Deportes.com

System Requirements Specifications

Group 28
11/25/2018

Software Engineering………………..Computer and System Department(CSE 2020)

Table of Contents

Introduction:

Executive Summary:

Nowadays we have up to 43 sports clubs in all Egyptian governorates, Also there are more than 20 athletics events which are held in Egypt monthly, and almost Egyptian teams or individual players in different sports participate annually in more than 50 various global competitions. On the other hand, people are very busy with their lives, work and their living conditions as they are spending a lot of time in the way to go home because of the traffic jam, So, they don't have enough time to search about sports news or sports events which can be shared and so they start to ignore about their health care or entertainment, that's why Deportes is here

As Deportes will be a large wide sports communication channel which is specific to provide users all information about the sports activities in the Egyptian sports clubs, it also offers usually notifications about any athletics events and finally it shows news about sports tournaments.

Document Overview:

This document introduces "Deports" website Requirements Specifications, it gives an overview of the functionality of the product by describing the informal requirements and is used to establish a context for the technical requirements specification.
It includes System Description, System User, System Modules, System Functions, System Models, Non-Functional Requirement, Domain Requirement and System Interface.

Abbreviations and Terminologies:

Term	Definition
User	Anyone who has an account and get always emails or notifications from the admin
Stakeholder	Any person with an interest in the project who is not a developer
Participant	Anyone visits the website to participate any event
Reviewer	A person that examines the sports news or sports clubs information and has the ability to recommend approval of the publication or to request that changes be made in the information.
Database	Collection of all the information monitored by this system.

System Requirements specifications	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Responsible Person	An individual must be identified who has authority for the administration of the names within the domain, and who seriously takes on the responsibility for the behavior of the hosts in the domain

References:

Link: https://drive.google.com/drive/folders/1BYSGqF1ft0vj2p0PnBh2Hp7hA8eiYxNJ
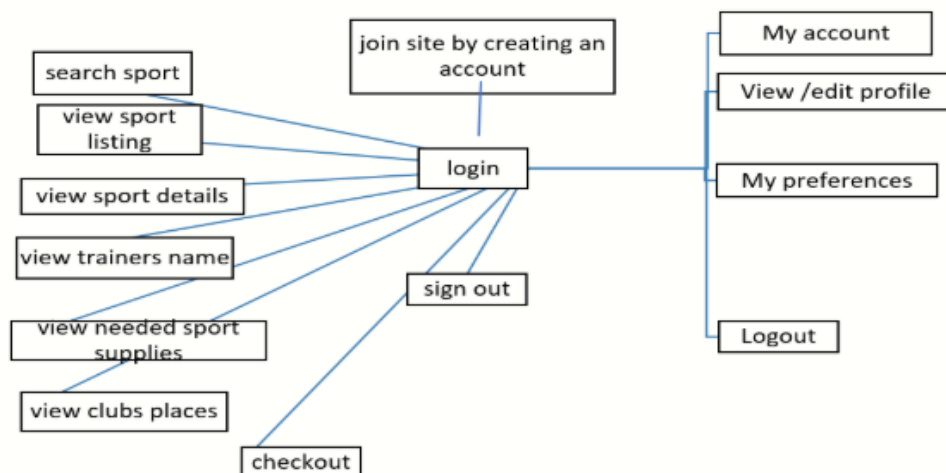
Recommended practice for software requirements, which we take it as a reference to make Deportes Website system requirements specification.

System Description:

This section gives background information about specific requirements of the Deports website and the content of each block in it.

Block diagram of website

Figure.1
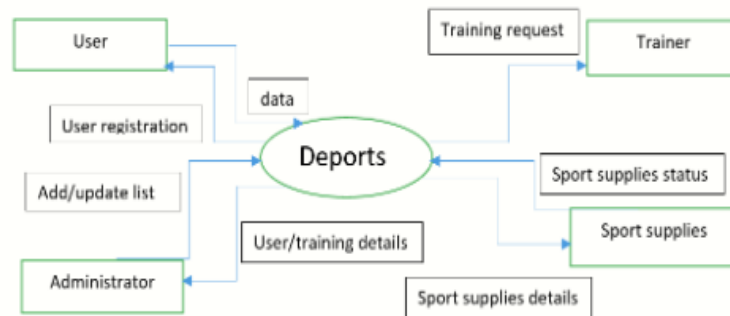
Context analysis diagram

Figure.2

Users:

User could choose his preference sport, determine its supplies and choose the best trainer in this sport.

Modules:

1. User Interface:
   Responsible for homepage, register and control be using buttons
2. Database:
   Make a request to see the database
3. Controller:
   Ask to make action on the website.
4. Submit:
   Submit any of the data on the website.

System Users:

User Characteristics:

1. Participant Description:

Participant is the person to whom the service is provided, as he/she can make search for any sport, for any location to find the best place where can practice, he also can send a feedback about any place or sports club, he/she should firstly make an account on the website to get daily notifications about all global sports news and in specific about Egyptian tournaments.

2. Reviewer Description:

A person that examines the sports news or sports clubs information and has the ability to recommend approval of the publication or to request that changes be made in the information.

User Interface:

This section will give detailed description about the Hardware, Software and communication interfaces between all website Participants and reviewers, it also provides a basic prototype of user interface.

1. Software Interface:

The Communication between the Database and the website portal action which consists of simultaneously operations act on reading or modifying the data, it happens through the sources of data such as:

1- Searching/Filtering action as participants ask for a certain specific data (Sport) and he gets it from the direct path to the database.

2- Click on a specific sport button located at the homepage, thus it will link with a new page containing all detailed information about this sport from the database.

3- Insert a Feedback text by the participant on a specific practice place, this feedback is direct sent and stored at the database as it will be shown on the webpage until the participant will click on the button to delete it, so it will remove from the database.

4- All news about athletics events or sports tournaments that are stored at the Database will be shown at the beginning of the homepage to get participants attraction to join it.

2. Hardware Interface:

If the website has any designed hardware module, there will not be hardware interface by any user or participant just the developer or reviewer who can monitor the hardware modules.

System Modules:

Modules Description:

We use activity diagrams for the description of the website system processes that describe the functionality of the system Architecture according to meet the functional requirements and Non-Functional requirements.

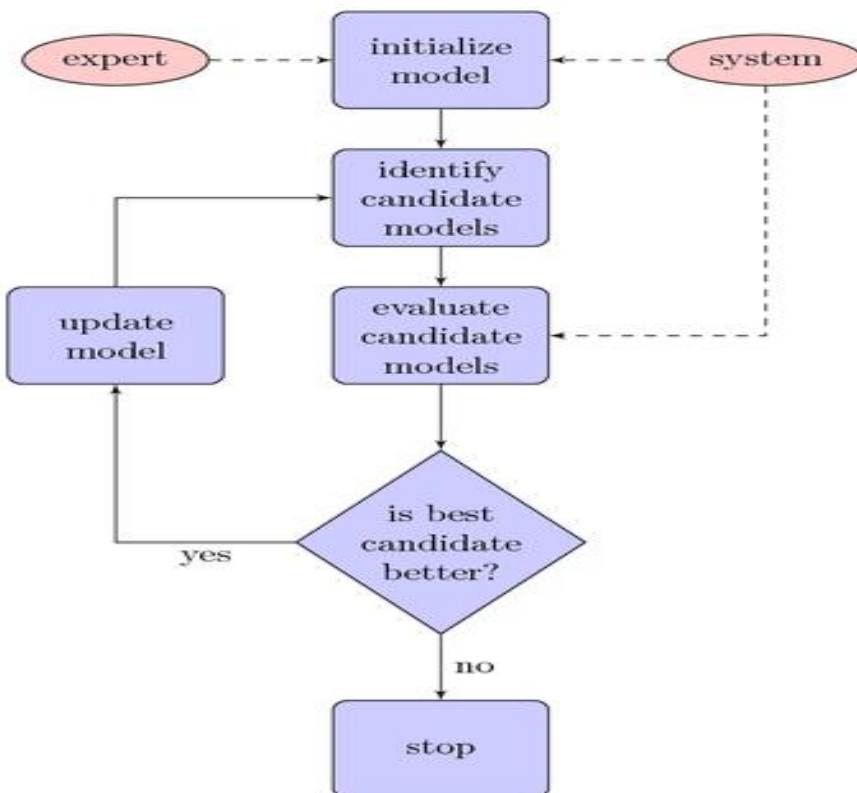All Modules take steps to start doing its function:

Figure.3

User Interface Module:

1-Register process Module Activity Diagram:

Register module is a simple module which is responsible for finishing new users account process on the website with the admin, it includes a login module which link with the admin to accept the user request.
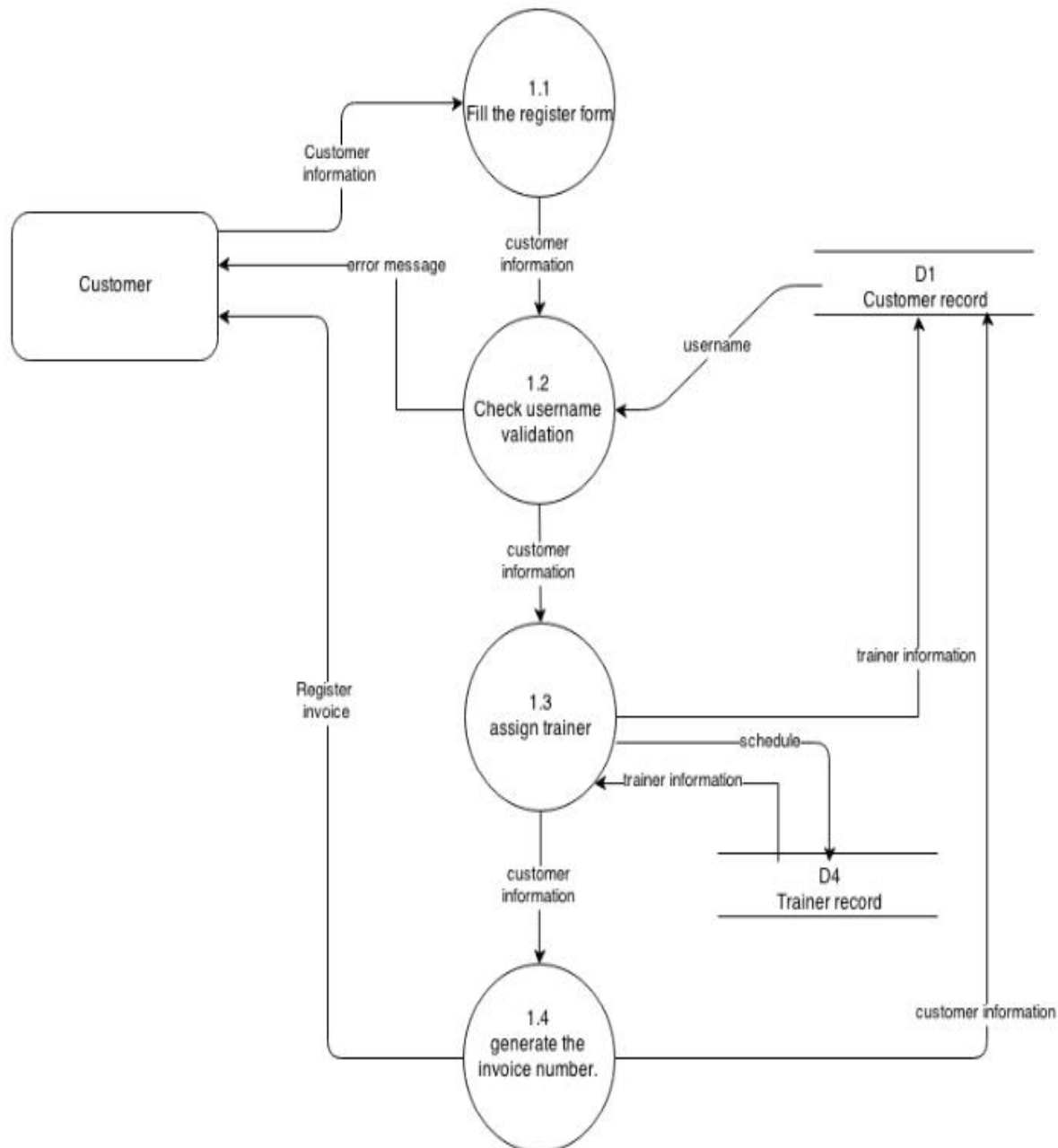
Figure.4

2-Login process Activity Diagram:
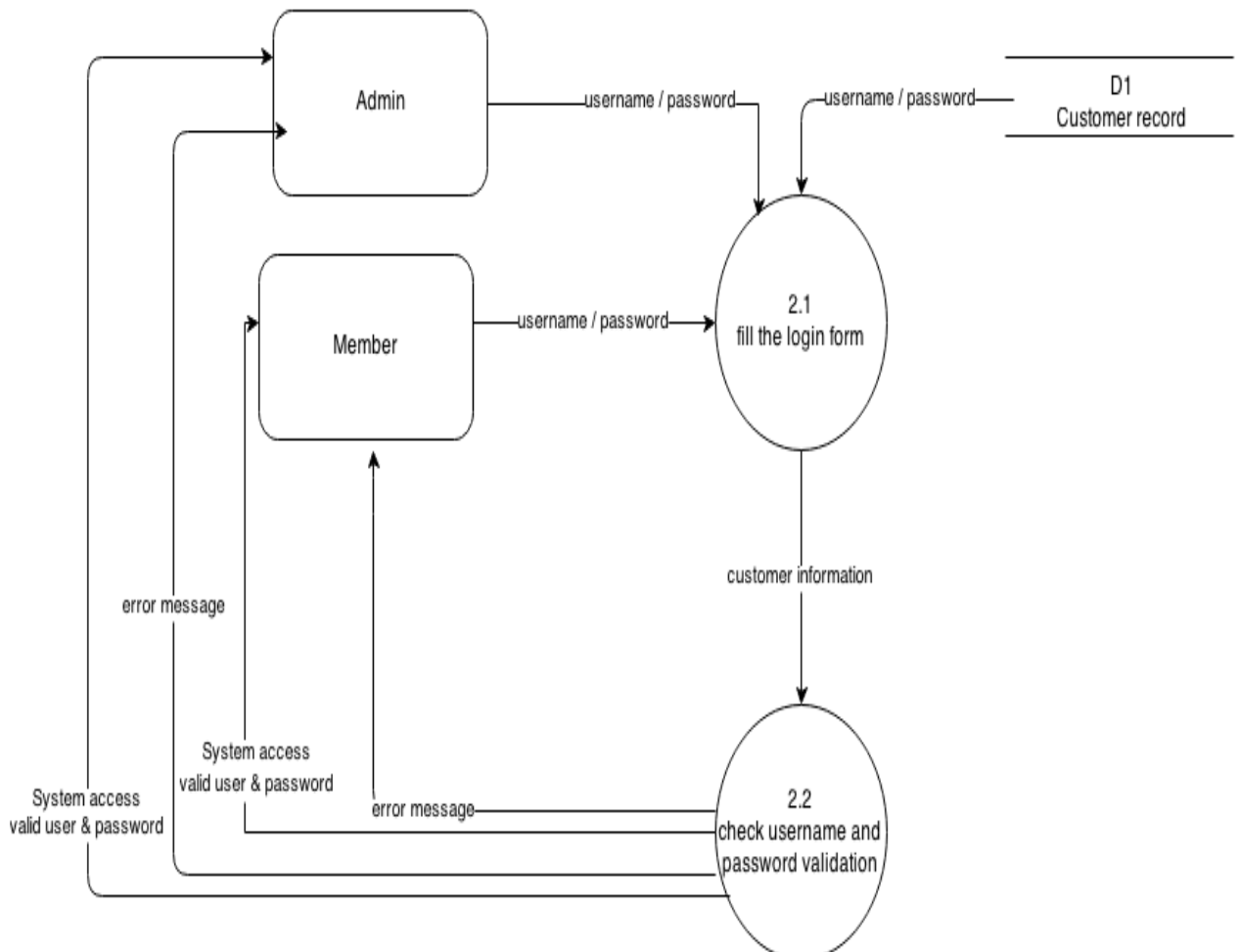
Figure.5

Database Module:

The Database module is an advanced module that allows read/write access to databases (through ODBC data sources) without the need for extensive scripting. With the database module, business or job management software can be used in conjunction with Switch workflows to further automate job decisions.

An active Database module license provides you with access to:

- The ODBC data sources option in the User preferences dialog (required to set up a database connection and to use database variables).

- The Database connect flow element (available in the Database category of the Flow elements pane). This flow element allows you to configure certain settings in databases without having to script them.
- Database variables allow you to use any values from your database in Switch.

- Database values when defining metadata for a Submit point on the Switch Web Portal.

This figure describes information Management Process and Data Flow Digram through the website system and database Module.
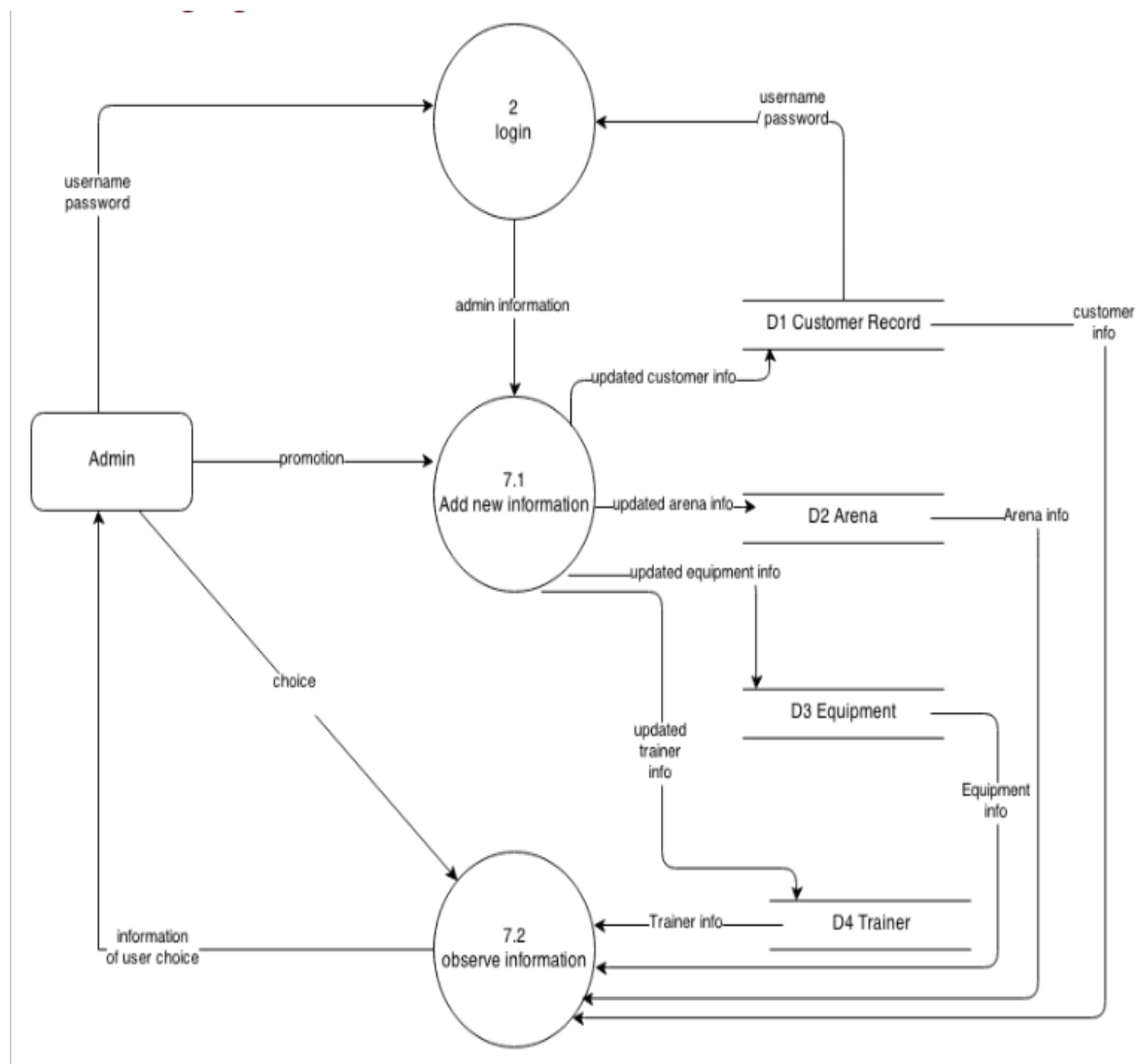
Figure.6

Controller Module:

Module which includes functions that can take decisions dependant on the user action from the user interface as it takes its inputs from the user interface Module from (Buttons Request, Search bar, direct contact, Reviews….etc) and then begins to make analysis to the input signal and finally take decisions to

1. **Database Module:**

   - select specific Data and then send to User interface to show it again to the user or
   - Delete specific Data which the User selected.
   - Hide some Information from the user interface pages.
   - Return some stored data in another page at the website to the user visit page after his searching.

2. **User Interface Module:**

   - Hide/Show specific designs to the user.
   - Control the dynamics Movement from the Websites Pages.
   - Performance depends on the time delay to the controller respond.

System Functions:

1. Sign in function (signIn_log_1) :

Description	The user first signs up for the website or sign in if he/she already has an account.
Pre-Condition	The user must have an email.
Input	The users input their information like name and email.
Output	Then they are allowed to use the website's services.
Post-Condition	The system adds their information to the data base so they can log in easily.

2. Select Sport function (Select_Sport_2) :

Description	The user then will choose the sport he/she wants from a list of sports displayed on the website.
Pre-conditions	The user must login.
Input	There will be a list of sports that the user chooses from.
Output	He/she will be redirected to a page of places that they can practice this sport in.
Post-conditions	The system redirects the user to another page.

3. Filtering(Searching) function (filter_filter_3) :

Description	The user can be able to make search and filter the options to specify the practice place of a sport by location or stars or so.
Pre-conditions	The user must sign in first and choose the sport.
Input	They will choose from a dropdown list that will be already implemented in the website.

Output	The system rearranges the places by the chosen filter.
Post-conditions	The system then redirects the user to the information page.

4. After picking function (information display_info_4) :

Description	After the user chooses the place the system redirects him/her to the information page.
Pre-conditions	The user must have chosen the sport and place.
Input	The user chooses a place from a list of places provided by the website.
Output	The system sends him/her to a page of information.
Post-Condition	

5. Send Feedback Function (Send_Feedback_5) :

Description	The user can be able to send a feedback about a specific practice place of one of a sport to make a recommendation for it to other visitors of the website.
Pre-conditions	He/she had joined this sport at this place from the website.
Input	Click on submit button to enter the text.
Output	The text will be shown on the sport webpage at feedback bar
Post-Condition	Feedback text will be stored at the database

System Models:

Sequence Diagram:

- Sequence diagram shows to us a sequence between some behaviors that happen between different actors (users, website and developer).
- When user request to find clubs, website respond and shows them.
- When user request to know about sports, website show all details.
- When user wants to leave feedback at the same time server send a notification to developer to review it.
- Developer reviews it and gives approval then display the feedback of user.
- Developer pushes notification to users.
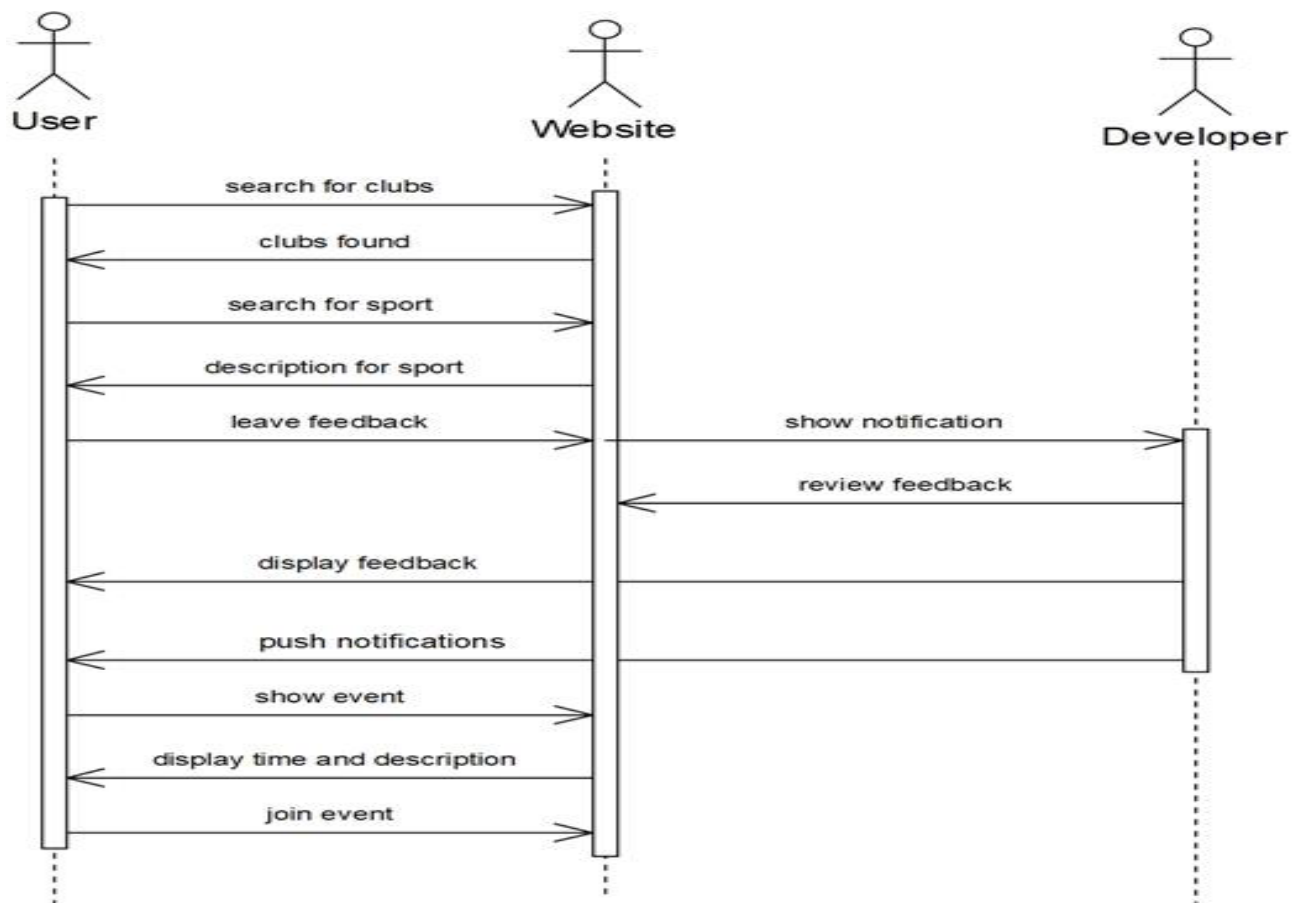- User requests to show events, then website show all details
- User joins the event.

Figure.7

Use Case Diagram:

The diagram shows the relations between actors (Users and developer) and different use cases.

- As we see that developer push notifications to users.
- If User wants to attend an event -> have a look on time and description.
- If User wants to participate in tournaments -> have a look on time and description.
- If User wants search on website about clubs, then search about clubs in specific location, then choose sport want to know more about so have a look on time and description.
- As we can see (time and description use case) is included from many basic use cases.
- When user wants to leave feedback, it must be reviewed by developer and then give the approval and display it on website.
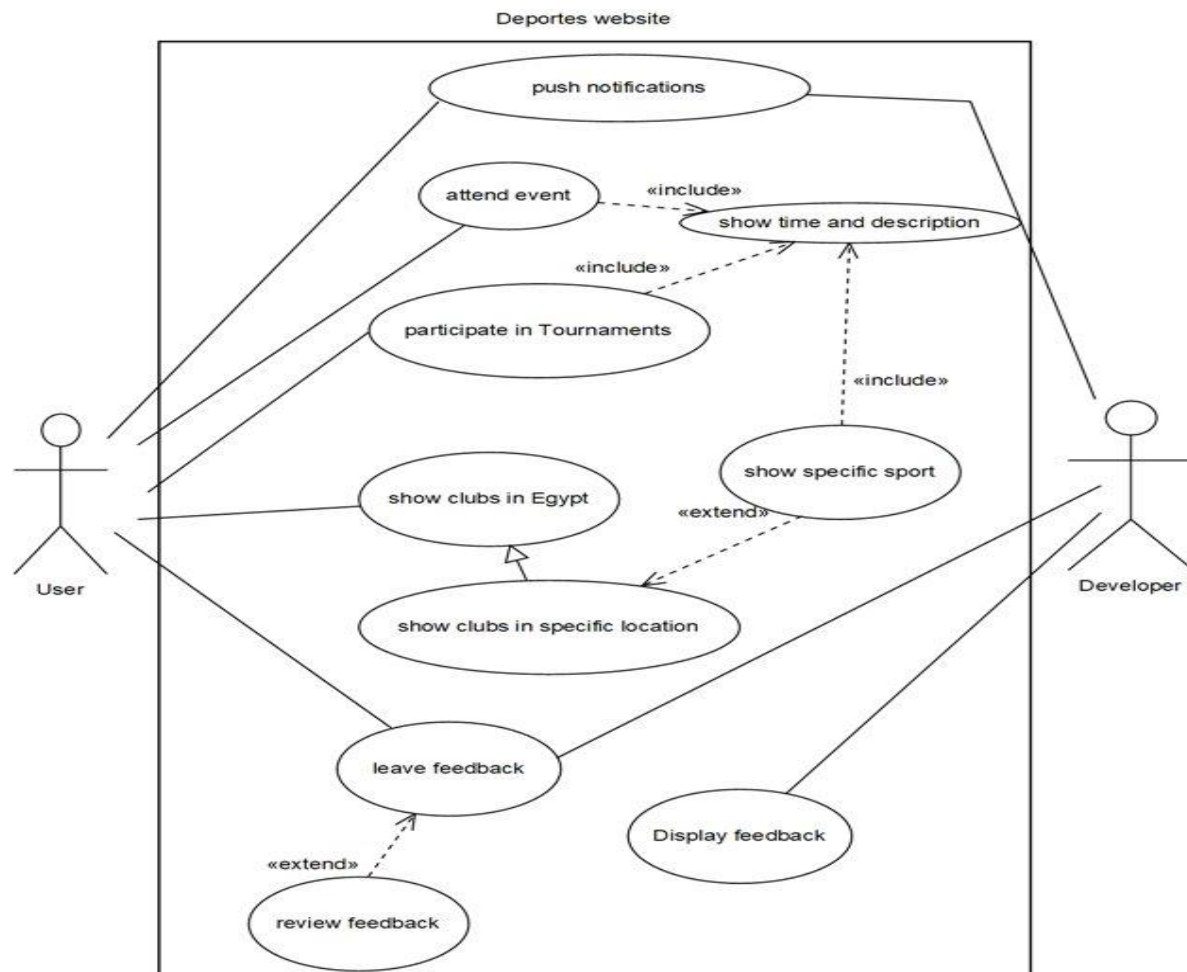

Figure.8

Non-Functional Requirements:

Performance Requirements

Performance requirement by the user side is, web application should be developed as a lightweight web app so that it can work on almost any platform even with slower internet connections. Expected number of simultaneous user should be at least 100. System should be able to deal with 100 users at the same time. Also database of the system should handle at least a thousand of users at any periods.

Requirement Security:

Product should be able to protect privacy of user data. Workspace of the user should only be accessed through user own credentials and any other user should not be able to access to the user private data. Since execution will also be done user should be restricted in terms of user rights. User should only access to his own workspace and should not access to any other workspace with the programs they run. Also rights of the user should be restricted so that user can not harm to system by the programs they run or by the commands they run on terminal.

Portability Requirements:

As explained in the performance requirements section, software should be lightweight, So that it can run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase. Such as using JavaScript and HTML5 instead of Apache Flex Portability also means running on most number of different platform without an additional effort. To achieve this, web application should be developed by using the common technologies and tools which are provided by all common web browsers and operating system such as HTML5, JQuery.

Domain Requirements:

Definition:

A **domain** is a field of study that defines a set of common requirements, terminology, and functionality for any software program constructed to solve a problem in the area of computer programming, known as domain engineering. The word domain is also taken as a synonym of *application domain*.

Purpose of the Domain field:

Domain Understanding is the Key to Successful System Development as all Developers are often in a hurry to start software design and construction and, as a result, have an unfortunate tendency to overlook, to their cost, the very significant advantages offered by developing a prior and adequate understanding of the surrounding application domain, So understanding of a development project's specific application domain is an essential precursor to successful systems development.

Goal:

The global objective of Domain Requirements Description activity is to provide an overview of the system's context and capabilities. This activity aims thus at gathering needs and expectations of the application's stakeholders and provide a complete description of the behavior of the application to be developed. These requirements should be described using the specific language of the application domain and perspectives of users. This activity specifies the application's system-level functional and non-functional requirements. It has to establish a first estimation of the scope, the size, the complexity of the application and the amount of associated costs.

Requirements:

There are several requirements that must be met to establish a domain. In general, it must be responsibly managed such as:
- There must be a responsible person to serve as an authoritative coordinator for domain related questions.
- There must be a robust domain name lookup service, it must be of at least a minimum size.
- The domain must be registered with the central domain administrator (the Network Information Center (NIC) Domain Registrar).

Let's discuss the requirements in details:

- **Responsible Person:**
An individual must be identified who has authority for the administration of the names within the domain, and who seriously takes on the responsibility for the behavior of the hosts in the domain, plus their interactions with hosts outside the domain. This person must have some technical expertise and the authority within the domain to see that problems are fixed. The responsible person must be competent and available to receive reports of problems, take action on the reported problems, and follow through to eliminate the problems if a host in a given domain somehow misbehaves in its interactions with hosts outside the domain.

- **Domain Servers:**
A robust and reliable domain server must be provided. One way of meeting this requirement is to provide at least two independent domain servers for the domain. The database can, of course, be the same. The database can be prepared and copied to each domain server. But, the servers should be in separate machines on independent power supplies, et cetera; basically as physically independent as can be. They should have no common point of failure.

One of the difficult problems in operating a domain server is the acquisition and maintenance of the data. In this case, the data are the host names and addresses. In some environments this information changes fairly rapidly and keeping up-to-date data may be difficult. This is one motivation for sub-domains. One may wish to create sub-domains until the rate of change of the data in a sub-domain domain server database is easily managed.

In the technical language of the domain server implementation the data is divided into zones. Domains and zones are not necessarily one-to-one. It may be reasonable for two or more domains to combine their data in a single zone.

The responsible person or an identified technical assistant must understand in detail the procedures for operating a domain server, including the management of master files and zones.

The operation of a domain server should not be taken on lightly.
There are some difficult problems in providing an adequate service, primarily the problems in keeping the database up to date, and keeping the service operating.

- **Domain Registration:**

  Top level domains must be specially authorized and registered with the NIC domain registrar. The administrator of a level N domain must register with the registrar (or responsible person) of the level N-1 domain. This upper level authority must be satisfied that the requirements are met before authorization for the domain is granted. The registration procedure involves answering specific questions about the prospective domain. A prototype of what the NIC Domain Registrar may ask for the registration of a second level domain is shown below. These questions may change from time to time. It is the responsibility of domain administrators to keep this information current. The administrator of a domain is required to make sure that host and sub-domain names within that jurisdiction conform to the standard name conventions and are unique within that domain.

Finally An illustration of the three questions that arise in an effective engineering System. Are the requirements (abstraction) appropriate for properly reflecting end-user tasks; is the design (reification) adequate for the purpose(s) in hand; and have we managed all of the risks inherent in the introduction of the new system into the application domain?
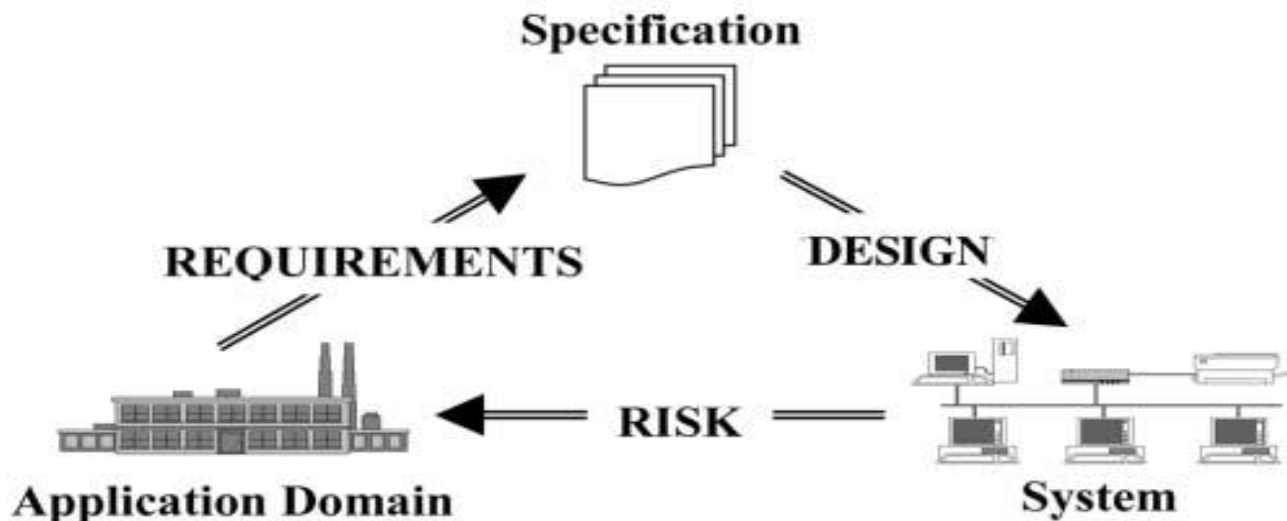

Figure.9

This diagram answers and emphasis the central position occupied by the specification, as a domain descriptive theory of both the application system and its associated domain.
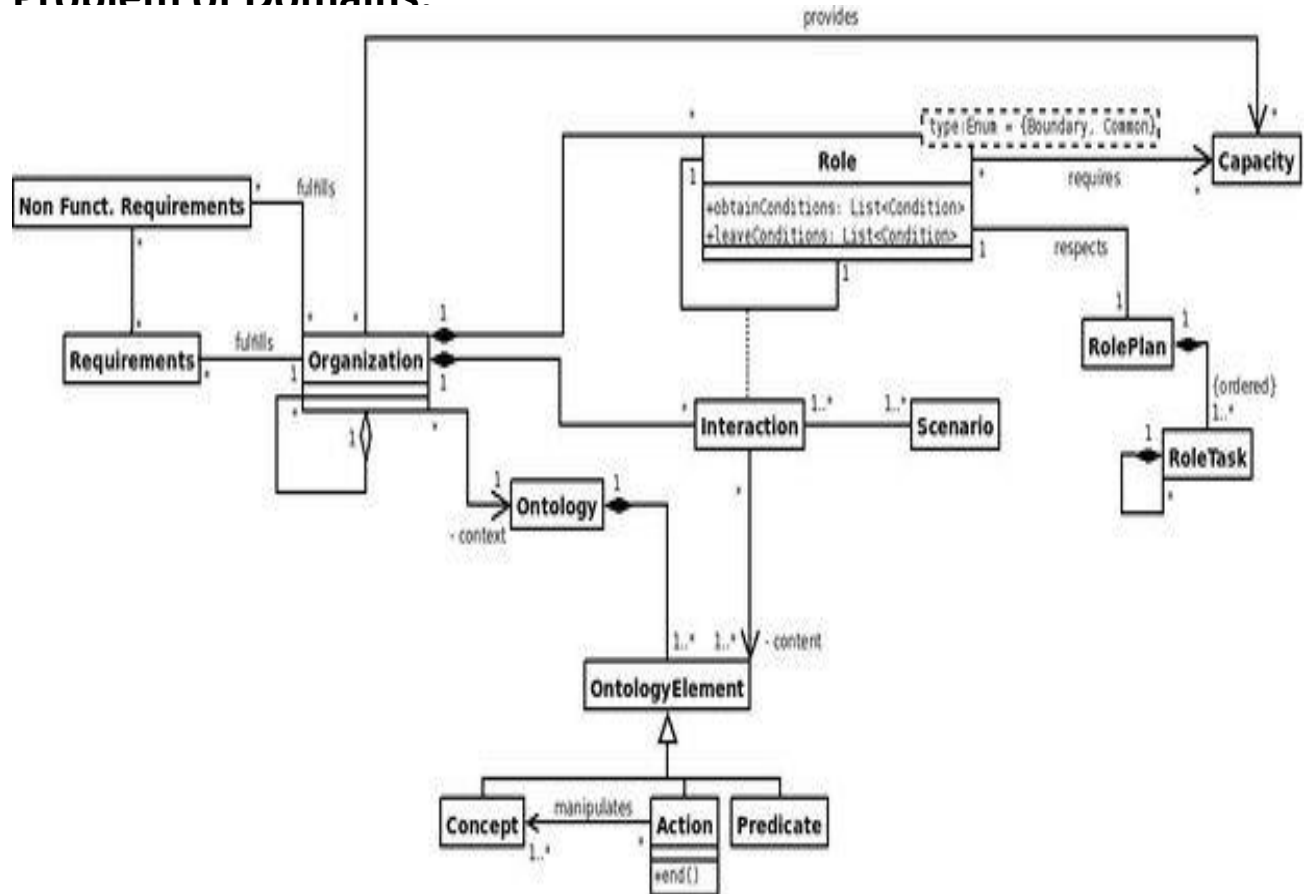
Problem of Domains:

Figure.10 "The UML diagram of the Problem Domain model"

The Problem Domain model includes elements (Figure 1) that are used to catch the problem requirements and perform their initial analysis: Requirements (both functional and non-functional) are related to the organization that fulfills them. An organization is composed of Roles interacting within scenarios while executing their Role plans. An organization has a context that is described in terms of an ontology. Roles participate to the achievement of organization's goals by means of their Capacities. In this subsection we will discuss the three most important elements of this domain: organization, role, capacity.

The **Main concepts of domain problems** are shown at this table below:

Functional Requirements	A function that the software has to exhibit or the behaviour of the system in terms of interactions perceived by the user
Non-Functional Requirements	A constraint on the solution. Non-functional requirements are sometimes known as constraints or quality requirements.
Concept	A category, an abstraction that shortens and summarizes a variety/multiplicity of objects by generalizing common identifiable properties.
Predicate	Assertions on concepts properties.
Action	A change realized by an entity that modifies one or more properties of one or more concepts.
Role	An expected behavior (a set of role tasks ordered by a plan) and a set of rights and obligations in the organization context. The goal of each Role is to contribute to the fulfillment of (a part of) the requirements of the organization within which it is defined. A role can be instantiated either as a Common Role or Boundary Role. A Common Role is a role located inside the designed system and interacting with either Common or Boundary Roles. A Boundary Role is a role located at the boundary between the system and its outside and it is responsible for interactions happening at this border (i.e. GUI, Database, etc).
Interaction	A dynamic, not a priori known sequence of events (a specification of some occurrence that may potentially trigger effects on the system) exchanged among roles, or between roles and entities outside the agent system to be designed. Roles may react to the events according to theirs behaviors.
Role Task	An activity that defines a part of a role behavior. A Role Task may be atomic or composed by a coordinated sequence of subordinate Role Tasks. The definition of these Role Tasks can be based on capacities.
Scenario	Describes a sequence of role interactions which a (part of) requirement.

System Interface:

Interface:

Specify:

1. The logical characteristics of each interface between the software design and its users.
2. All the aspects of optimizing the interface with the person who must use the system.

This is a description of how the system will interact with its users. Is there a GUI, a command line or some other type of interface?

Are there special interface requirements? If you are designing for the general players population for instance, what is the impact of ADA (American with Disabilities Act) on your interface?

Hardware Interface:

Specify the logical characteristics of each interface between the software design and the hardware components of the system. This includes configuration characteristics. It also covers such matters as what devices are to be supported, how they are to be supported and protocols. Designers should be able to look at this and know what hardware they need to worry about in the design. Hardware interfaces exist in many of the components, such as the various buses, storage devices, other I/O devices, etc. A hardware interface is described by the mechanical, electrical and logical signals at the interface and the protocol for sequencing them (sometimes called signaling).

Software interfaces

Specify the use of other required software design and interfaces with other application systems. For each required software design.

For each interface, provide:
1. Discussion of the purpose of the interfacing software as related to this software design.
2. Definition of the interface in terms of message content and format.

 The system must use SQL server as the website's database component. Communication with the DB is through ODBC connections. The system must provide SQL data table definitions to be provided to the website DBA for setup.

Communication interfaces

Specify the various interfaces to communications such as local network protocols, etc. These are protocols you will need to directly interact with. If you happen to use web services transparently to your application then do not list it here. If you are using a custom protocol to communicate between systems, then document that protocol here so designers know what to design. If it is a standard protocol, you can reference an existing document or RFC.