Design

Design in system development is basically the transformation of user requirements into the big pictures embedded with functionalities and operations. In this phase the logical and physical design are described in detail through different modelling and process diagrams, annotations and other documentations. The output of this phase will outline the new system as a collection of modules or subsystems.

For designing the sports management system, I have decided to undertake the following tasks.

* Establishing the relationship and class diagram between the classes.
* Modeling the functionalities and performances of system and users in terms of diagram.
* Prototyping the interfaces.

1. Structural model
   1. Final class diagram

Class diagram is the static view of an application which does not only visualize, describe and document the system but also constructs the executable code for developing the software application. Class diagram shows the collection of classes, interfaces and applied associations and constraints.

The class diagram has been constructed due to the following reasons:

* Analyze and design ta static view of an application.
* Describe the functionalities performed by the system.
* Forward and reverse engineering.

1.2 Data flow diagram

Data flow diagram(DFD) are used to identify, model and document how the data moves around, enters and leaves the system, what changes the data and where is it stored simultaneously. Basically, DFD are used to illustrate the data into the system, within the system and out of the system.

DFD has been built for sports management system due to the following reason:

* It is simple to create and easy to understand.
* It is used to illustrate the system boundary.
* Complex system can be illustrated to different levels of details.

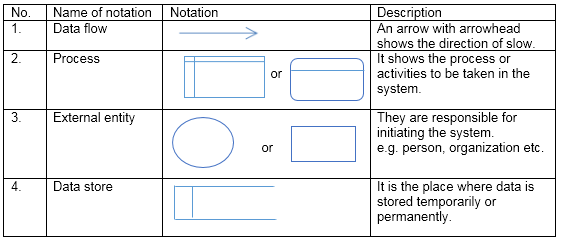


Table1: Notation for DFD

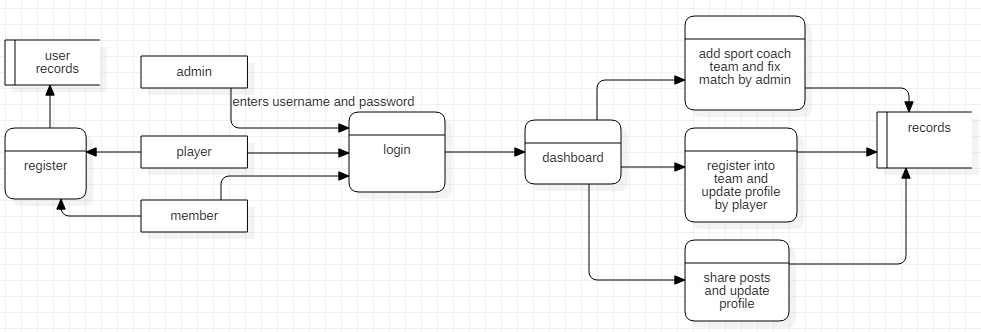


Fig1: Zero (0) level DFD

In the above diagram, the whole overview of the sports management system has been illustrated. The interaction of different users performing different functionalities has been shown in simple structure rather than in a detail. This DFD level shows, how the user registers, adds and manipulate the data in a distinct flow.

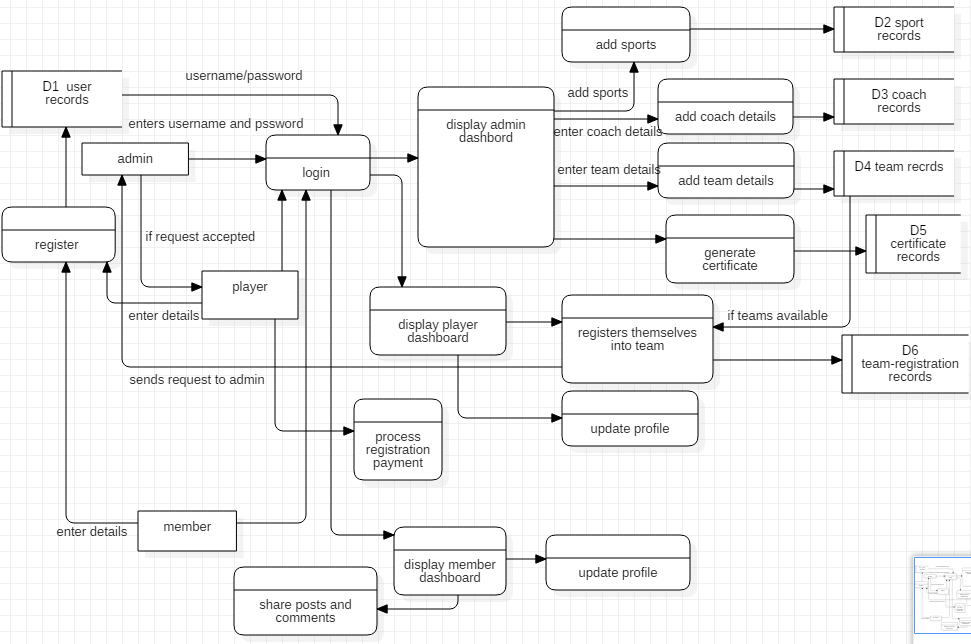


Fig2: First (1st) level DFD

The processes from zero level has been broken down into the sub-processes mentioned with sources of information. The admin adding sports, coach, team details and fixing the match and scheduling along with issuing certificates has been shown whereas the players registering themselves in the team is shown the above diagram. Similarly, the members sharing their posts has also been shown by the diagram. the whole processes are embedded in a particular flow and stores the data into the data store simultaneously.

1. Behavioral model

2.1 Activity diagram

Activity diagram is similar to UML diagram which is suitable for modeling the activity flow of the system. It is basically used to illustrate the control flow from one activity to another.

(<https://www.tutorialspoint.com/uml/uml_activity_diagram.htm>)

Activity diagram can also be useful in following ways:

* Model the business requirements
* High level understanding of the system’s functionalities.
* Investigating business requirements at a later stage.

The notation that has been used for representing the diagram is given below:

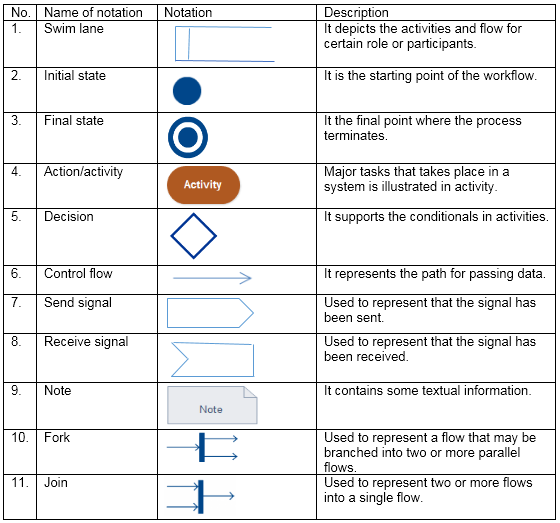


Table2: Notation for activity diagram

For representing the model between different activities, I have used StarUML.

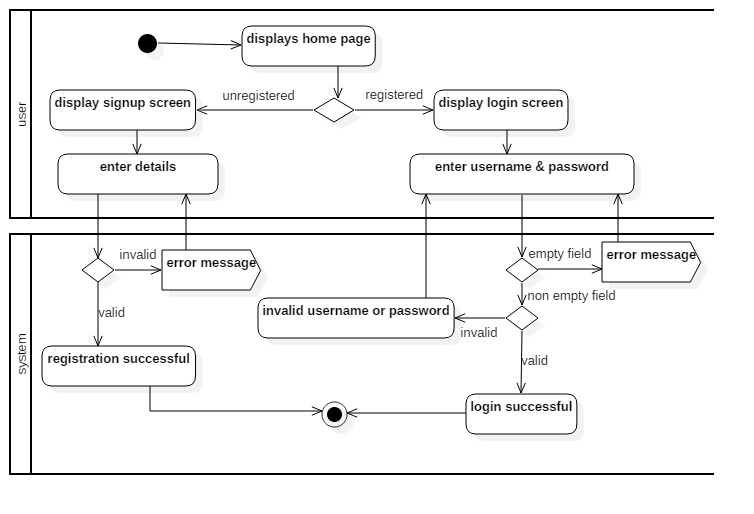


Fig3: Activity diagram for login and registration for users

Description:

The activity performed by the users for login and sign up has been illustrated in the above diagram. Initially, the user visits the page and should login or signup for the further performance. The user needs to provide the username and password for logging into the system and is only successful if it’s valid and non-empty. Similarly, if the user doesn’t possess credentials for login, he/she needs to register first with valid information.

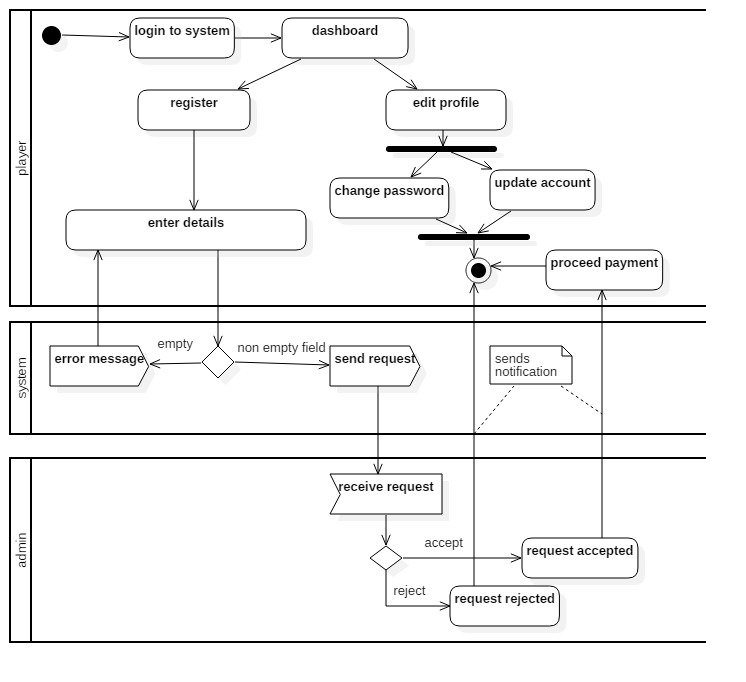


Fig4: Activity diagram for registering team by players

Description:

In this activity, the process for registering players into the team has been presented. Firstly, the player logs into the system and the dashboard is displayed where he can either update his profile or perform the registration. For the registration, he needs to provide the details without leaving any fields empty otherwise the error message will be displayed. Then the user sends request to the administrator where he either accepts or declines the request. If the request is accepted, the user gets notified to proceed the payment for team registration otherwise the user will be notified by the rejection message and the process is terminated.

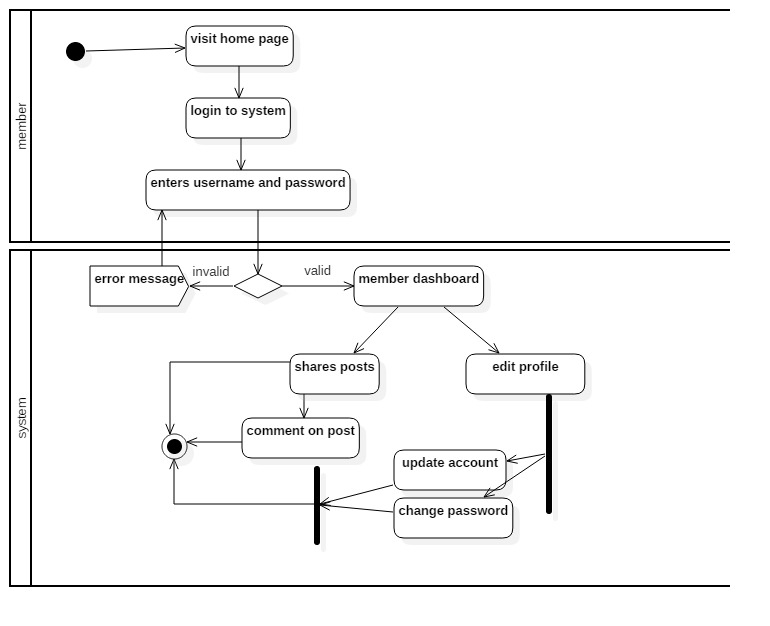


Fig5: Activity diagram for online forum

Description:

In the above diagram, the activity performed by the member for online discussion has been visualized. Initially, the member visits the page and logs into the system where the provided login credentials are verified. If the information is valid, the dashboard will be displayed where otherwise the error message will be displayed. the member can update their profile and share the posts and comments through dashboard and the process is terminated simultaneously.

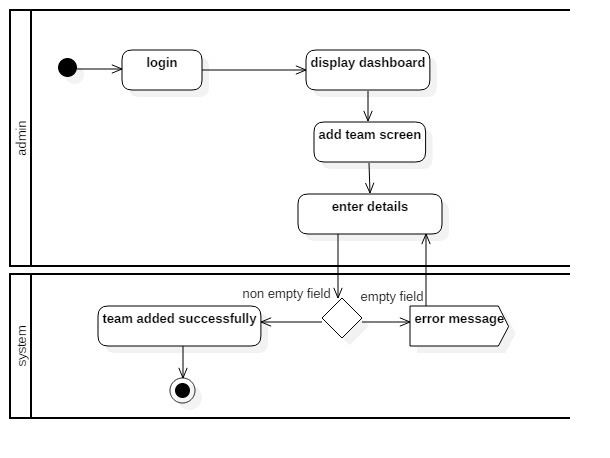


Fig6: Activity diagram for adding team

Description:

The activity performed by the admin for adding the team details has been shown in the above diagram. As the login becomes successful, the dashboard for admin will be displayed and the admin will add the details via screen for adding team. if the entries re valid without leaving any field empty, the team will be added successfully followed by the termination of process otherwise the error message will appear. The above diagram explains the particular flow of activities that can be useful for describing the process in an executable system.

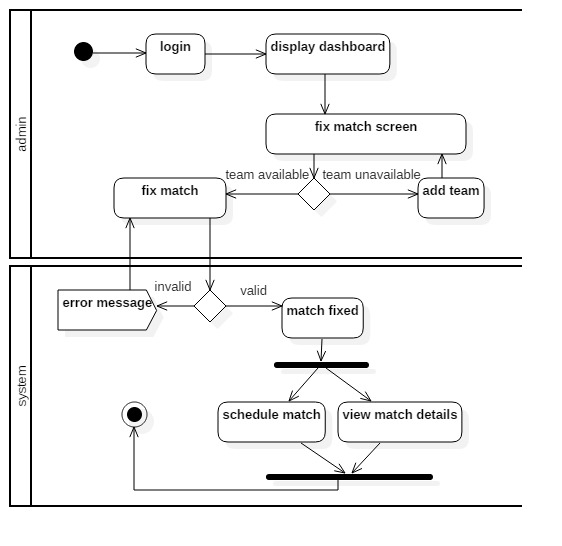


Fig7: Activity diagram for fixing match and schedule

Description:

The above diagram describes the flow between fixing the match and scheduling performed by admin in sports management system. Initially, the admin has to login to the system and displays the dashboard where he can fix the match between the teams. Before fixing the match successfully, the availability of teams is decided where the match fixes successfully if the teams are available and valid otherwise the team is added first and is then proceed. Similarly, the details of matches are viewed and scheduled which is demonstrate by using fork and the join is used for terminating the process simultaneously.

2.2 Sequence diagram

Sequence diagram are the interactive diagram that details how the operations are carried out between the objects in a sequence of time represented by the vertical axis.

For any system, sequence diagram can be useful in following ways:

* Represent the details of use case.
* Model the logic of sophisticated function or operation.
* Detailed understanding of how objects interact with each other.

Notation

The notations that has been used while establishing the sequence diagram are tabulated below:

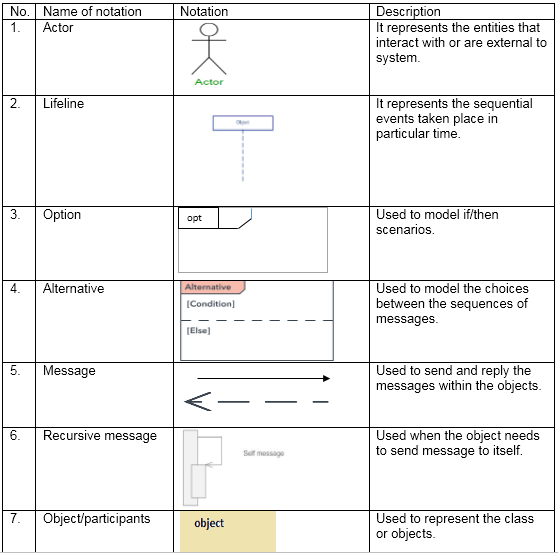


Table3: Notation for sequence diagram

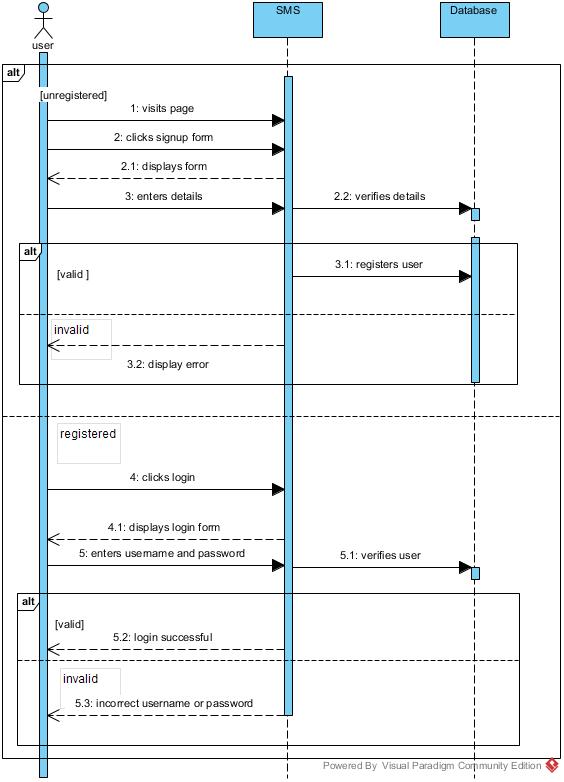


Fig8: Sequence diagram for login and signup

The sequence diagram in the above diagram shows the interaction of users and login and registering procedure in ***sports management system*** ***(SMS*).** It consists of mainly three parts: the actor represented by user, two entity objects as SMS and database and the data route between them. The diagram illustrates the sequence of login of signup performed by the users in a particular sequence of time respectively.

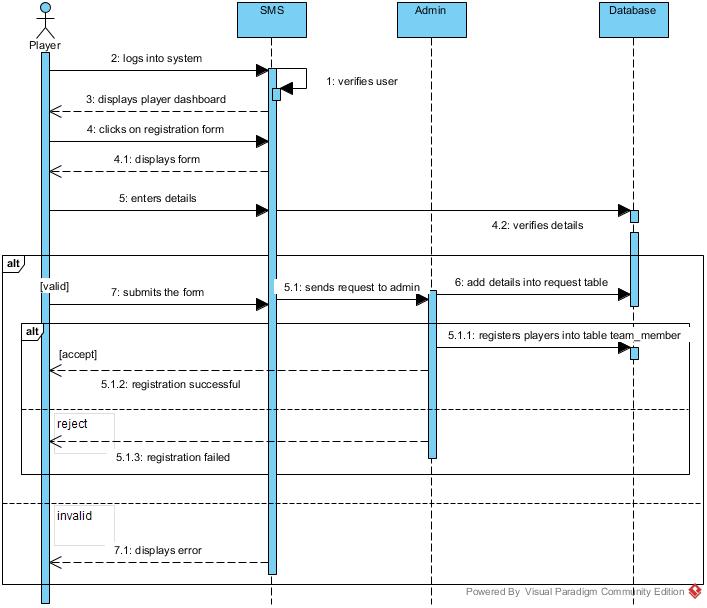


Fig: Sequence diagram for registering into team.

The above sequence diagram shows the process of registering into team by the players. It shows the interaction between the system, admin and players how request are sent and received and what happens when the request is accepted or declined.

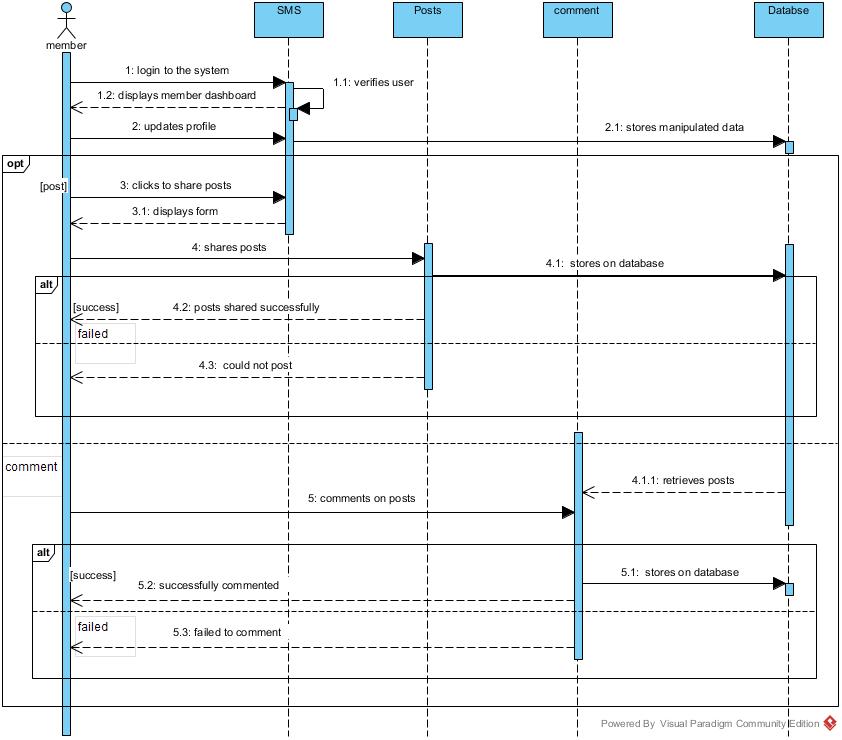


Fig10: Sequence diagram for online forum discussion

The above sequence diagram illustrates the process for online forum discussion performed by the member. The member can either share the posts or comment on posts which are represented by the diagram in detail. It explains the success and termination of process in a sequence of time.

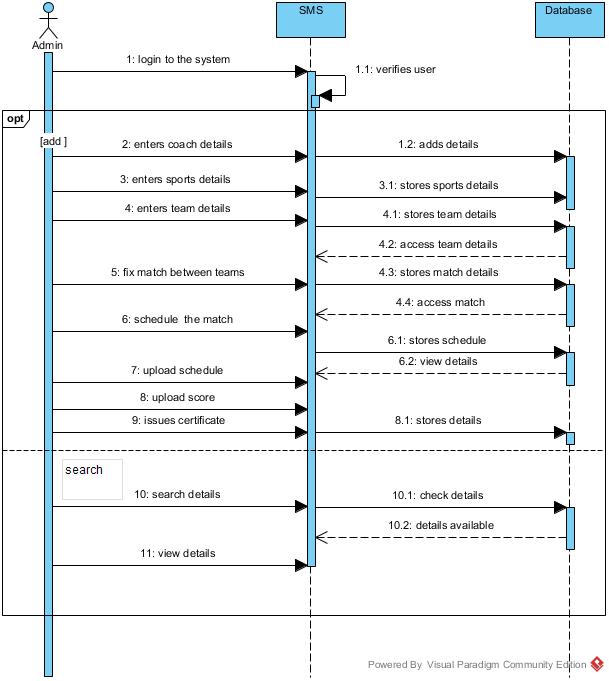


Fig11: Sequence diagram for admin performance.

The above sequence diagram shows the activities performed by admin. the activities like adding details of sports, coach, team and fixing match along with scheduling followed by issuing certificate is demonstrated in the above diagram. The diagram also illustrates the process of searching details respectively.

1. Database model

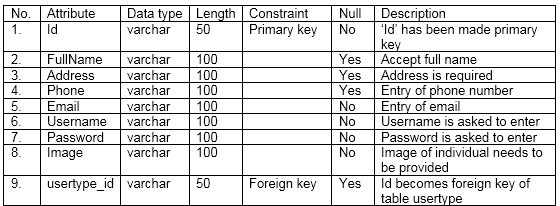
3.1 Data dictionary

Data dictionary generally contains the metadata i.e. the data of the data. It conveys the information like table name, details of the tables and their respective constraints.

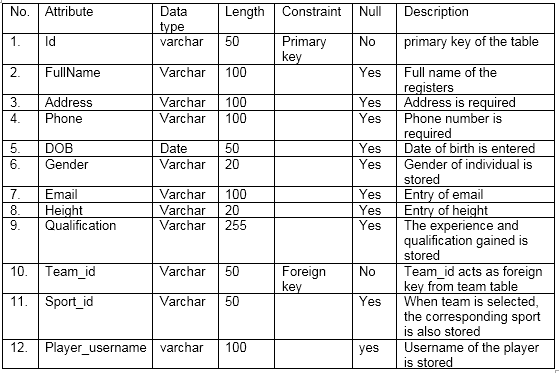
For developing any program, data dictionary can be useful to understand where the data fits in the structure, what might be the values and how they are interrelated.

The data dictionary for sports management system is modelled as follows:

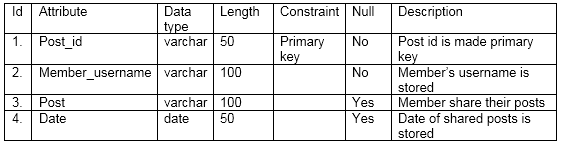
* Data dictionary for player/member



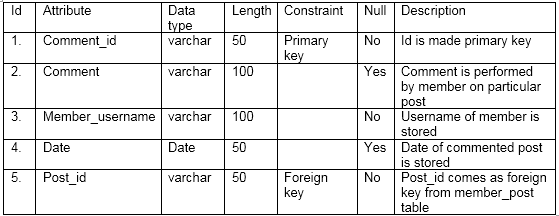
* Data dictionary for Team\_register



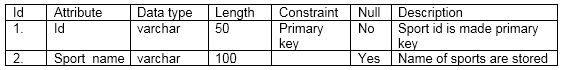
* Data dictionary for member\_ post



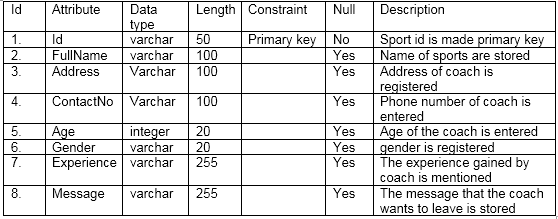
* Data dictionary of comment



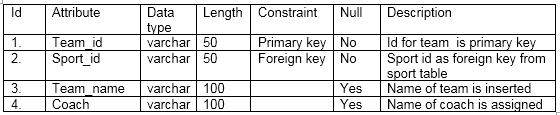
* Data dictionary for sport table



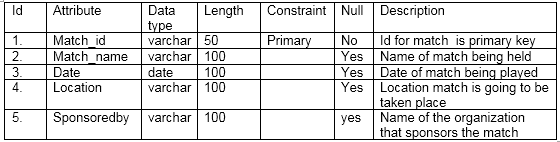
* Data dictionary for coach table



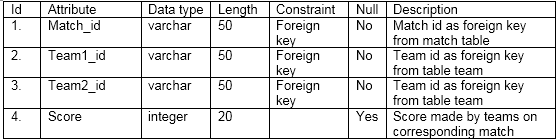
* Data dictionary for team table



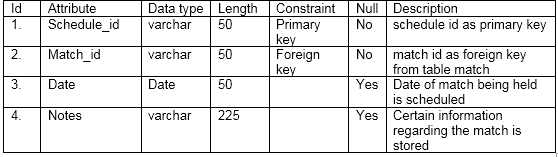
* Data dictionary for match table



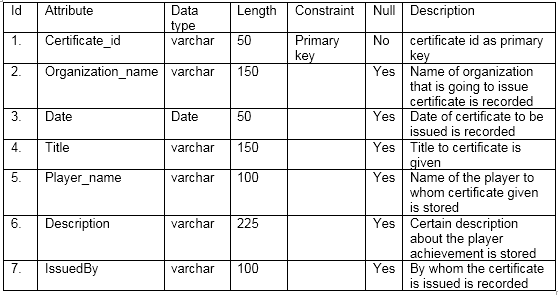
* Data dictionary for result table



* Data dictionary for schedule table



* Data dictionary for certificate table



3.2 ER Diagram

ER (Entity Relationship) Diagram is the graphical representation of a system that depicts the relationship among the people, objects or the system. Basically, an ER diagram is a data modelling technique for defining the business process and relational database.

ER diagram provides the visual initiation for the database design and can also be used to determine the requirements throughout the system. Beside this, ER diagram can be useful in following ways.

1. ER diagram can be served as the reference for debugging or re-engineering the business process if required in future.
2. ER diagram can be used for organizing the relational databases used in business process.
3. It can be useful for troubleshooting the problems in logic or deployment by verifying the database where the wrong is.
4. ER diagram can be useful for planning the data structures and setting up the useful databases in the near future.

Notation

The notation that are used in representing ER diagram is as follows:

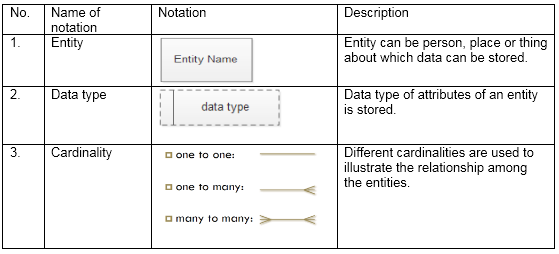


Table4: Notation for ER diagram.

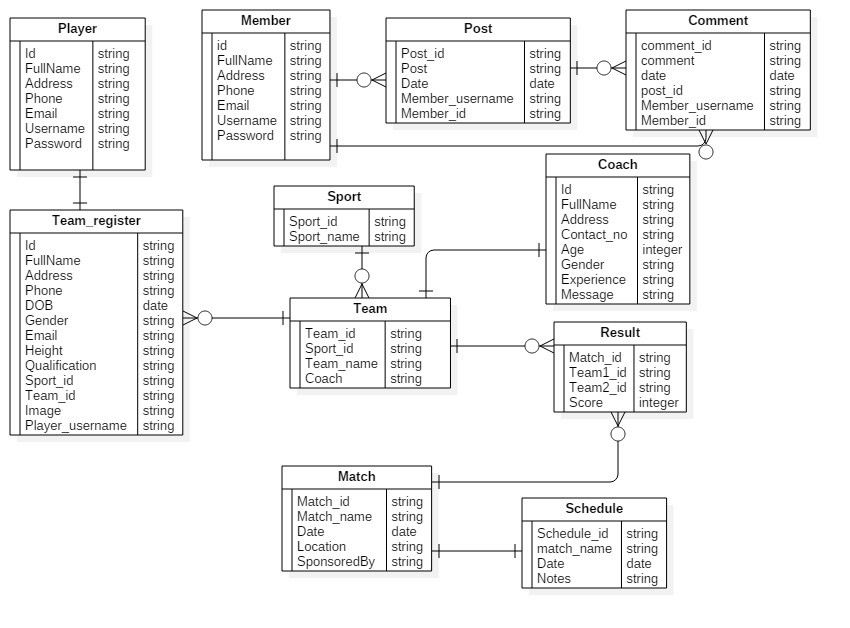


Fig12: ER diagram for sports management system

Description of diagram:

The relationship between the entities with different attributes has been shown in the above diagram. The relationship between the member, posts and comment is one to many since a member can share many posts and comments but a post and comment cannot be shared by many members. Similarly, the relationship between the player and team\_register is one to one since a player can only get registered in a single team. Likewise, a category of sport can have many teams and one coach can be associated within a team only. The relationship between the team and match is many to many since a match can have many teams and a team can play many match therefore a linking (child) table has been generated between them however the relationship between the match and schedule is one to one as a category match performs in a particular time.

1. Architectural Model

The architecture for sports management system is based on the three-tier architecture.

Three-tier architecture is the client-server architecture in which the interfaces, business logics and data stores are developed and maintained as independent modules. The three tiers are:

**1. Presentation tier:**

Presentation tier is the front end layer and consists of the user interfaces. Html, css, JavaScript etc. technologies are applied in this layer.

**2. Application tier:**

This tier consists of the functional business logic. It is often written in core PHP, c#, java etc.

**3. Data tier:**

This tier comprises of data storage or database. For example: MySQL, MongoDB, Oracle, SQL server etc.

The main reason for considering the three-tier architecture in sports management system are:

* Flexibility in managing the data. the change in the business logic does not impact the whole system.
* More secured architecture since the client is not allowed to access the database directly.
* Business logic can be used in other projects and in future since they are developed and tested.

**Database**

**Back End**

**Front End**

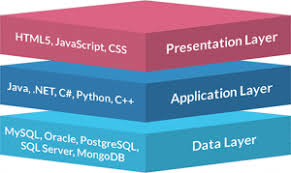


Fig13: Three tier architecture

The above diagram explains that the data stored in data layer is presented in presentation layer on which the users interacts through the medium of application layer.

1. UI modelling

UI modelling is the approach to model the complex system which does not only provide the guidelines but also the development process including the standardized visual models and notations. One of the best way for modeling the interface is prototyping.

5.1 Prototyping

In general, prototyping is an early approximation of the final project and is performed until an appropriate paradigm is achieved to develop the entire system. Simply, prototyping can be considered as the blueprint of the system however many of the details are not built in prototype.

The prototyping has been adopted due to the following benefits:

* Early detection of errors.
* Users get better understanding of the system being developed.
* Usability testing of the system.
* Improve the efficiency in system development.

The prototype that has been built for developing *sports management system* is mentioned below:

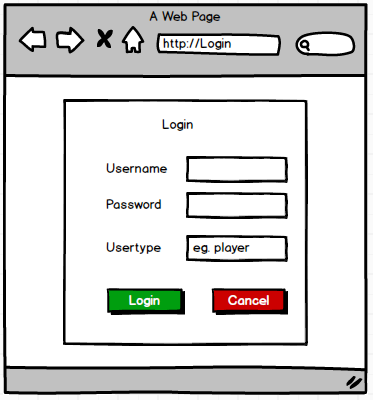
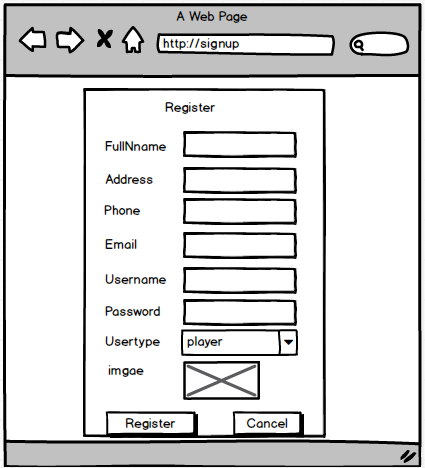


Fig14: Registration Fig15: login

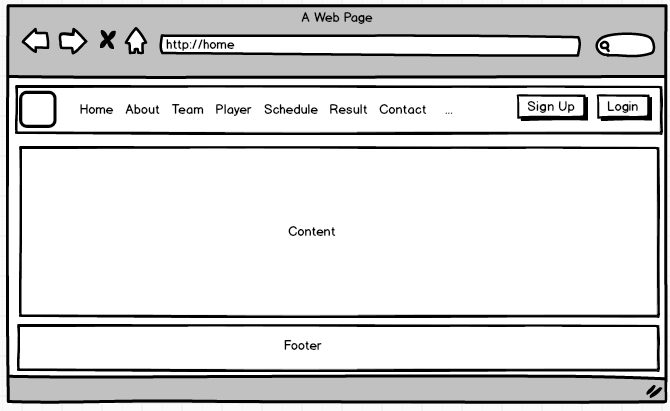


Fig16: Home page

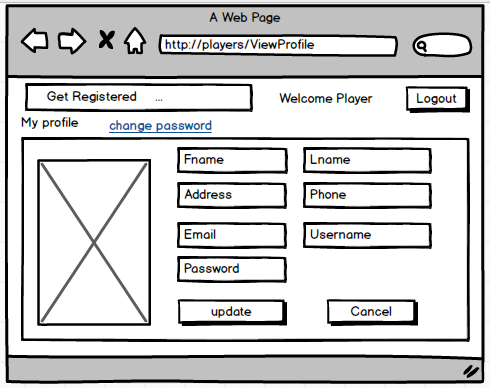


Fig17: Player dashboard

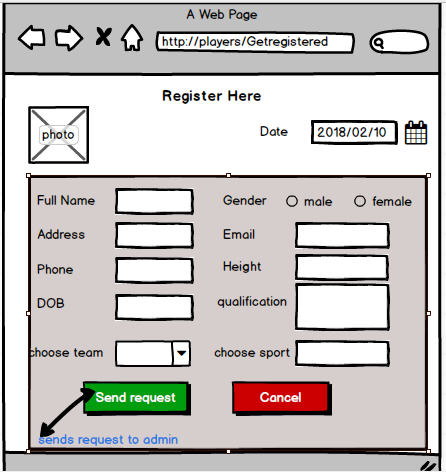


Fig18: Team registration

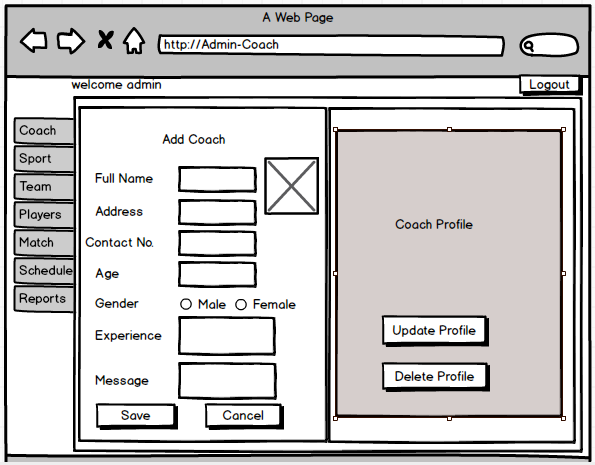


Fig19: Admin dashboard

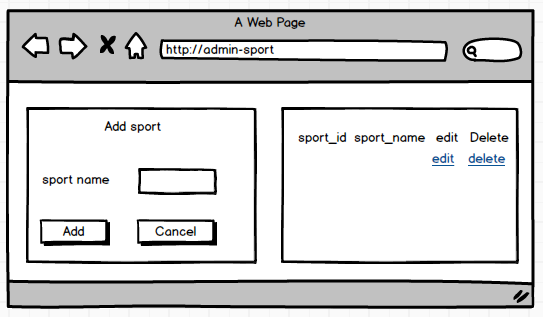


Fig20: Add sport

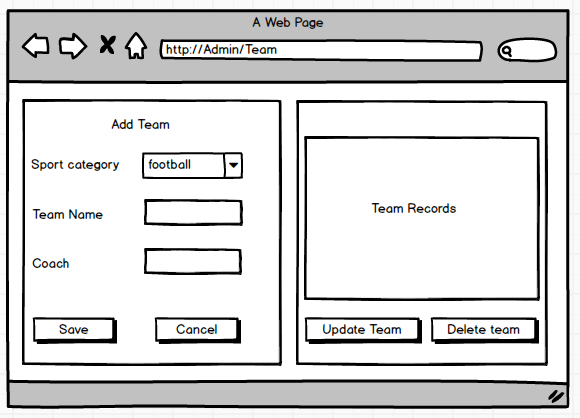


Fig21: Add team

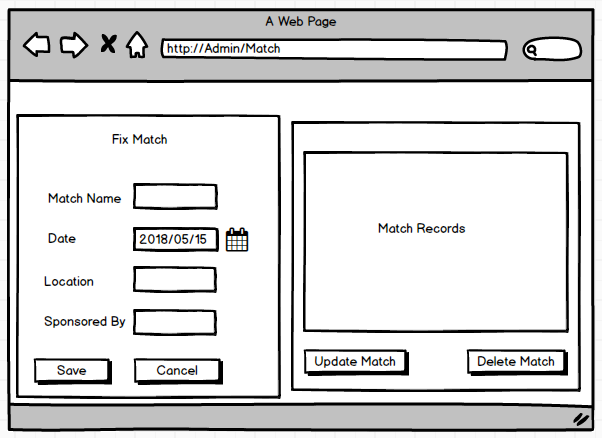


Fig22: Create match

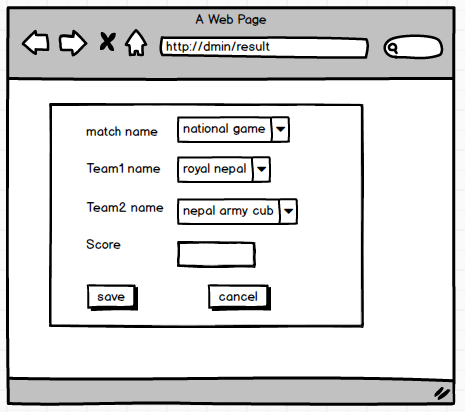


Fig23: Result of match

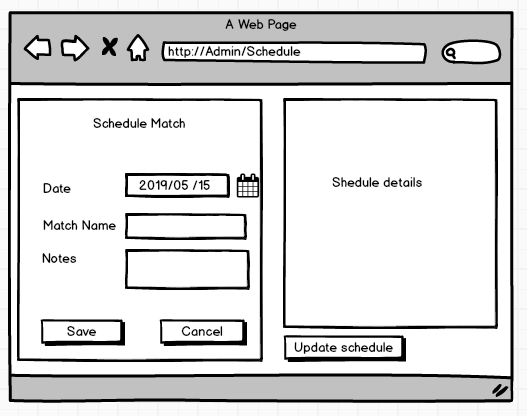


Fig24: Schedule matches

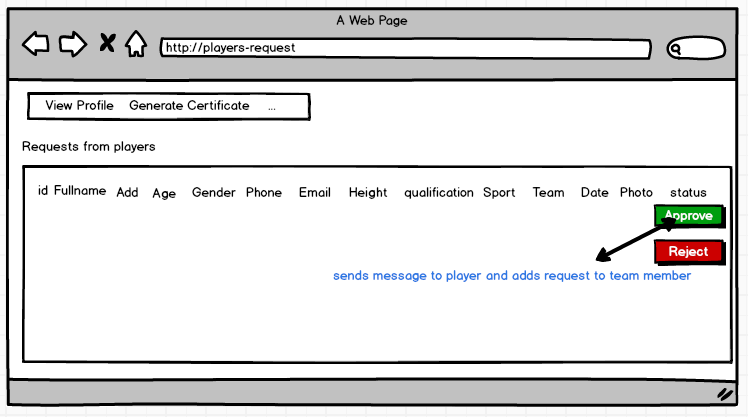


Fig25: Generate certificate

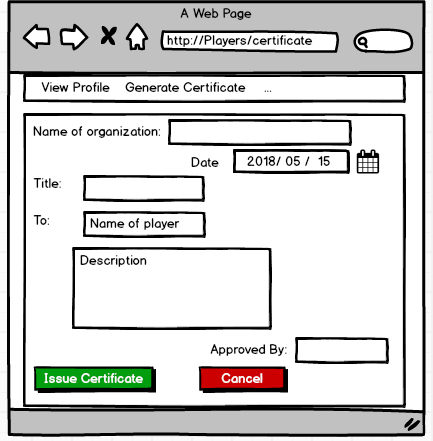


Fig26: Create certificate

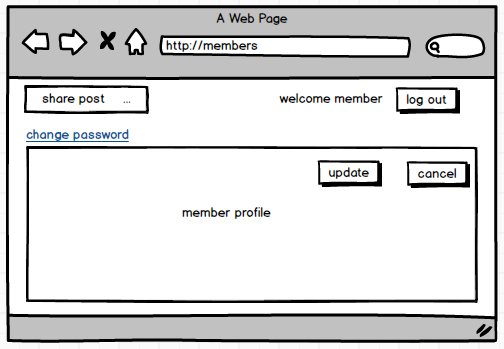


Fig27: Member dashboard

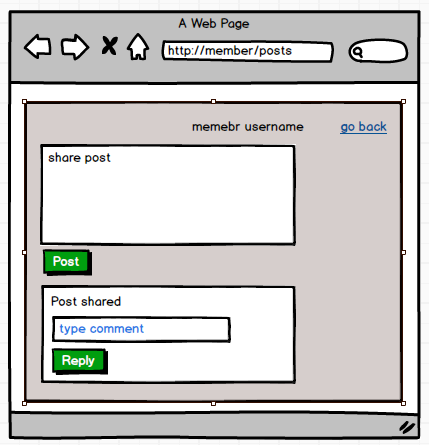


Fig28: Online forum discussion