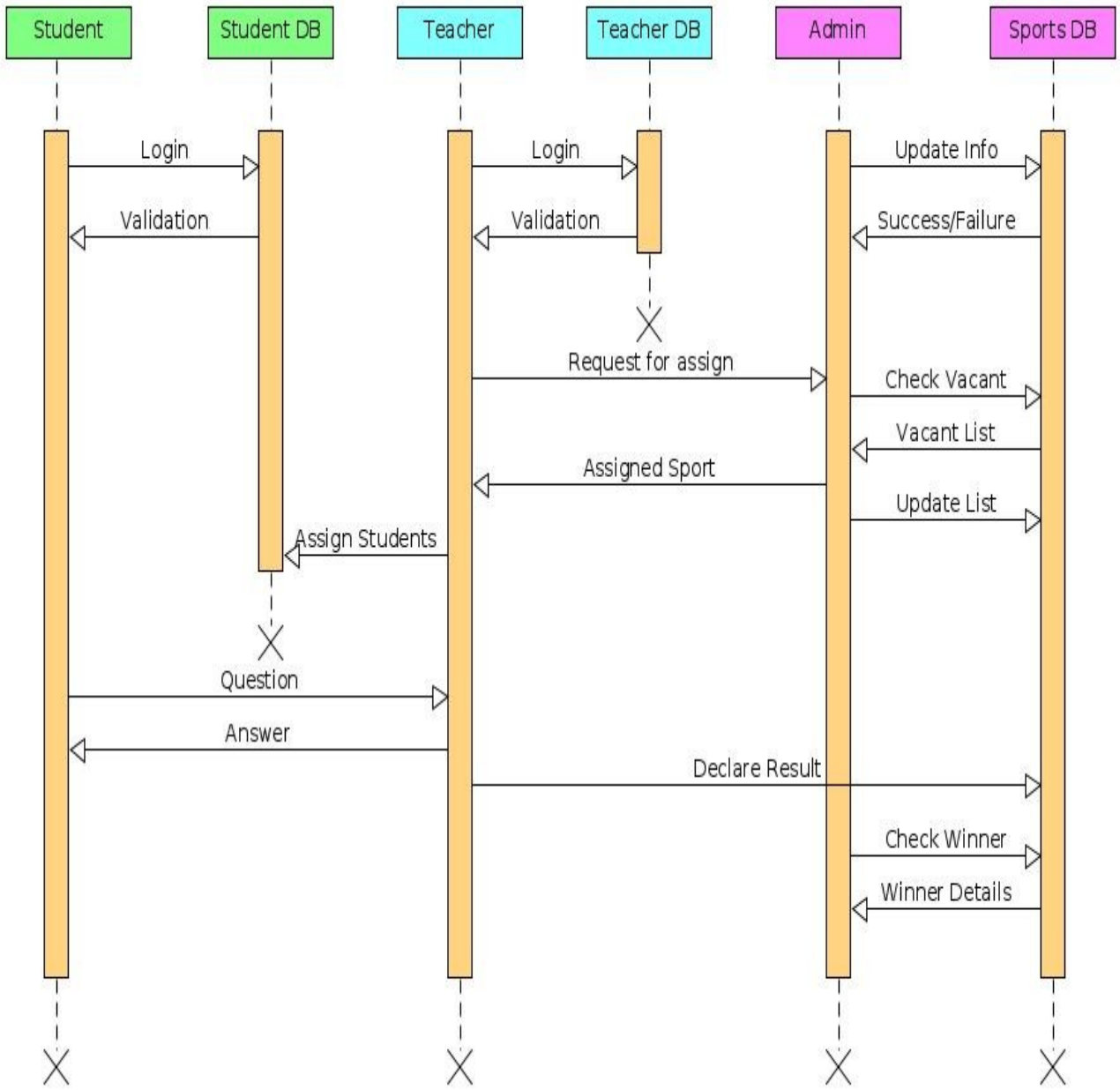
7. Usecase Diagram



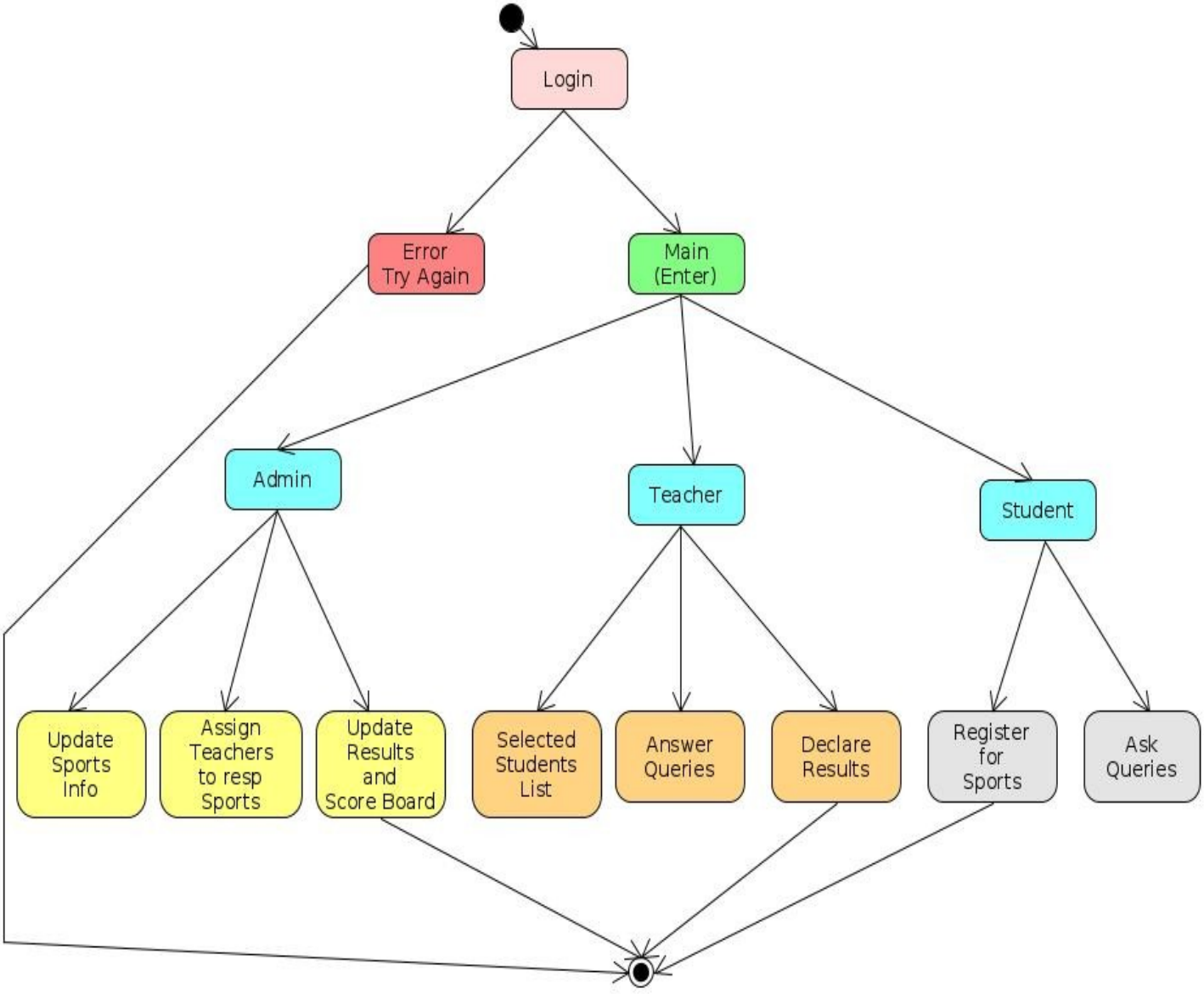
8. Dataflow Diagram



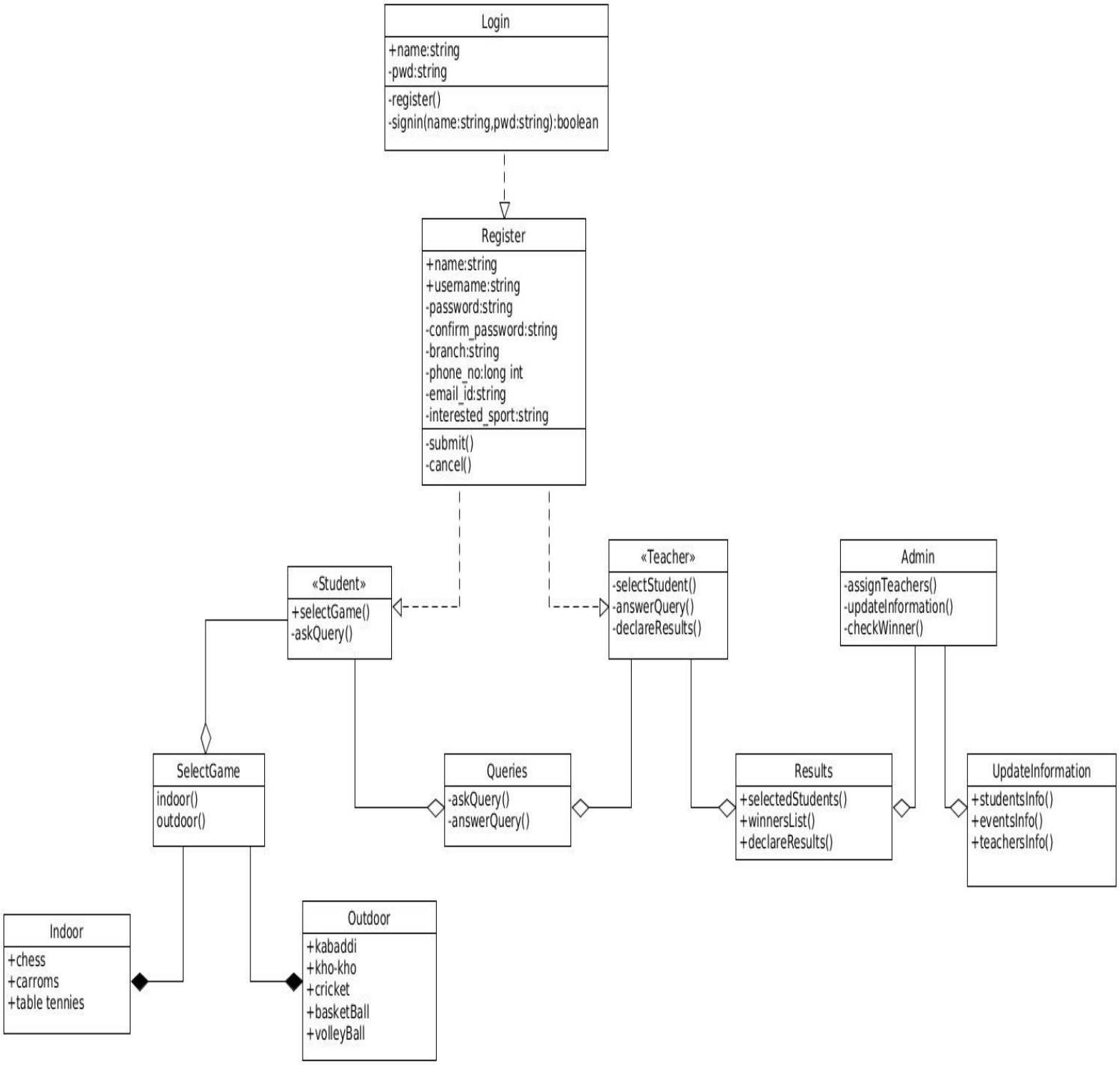
9. Sequential Diagram



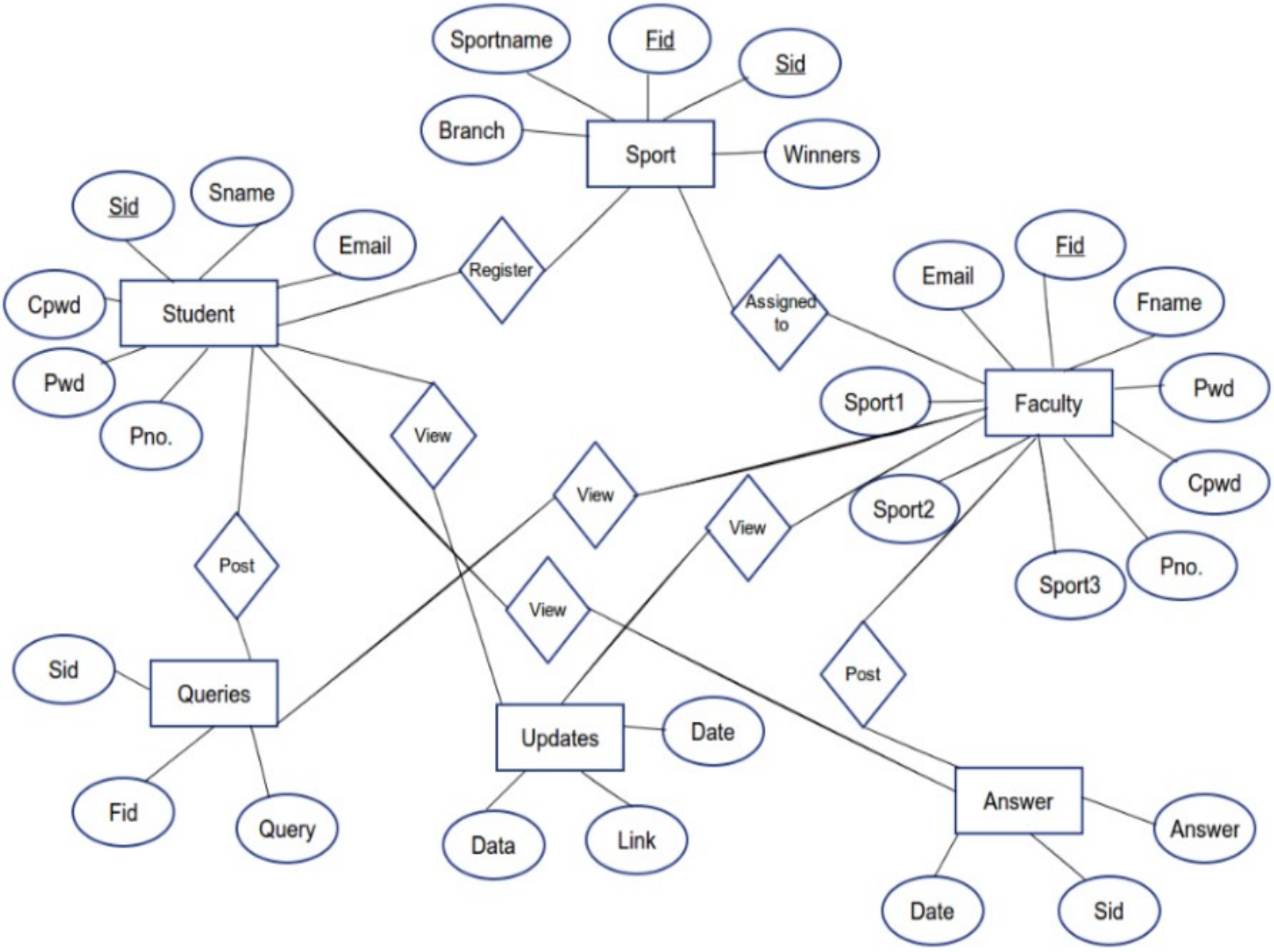
10. State chart Diagram



11. Class Diagram



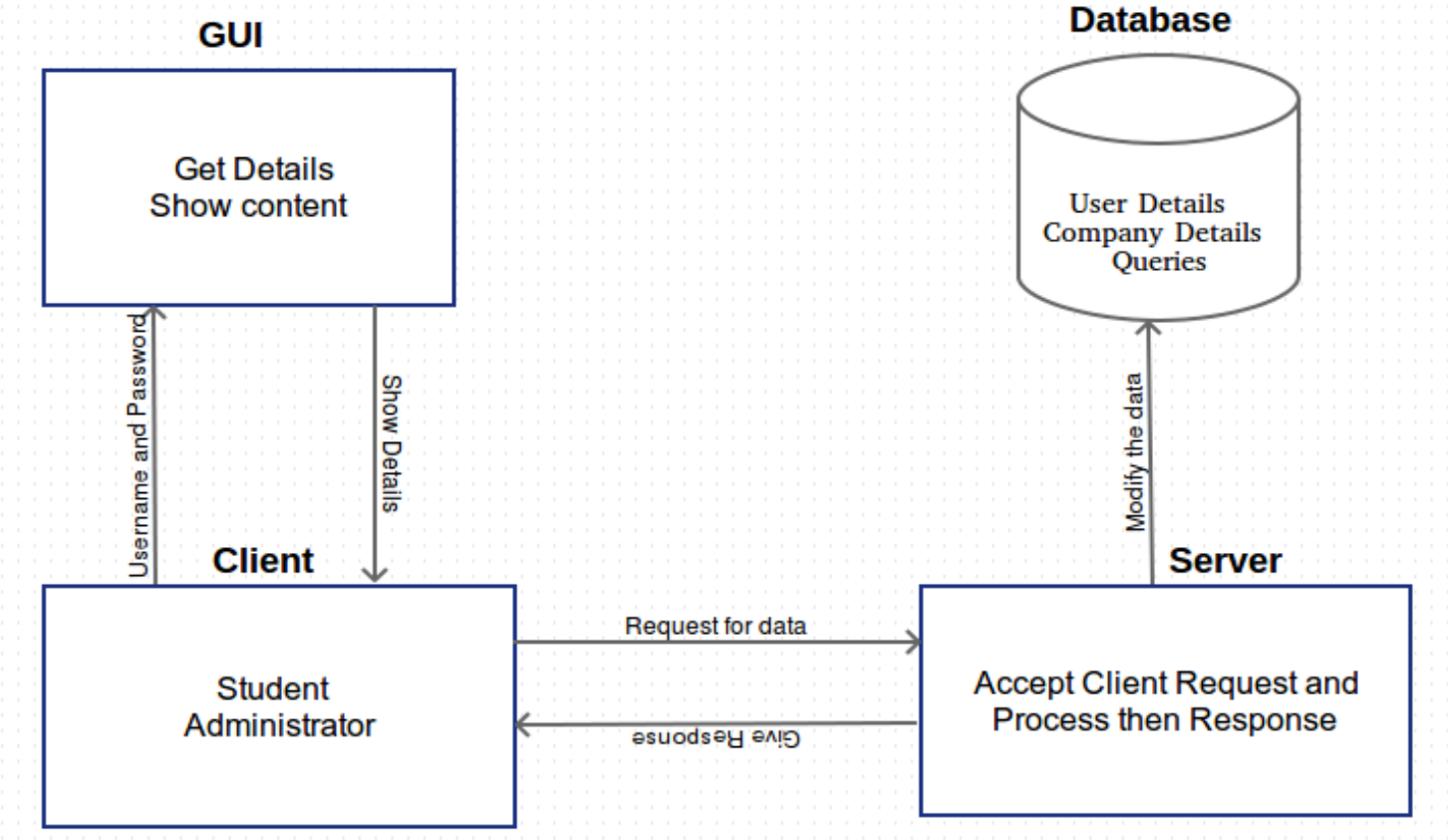
12. Entity Relationship Diagram



13.0 Architectural and Component-Level Design

Describes the overall system architecture and component partitioning in general terms. Include a rationale for the partitioning choices.

13.1. Architecture diagram

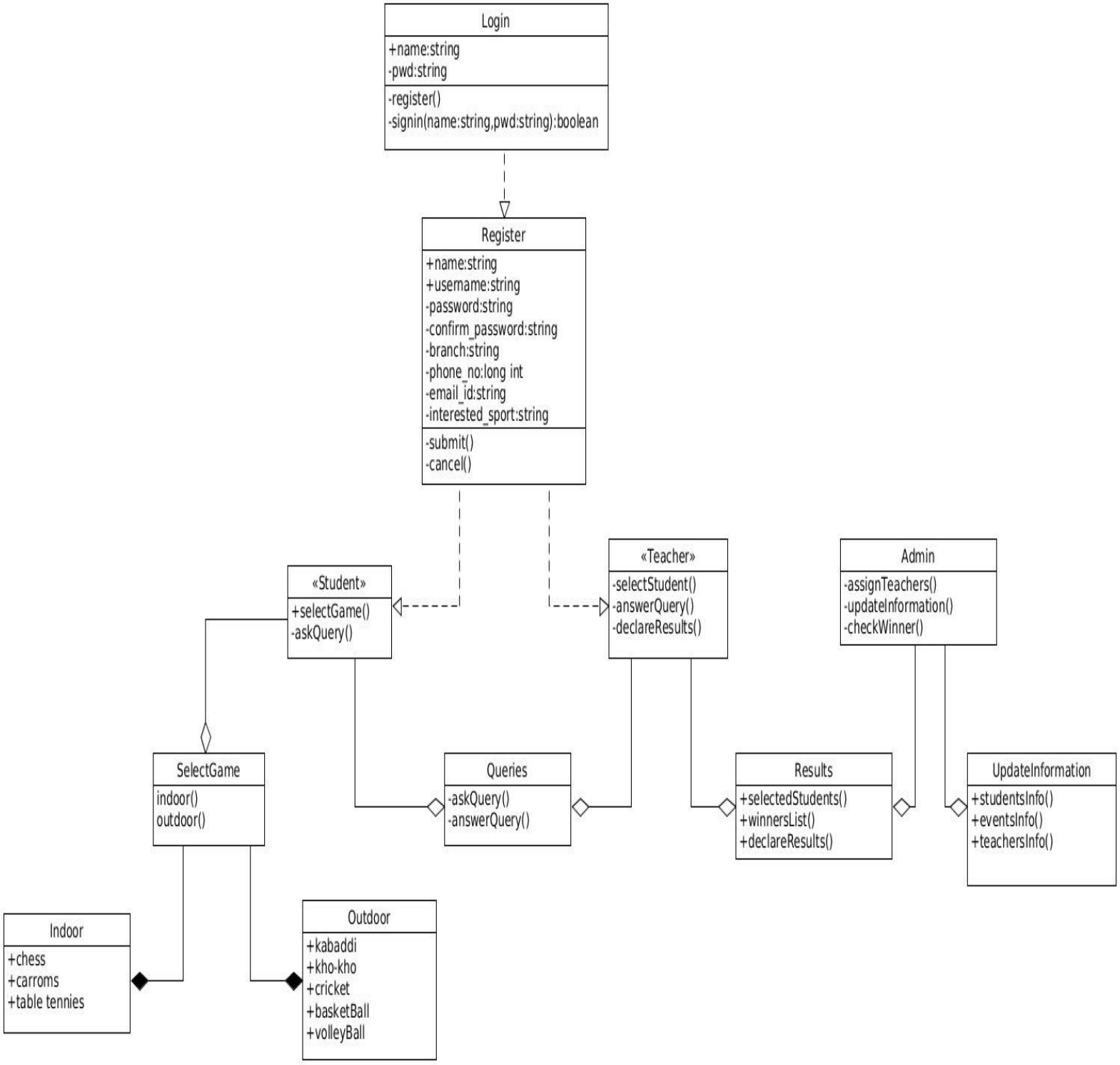


13.2 Description of Components

We divided our project into 4 modules. Those are

1. Graphical User Interface
2. Database
3. Server
4. Client

13.2.1 Class Diagram :



Class Description:

- **Student :** It contains basic information of the Student Details.
- **Login :** It deals about the authentication of user.
- **Admin :**Admin will assign faculty to the particular sport.
- **Faculty:** Faculty will declare the sport result.
- **Queries :** It contains queries and answers.

13.2.2 Home

This componet describes layouts like home page that is nothing but here a login page,feedback on our website..etc. This can be showed to student, faculty, and administrator.

HOME PAGE :

Methods:

Precondition : Student, faculty, and admin will see this page can select tabs present on the top.

Postcondition : After selecting a tab student, faculty, and admin will directed to student login page, faculty login page, and admin login page respectively.

Algorithm :

```
check = login(username, password)
if check is true
        return “front page”
else
        return “either username or password is invalid”
```

Error handling : User details are checked with database whether the user is registered or not.

13.2.3 Student

This component describes about all student classes.

Class 1 : Login

Attributes:

1.Sid
2.Password
3.Submit
4.Register

Methods:

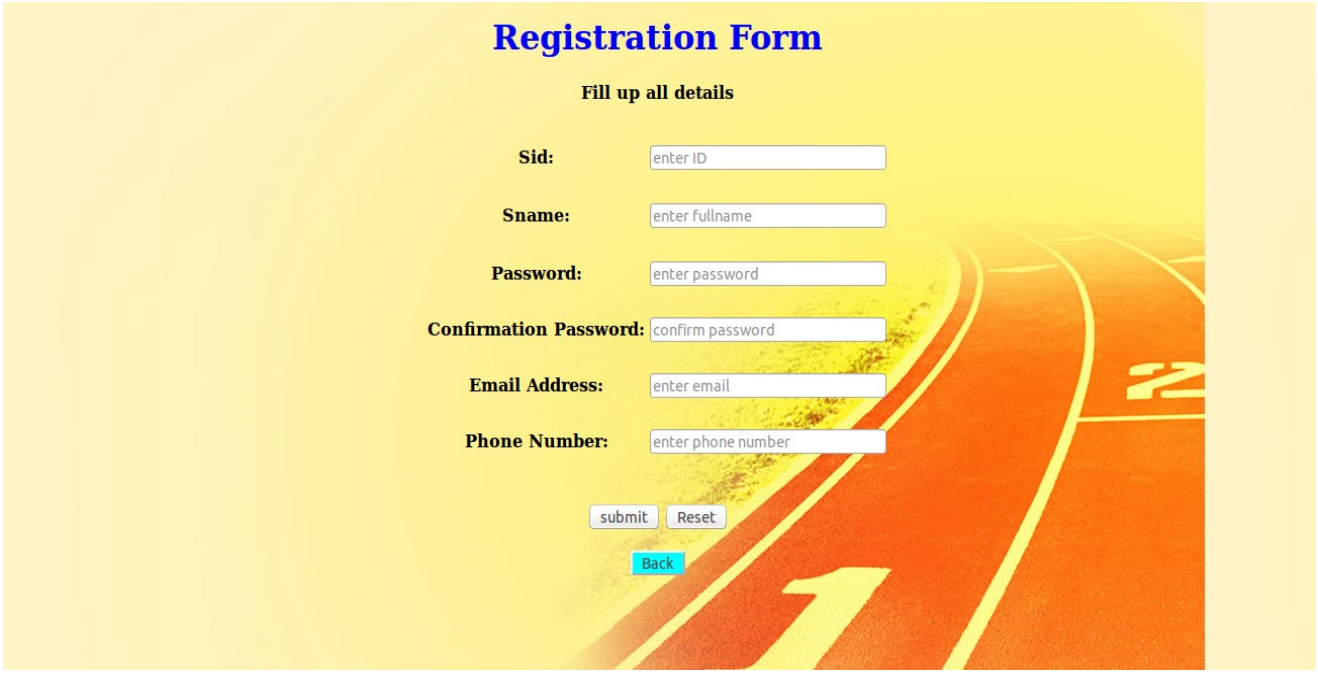
Precondition : User should be logged in and he should select a company.

Postcondition : Student will directed to the student home page.

Algorithm : Student click the login button, Student credentials are checked with the student database details.

Error checking : Student credentials are checked with the student database details if details not incorrect it will be in login page.

Student's Registration :



Attributes:

1. Sid
2. Sname
3. Password
4. Conform password
5. Email Address
6. Phone number

Methods:

Precondition : Student should click on the register button present in login page.

Postcondition : After filling all the details student should submit the form.

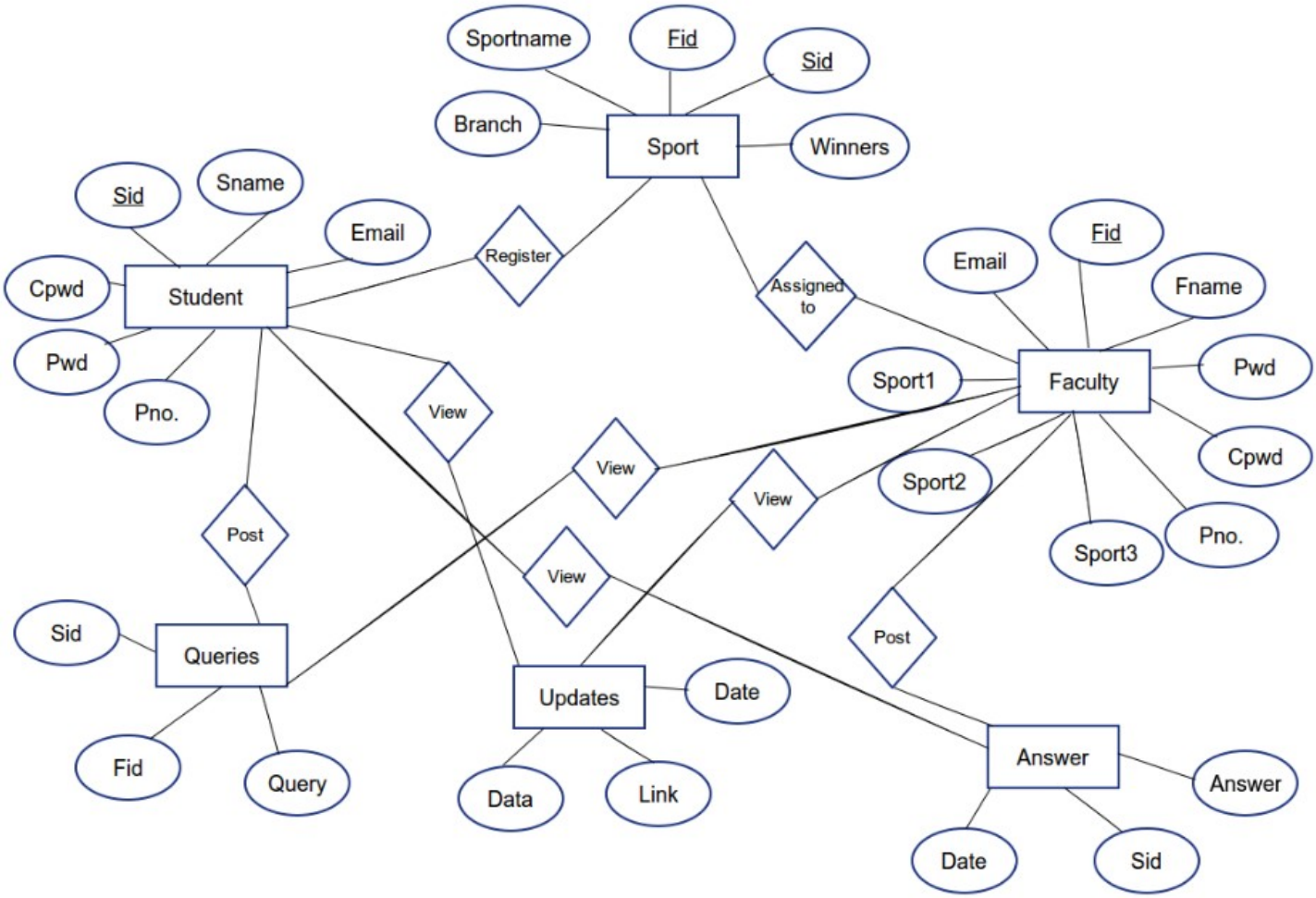
Algorithm : Student details are inserted into student database if all are correct.

Error Checking : Student details are checked with all validations.

14. Data Architecture

Data collected from users are stored in the database in the form of tables with which we manipulate corresponding records.

14.1. Entity-Relation Diagram



14.2. Data Dictionary

Table Name:Student

Field Name	Field Type	Other information
Sid	Varchar	Primary key: Size:10
Sname	Varchar	Composite Attribute Size:20
Pwd	Varchar	Size:20
Cpwd	Varchar	Size:20
Email Id	Varchar	Multivalued Attribute Size:20
Phn no.	Varchar	Multivalued Attribute Size:12

Table name:Faculty

Field Name	Field Type	Other information
Fid	Varchar	Primary key Size:10
Fname	Varchar	Composite Attribute Size:20
Pwd	Varchar	Size:20
Cpwd	Varchar	Size:20
Email Id	Varchar	Multivalued Attribute Size:20
Phn no.	Varchar	Multivalued Attribute Size:12
Sport1	Varchar	Singlevalued Attribute Size:20
Sport2	Varchar	Singlevalued Attribute Size:20
Sport3	Varchar	Singlevalued Attribute Size:20

Table Name:Sport

Field Name	Field Type	Other information
Sportname	Varchar	Singlevalued Attribute Size:20
Fid	Varchar	Primary key Size:10
Sid	Varchar	Primary key Size:10
Branch	Varchar	Singlevalued Attribute Size:10
Winners	Integer	Size:10

Table Name:Queries

Field Name	Field Type	Other information
Sid	Varchar	Primary key Size:20
Fid	Varchar	Primary key Size:20
Query	Varchar	Size:100

15 Client-Server Module Description

15.1 Client and Server Module:

The client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests. Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.

15.2 Client and Server Communication:

A service is an abstraction of computer resources and a client does not have to be concerned with how the server performs while fulfilling the request and delivering the response. The client only has to understand the response based on the well-known application protocol, i.e. the content and the formatting of the data for the requested service.

Clients and servers exchange messages in a request–response messaging pattern. The client sends a request, and the server returns a response. This exchange of messages is an example of inter-process communication. To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in a communications protocol. All client-server protocols operate in the application layer. The application layer protocol defines the basic patterns of the dialogue. To formalize the data exchange even further, the server may implement an application programming interface (API). The API is an abstraction layer for accessing a service. By restricting communication to a specific content format, it facilitates parsing. By abstracting access, it facilitates cross-platform data exchange.

A server may receive requests from many distinct clients in a short period of time. A computer can only perform a limited number of tasks at any moment, and relies on a scheduling system to prioritize incoming requests from clients to accommodate them. To prevent abuse and maximize availability, server software may limit the availability to clients. Denial of service attacks are designed to exploit a server's obligation to process requests by overloading it with excessive request rates.

15.3 Server Side Algorithms :

Login:

login(username, password)
make a connection with the mysql
select the database
select the Users table
if username is existed in the table
if password is matched
return “true”
else
return “Password is invalid”
else
return “Username is invalid”

View Updates :

make a connection with the mysql
select the database
select the updates table
display all the updates along with date

Post a query :

checkQuery(query)
make a connection with the mysql
select the database
select the Queries table
check if fid is in the faculty database and sid is in the student databse
insert the query into table
else
return “false”

Post an answer :

checkAnswer(answer)
make a connection with the mysql
select the database
select the Queries table
verify the answer
check fid is in faculty database and sid is in student database
insert the answer into table
return “true”
else
return “false”

Assign Teacher:

```
make a connection with the mysql
select the database
select the faculty and sport table
check if fid=null in sport table
        then select sport preferences from faculty table
        then check if fid is in sport table or not
        if not then assign that fid to that sport
else
        return “false”
```

Display sports :

```
make a connection with the mysql
select the database
select the faculty and sport table
check whether fid is assigned or not
        if not display error
else
        select sport details from sport table
        compare fid in sport with fid in faculty
                then get values of faculty details from faculty table
        display all details
```

15.4 Client Side Algorithms :

Login:

```
get username
get password
if username is invalid  // null or less size or not in proper format
        return “Enter valid username”
if password is invalid  // null or less size
        return “Enter valid password”
else
        login(username, password)
```

Post a query :

```
get query
if sid,fid,query is invalid // null
        return “Enter valid details”
else
        checkQuery(query)
```

Post an answer :

get answer
if fid,sid is invalid // null
return “Enter valid details”
else
checkAnswer(answer)

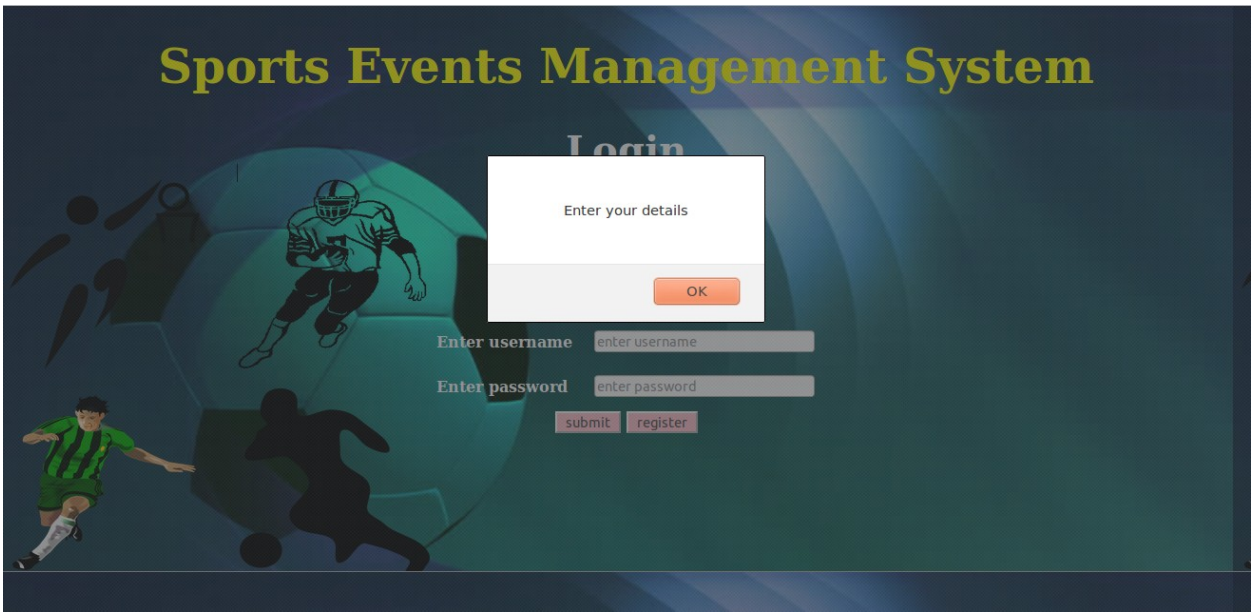
Admin Login:

Admin will login with his/her credentials(Username and Password)
if username and password is true
goto assigntea.html
else
return Invalid login

Sport Registration:

after logged into the student home page
select sports registration tab
enter student Id, branch, and interested sport
then click on submit
if student id is valid (check with student database)
then that student registered successfully.

16.Test Cases:



Description:

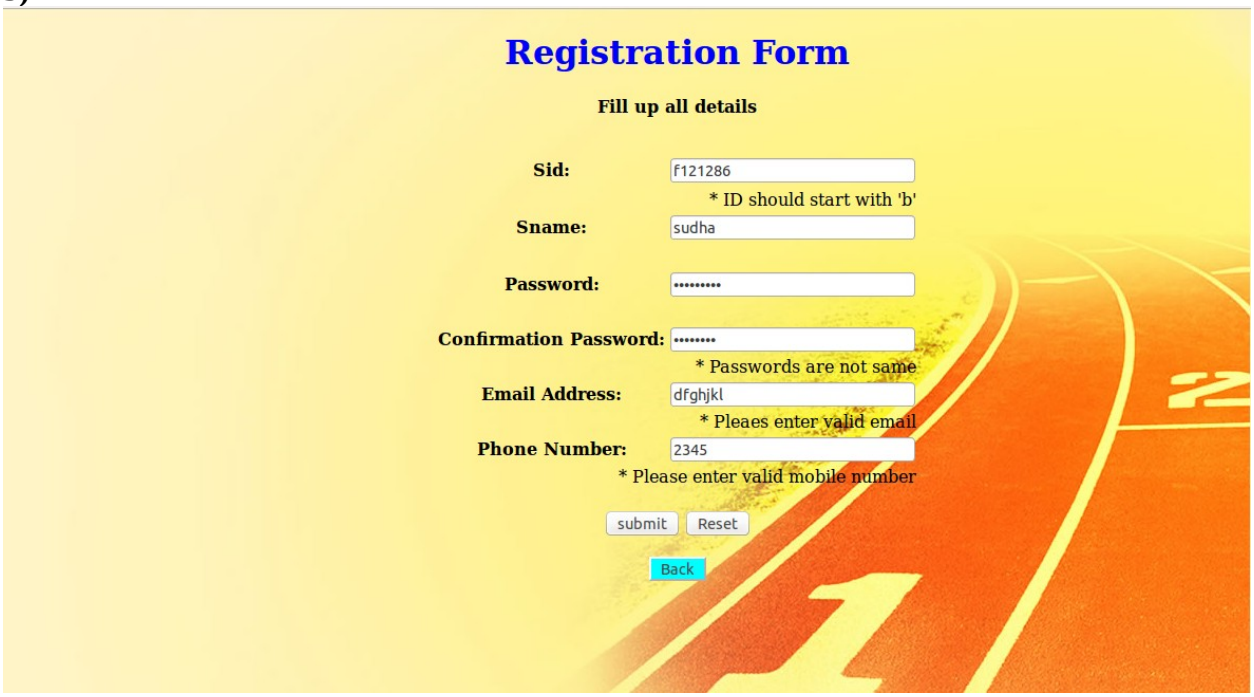
This window shows that when any person dont enter their username and password and click on “submit” then - “enter your details” windows will appear on the screen.
So, one need to fill up the valid details and enter the values.

2)

Description:

In this part, one has to register to the website to get log in.So, one should not click on to "submit" untill that person fill up all the details given in the registration form. So,if we click on ok even if we dont fill one field then "enter your details" will appear on the window.

3)



Description:

The registration form will have validations which one need to follow. These are to be followed:

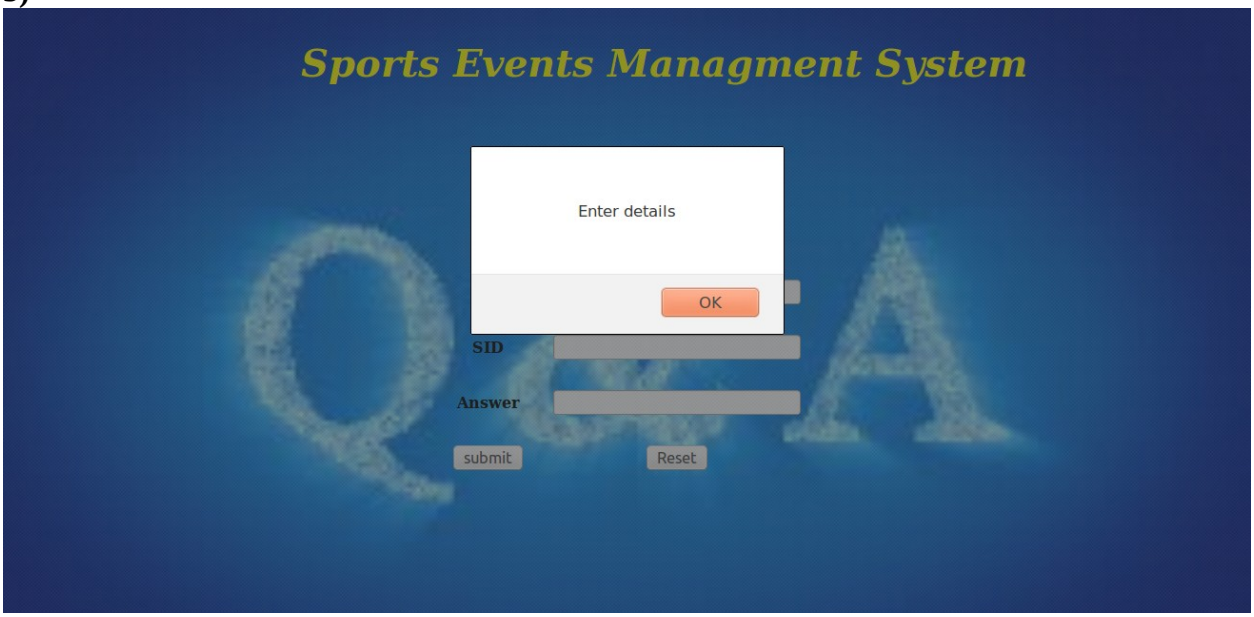
1)Person need to give student id starting with "B".
2)Person need to give password more than 8 characters.
3)One need to follow the validation of correct valid phone no.

4)

Description:

The query section may arise some problems like if any student want to raise any query should give correct valid student id and faculty id, then only they will come up.

5)



Description:

This problem will show only when the student will not give any details regarding the queries.