**4.2 Proposed system**

**4.2.1 Overview**

The proposed system mainly deals system architecture and subsystem decomposition with hardware and software design. System architecture is the conceptual model that defines the structure, behavior, and more view of the system. It consists of system component and sub system developed that will work together to implement the overall system. It is also a set of representation of existing system. These representations initially describe general, high level functional organization and are progressively refined to more detail and concrete description it primarily concentrates on the internal interfaces among the system's components or subsystem and on the interface between the system and its external environment. The other thing is component diagram. It is unified modeling language (UML) and it shows how components are wired together to form larger components of software system.

Also the proposed system deals hardware software design; it refers to the identification of a system's physical components and their interrelationships. This description allows hardware designers to understand how their components fit into system architecture and provides to software component designers important information needed for software development and integration.

We used four basic layer in this architecture the first one is client side layer which is represented by computer used by user to enter query and receive respond from server. The second layer is server side which accept request from client and give respond only when there is third part which is internet connection between them. Then internet service portal customize website and aggregate information from wide array in a consistent manner and provide necessary information to the server in order to give respond for the request. The fourth layer is database layer to access data from the database to perform operations.



***Figure 4.2.1 True system architecture***



***Figure 4.2.2 general system architecture***

**4.2.2 Subsystem decomposition**

• Collection of classes, associations, operations, events and constraints that are closely interrelated with each other. In UML subsystems are modeled as packages





**Subsystem description**

|  |  |  |  |
| --- | --- | --- | --- |
| subsystem | | purpose | classes |
| System administrator | | Responsible to update database files, create user, to manage signup, put standard for control mechanism | Administration, user, registration ,player, coach, referee |
| Inter-departmental cup | | Responsible to provide all systems function, register funs, release notification, record and release score etc. | User, administration, player, coach, referee, team, registration |
| Half-life cup | | Responsible to provide all systems function, register funs, release notification, record and release score etc. | User, admin, player, coach, referee, team, registration |
| GC cup | | Responsible to provide all systems function, register funs, release notification, record and release score etc. | User, admin, player, coach, referee, team, registration |
| Notification | Responsible for providing new information to the user, release schedule, notify new score and any other announcement, and respond regard for the users comments | | Fixture, schedule, announcement |

**4.3.3 Hardware software mapping**

The Hardware Software mapping is described to indicate the various hardware devices and equipment’s used in the system and its interaction with the software components. The system will have two processes, deployed in single or separate machine, that run in parallel, namely, web server process and the database process. The database process, which runs on MYSQL database engine, is responsible for maintaining data manipulation operations.

Where us the web server process is responsible to host the web pages of the system and process clients’ request. In case of the client side, only a browser is required to access the objects.



Fig hardware software mapping of the system

**Component diagram**

The purpose of a component diagram is to show the relationship between different components in a system. The term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system. There exists a whole development approach that revolves around components: component-based development (CBD). In this approach, component diagrams allow the planner to identify the different components so the whole system does what it's supposed to do.



Fig component diagram

**Component Description**

**Authentication controller**

Description: This is a class responsible to authenticate a user

Role: it is responsible to check if the validity of a user trying to login to the system

**Page controller**

Description: This is a route class which will return to each page as needed

Role: it is responsible to route each page requests in the system.

**PID generator**

Description: the base class which is used during player registration

Role: this is a class which will generate a PID for a player during registration.

**Data organizer**

Description: the base component which is directly linked to the users of the system and sends it back to the view called view profile.

Role: is responsible for viewing the personal information of the player, editing and updating some status.

**Component controller**

Description: this is a class which extends the base controller class responsible to look for components.

Role: this class is responsible to control all the components in the system.

**Form controller**

Description: the base form controller class in which governs the form submission in the system and sends back to the view called view form.

Role: this class is responsible for controlling of form submission before further process.

**Profile model**

Description: this is a class which is linked with the profile table in the database.

Role: responsible for the extraction of record from and to the table.

**Component model**

Description: this is a class which is linked with the database table in which used to determine the component to display to the user.

Role: responsible for the getting record from the database in order to display the appropriate component to the user.

**Score model**

Description: this is a class which will be connected to the score table in the database and responsible to send information to the specified data organizer controller.

Role: will fetch data from the database table and will pass it to the specified controller.

**Message model**

Description: this is a class which will help to connect to the table in the database, and will help in the data organizer class.

Role: this will help to connect to the message table in the database and will retrieve record from the table for further processing.

**Deployment diagram**

We use deployment diagram to show software and hardware work together. It describes system hardware, software, and network connections for distributed computing. It covers server configuration and network connections between server nodes in real-world setting. The Deployment model describes how and where the system will be deployed. Physical machines, devices and processors are reflected as nodes, and the internal construction can be depicted by embedding additional nodes or artifacts. Artifacts, such as executable, are allocated to nodes to model the system's run-time configuration. The allocation is guided by the use of deployment specifications. The physical locations, deployment of artifacts and connectivity between nodes of the final deployed system are depicted in the Topology package.



Fig deployment diagram

4.3.6. Access control

Access control is formal security model of computer systems which assigns actors to different level of accessing classes. Access control can be represented by access matrix where its rows show the actors and its columns show the classes that are to be protected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Actors Objects | Fixture | Score Board | Player | Referee | Coach | Registration Form |
| Admin | View()  Create() | View()  Create() | View()  Update() | View()  Update() | View()  Update() | View()  Send() |
| Coach | View() | View() |  |  | View()  Update() | Approve() |
| Referee | View() | View() |  | View() |  |  |
| Player | View() | View() | View()  Update() |  |  |  |
| Visitor | View() | View() |  |  |  | Fill()  Submit() |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actors Objects | Card | Team | Player List | Administration | Account |
| Admin | View() | Assign Coach()  Assign Referee() | View()  Update()  Manage Account() | Manage Account() | Create()  View()  Delete()  Update() |
| Coach | View() | Create Team()  Set Position() Notify() | View()  Select() |  | Login()  Forget Password()  Logout() |
| Referee | Manage() |  |  |  | Login()  Forget Password()  Logout() |
| Player | View() |  |  |  | Login()  Forget Password()  Logout() |
| Visitor |  |  |  |  |  |