Cricket Club Hub

By

Abdul Rehman

Supervised By

Dr. Muhammad Usman

Department of Computer Science

Quaid-i-Azam University

Islamabad

2013-2017


بسم الله الرحمن الرحيم

QUAID-I-AZAM UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

Dated: _____

FINAL APPROVAL

This is to certify that we have read the final report submitted by Abdul Rehman and it is our judgment that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islamabad for the degree of the Bachelor in Computer Science.

Committee

1. ***EXTERNAL EXAMINER*** **_____**

2. ***SUPERVISOR*** **_____**

Dr. Muhammad Usman
Department of Computer Science

Quaid-i-Azam University, Islamabad.

3. ***INCHARGE*** **_____**

Dr. Onaiza Maqbool

Department of Computer Science

Quaid-i-Azam University, Islamabad.

ACKNOWLEDGMNETS

In the name of Allah, the Most Beneficent, the Most Merciful.

I would like to thank **Ahmar Najeed Satti** for providing me information and background history of Diamond Cricket Club and Ground located at Islamabad. I would like to thanks all our teachers for guiding me thoroughly in solving our problems related to my project and especially to my supervisor his cooperation, kind support and intellectual abilities to bring this project to completion.

I would like to thank my supervisor **Dr. Muhammad Usman** for guiding me in understanding the concepts related to my project. I would also like to thank our families and friends, especially **Muhammad Taqi** and **Muhammad Yaqoub Khan** for their continuous encouragement and moral support.

Abdul Rehman

Abstract

Diamond Cricket Club located at Islamabad offers several activities between team, members for promoting the importance of sports in daily life. Diamond Cricket Club has number of staff who manage the members data, team data, financial records, diamond ground booking data and match scheduling details. This staff includes members, who also play a role of players in team. The current system is totally based on paper work, which causes many problems for storing and retrieving data. The administrator staff faces problems to save paper-based record. It is not adequate approach to store data of members, team and their financial records, winning awards data in form of papers. Therefore, we need to establish a system which performs the club operations in reliable way and maintain records of team, club members and all other relevant details such as financial records, ground booking, member monthly fees, membership dues and payment details of match tickets.

A web-based system is introduced to address above-mentioned problems in a structured way and we can also store data in a digital form. The system manages player portfolio, team data, financial records, match scheduling, employee records with their payment details and ground booking records. Administrator, members of club and any free end user can use Duck-Worth Lewis calculator. Administrator can perform queries for making strategic and tactical decisions like selection of members in a team according to their previous history or status. Beside it, this system can be used by administrator as a server side and client can access it through the internet. The clients can get information about club activities such as upcoming matches from the system and he can book match tickets online by entering his details such as email or name etc. The clients can request for membership in club by entering their details later on server side administrator can reject or accept his request based on rules of club. Members and users can send feed-back to club. Only residence of Islamabad and Rawalpindi can send requests for membership and ground booking because initially, we are just taking two cities. The role of coach and cricket tournaments with score sheets are not included.

The tools used for developing this system are Dream Viewer, WAMP Server. Three or four languages have been used for constructing web-based system that are PHP (Personal Home Page) HTML5, AJAX, JQuery, Bootstrap and JavaScript. We need internet to access the system. Without the internet users will not access the system.

For future enhancement, we will include other operations like diamond cricket academy records such as trials details and each learner details. It will allow sponsors to elect team from club for advertisement. Cricket Academy System will also be included in the system. This system will support the live matches for clients. The graphs based records such as performance of members will be shown through multiple graphs. We will manages the user requests for membership in club and ground booking for more than two city. Also notification module will enhanced later by which, administrator will send important messages through Short Message Service (SMS).

Table of Contents

Table of Figures

List of Tables

"Education is the most powerful weapon which you can use to change the world."
By: Nelson Mandela (1918-2013)

Chapter 1

Introduction

The Diamond Cricket Club located at G-8/2 near Pakistan Institute of Medical Sciences (PIMS), Islamabad which provides facility to organize cricket matches with different teams at Diamond Cricket Ground. Also, Islamabad Cricket Association (ICA) office is also located at this club. Diamond ground is most expensive ground in Islamabad (ISB) and Rawalpindi (RWP). The ground is also booked for local and private teams. In week days the booking fee is approximately 15000-20000. In weekend days booking rate is approximately 25000-30000. Sponsorship can be done by an individual or by a company. All the expenses are managed by above resources like ground man, security guard, coach, and other technical staff salaries. Also a cricket academy is also located at this ground whose train boys. The management staff includes president, vice president, and financial manager. Schedule of match or tournament comes from ICA to president of club. The management staff combines to make a committee for the selecting players for upcoming matches or tournament with local teams. The performance of each player is noticed manually by organizing test matches with other team. Each player is registered to club then he can participate in teams according to rules and regulations defined by club vice-president. The financial aspect of teams and ground is handled by financial manager. New member can apply for membership in club also by giving his details along with two photos, bay form or Computerized National Identity Card (CNIC) copy and other relevant data manually. All the records are saved in form of papers or in written documents. At the end of each month, the registered member pay fee to club handled by financial manager.

1.1) Problem Definition

First of all, members and clients has no information about up to date match schedule which is comes from ICA to president of club and then president of club has to inform all the registered club and other management staff. Second, registered members has no portfolio management platform from

which any stakeholder or sponsor may be client can view the player history and other relevant information. Third, registered members face problems to taking information about the upcoming tournaments from club. Fourth, if a person wants to take membership of club then he need to fill a paper form which is time consuming process and there is no proper way of informing that member about membership process. Club management face problem to store the records of each member their scores sheets and other relevant details. Financial management also face problems to store the expenses of team, member's payment and other monthly expenditure regularly. Interested users who want to watch the scheduled matches at ground faces problem to book match tickets. President has no information of the previous records of teams like performance records or chart like graphs through which he made decisions for future. Problem in the selection of players in a team because all main staff persons arrange meeting for selecting the members which is time consuming and need time engagement. Every president tried to make efforts for the welfare of club. For this purpose, he needs feedback from members, management staff, sponsors and other interested people.

1.2) Proposed Solution

Hence a system has been proposed to be deployed in Diamond Cricket Club, Islamabad named as "Diamond Club Management System" which consists of Web-based application. Which allows management staff to manage player portfolio data, team selection, Duck-worth Lewis computations, and details of match schedules with tickets. The financial manager manages financial records of teams, players, match tickets and ground. The president acts as an administrator which has control on management system and can access the system by his email and password for security. The stakeholders or clients can only view match schedule of club. Registered members can access their account and receives notification from president. Members can change their account passwords for security purpose. For membership user will have to fill the application form and submitted. New member receives notification with password of his account from club if his request approved by admin. Any clients can view the members profile and read relevant data about that member who is registered from club. The member profile contains the D.O.B (Date of Birth), name, major teams and his playing role or batting style or bowling style. A client can easily access any member information. Any sponsor, member or client can send feedback to club administration for the welfare of club by entering user first name, email and message.

Administration staff have to login for approving the membership forms after reading data of new member. Coach and other committee have login for selecting members for matches. Admin have rights to change, modify, details of matches and payment records. In case of weather or other circumstances admin can delay the match and to cancel the match according to rules or agreement of other team sponsor or captain of team. Admin will keep track of winning team with specific team name, match date and its status. So that, he can make decision after viewing the previous information

of teams. Financial records managed by financial manager. Duck-Worth Lewis calculator used by anyone in case of weather or other circumstances.

1.3) Scope

The scope is like a boundary of system which defined three things.

1. Stakeholders
2. Data that are the input and output
3. Major functions and features that are to be delivered to users

There are three types of stakeholders who will interact with system, members who are registered from club, end user or any client and third one is administrator of club. This system is only build for "Diamond Cricket Club". This system allow administrator to registers new members and to manage records of teams and ground rent details. The system allow administrator to keep track of financial records related to members, employees and other monthly expenditures of club. It also allow administrator to select team, add, remove, delete a team. The administrator can add or remove players in a team after viewing player preferences. The System allow end user to view upcoming match schedules, member details and information of club. Free end user can send requests for membership or ground booking after viewing the given instructions as a help before filling the forms. The System allow users to send feedback to club. System also allow users and members to send request for booking of Diamond Ground. Duck-Worth Lewis computation can performed from all type of users. The system also provides a notification service for those who send request for membership and ground booking in case of approval or disapproval of their requests through email. Diamond Cricket Academy which provides service of trainings and online payment module through a bank or using a bank card is not include in our system scope. As this system is only build by viewing the activities of a club. The score sheets of match, details of tournaments, pitch reports, trials details are not included in the scope of our system.

1.4) Objectives

The primary objectives of this system is to provide end user to view upcoming match schedules, member profiles. End user can send membership, ground booking request and they can also book tickets for watch the scheduled match. System provide facility to members and users to send feedback for the improvement of services of club. System allow members to login their accounts provided from the administrator. The system allow administrator to keep track of club expenditures records, members profile, ground financial records as well as member financial records. Administrator can accept or reject the user request and in both case notification will send to that particular user through email. Ground booking requests also approved from administrator. If the membership request is accepted by administrator then automatically system generate password for his account for

maintaining his profile. The profile contains player play role, his personal information and major teams. All other functions are handled by system which are defined above in scope.

1.5) Report Structure

First chapter contains background history of Diamond Cricket Club. Problem definition and proposed solutions are designed for achievement of objectives. Scope of this project defines boundary of system and major functionality. At the end of first chapter, project management plan is also defined according to time and date with specifying the modules work. Second chapter is totally based on requirements gathering i.e. functional and non-functional requirements, software interfaces, communication protocols, software system attributes such as reliability, availability, security, maintainability, portability and performance of system. Also, include constraints, use cases, use case diagram, system architecture diagram, system sequence diagrams, domain model and entity relationship diagram.

1.6) Software Process Model

The software process we are going to use in this project is software development life cycle V-shaped model because project requirements are well-defined and fixed (static). The project is short and easily manageable because our project contains some modules which are defined in scope. This process model is also known as Verification and validation model and it is an extension of waterfall model which involves testing phase for each respective development stage. In my project requirements are fixed not dynamic. Project modules are defined in scope to fulfill the V-shaped model. This model also involves the testing at completion of each module. So, V-shaped model is suitable for constructing this software product.

1.7) Project Management Plan

The project management plan is description of the software approach and associated milestones according to IEEE standard. The tasks and sub tasks are scheduled with time to getting the software product at limited amount of time. The purpose of project management plan (PMP) is to document the agreed deliverables and dates.

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Analysis**	19 days	**10/6/16 8:00 AM**	**11/1/16 5:00 PM**		
2	✔	**Identify Requirements**	4 days	**10/6/16 8:00 AM**	**10/11/16 5:00 PM**		
3	✔	Review case study	1 day	10/6/16 8:00 AM	10/6/16 5:00 PM		Case study;;Abdul Rehman
4	✔	Define Requirements	2 days	10/7/16 8:00 AM	10/10/16 5:00 PM	3	;Abdul Rehman
5	✔	Meet Stake Holder	1 day	10/11/16 8:00 AM	10/11/16 5:00 PM		;;Abdul Rehman
6	✔	**Define Use cases**	5 days	**10/12/16 8:00 AM**	**10/18/16 5:00 PM**		
7	✔	Write Use cases	3 days	10/12/16 8:00 AM	10/14/16 5:00 PM		;ProjectLibre (Scheduling To...
8	✔	Draw use case diagram	1 day	10/17/16 8:00 AM	10/17/16 5:00 PM	7	;ProjectLibre (Scheduling To...
9	✔	Review use cases	1 day	10/18/16 8:00 AM	10/18/16 5:00 PM		;;ProjectLibre (Scheduling T...
10		**Develop Analysis Mode**	6 days	**10/19/16 8:00 AM**	**10/26/16 5:00 PM**		
11	✔	Develop Domain Model	1 day	10/19/16 8:00 AM	10/19/16 5:00 PM		;ProjectLibre (Scheduling To...
12	✔	Review Domain Model	1 day	10/20/16 8:00 AM	10/20/16 5:00 PM		;;Abdul Rehman
13	✔	Develop Activity Diagram	1 day	10/21/16 8:00 AM	10/21/16 5:00 PM		;ProjectLibre (Scheduling To...
14	✔	Review Activity Diagram	1 day	10/24/16 8:00 AM	10/24/16 5:00 PM		;;Abdul Rehman
15	✔	Develop System Sequenc	1 day	10/25/16 8:00 AM	10/25/16 5:00 PM		;ProjectLibre (Scheduling To...
16		Review System Sequence	1 day	10/26/16 8:00 AM	10/26/16 5:00 PM		;;Abdul Rehman
17		**Develop SRS**	4 days	**10/27/16 8:00 AM**	**11/1/16 5:00 PM**	**10**	
18		Identify non-functional re	2 days	10/27/16 8:00 AM	10/28/16 5:00 PM		;Abdul Rehman
19		Review requirements	1 day	10/31/16 8:00 AM	10/31/16 5:00 PM		;;Abdul Rehman
20		Finalize SRS	1 day	11/1/16 8:00 AM	11/1/16 5:00 PM		;Abdul Rehman
21		Analysis Phase Complete	1 day	11/2/16 8:00 AM	11/2/16 5:00 PM		
22		**Design**	13 days	**11/3/16 8:00 AM**	**11/21/16 5:00 PM**	**20**	
23		**Develop Architectural I**	2 days	**11/3/16 8:00 AM**	**11/4/16 5:00 PM**		
24		Develop Component Desi	1 day	11/3/16 8:00 AM	11/3/16 5:00 PM		;ProjectLibre (Scheduling To...
25		Review Component Desig	1 day	11/4/16 8:00 AM	11/4/16 5:00 PM		;;Abdul Rehman
26		**Develop Interface Desi**	3 days	**11/7/16 8:00 AM**	**11/9/16 5:00 PM**		
27		Design Output	1 day	11/7/16 8:00 AM	11/7/16 5:00 PM		;Abdul Rehman

Figure1.1: Scheduling of task and sub tasks for achieving milestone

According to this table first we have to understand the project requirements and scope. Second, we meet with stakeholder and club member for asking him which activities are performed within club. Third, we defines use cases with respect to the user and administrators requirements. Third, a domain model have constructed. At the end the whole project is divided into modules.

		Name	Duration	Start	Finish	Predecessors	Resource Names
28		**Design Input**	2 days	**11/8/16 8:00 AM**	**11/9/16 5:00 PM**		
29		Create Input Layout	1 day	11/8/16 8:00 AM	11/8/16 5:00 PM		;Abdul Rehman
30		**Validate Input**	1 day	**11/9/16 8:00 AM**	**11/9/16 5:00 PM**		**;Abdul Rehman**
31		Check existance of m	1 day	11/9/16 8:00 AM	11/9/16 5:00 PM		
32		Check Input Range in	1 day	11/9/16 8:00 AM	11/9/16 5:00 PM		
33		Check Transaction	1 day	11/9/16 8:00 AM	11/9/16 5:00 PM		
34		Review Interface Design	1 day	11/10/16 8:00 AM	11/10/16 5:00 PM		;;Abdul Rehman
35		**Develop Algorithm/Cor**	4 days	**11/11/16 8:00 AM**	**11/16/16 5:00 PM**		
36		Develop Interaction Diag	1 day	11/11/16 8:00 AM	11/11/16 5:00 PM		;ProjectLibre (Scheduling To...
37		Review Interaction Diagr	1 day	11/14/16 8:00 AM	11/14/16 5:00 PM		;;Abdul Rehman
38		Develop Class Diagram	1 day	11/15/16 8:00 AM	11/15/16 5:00 PM		;ProjectLibre (Scheduling To...
39		Review Class Diagram	1 day	11/16/16 8:00 AM	11/16/16 5:00 PM		;;Abdul Rehman
40		**Evaluate Design**	3 days	**11/17/16 8:00 AM**	**11/21/16 5:00 PM**	**23;26;35**	
41		Validate Requirments	1 day	11/17/16 8:00 AM	11/17/16 5:00 PM		;Abdul Rehman
42		Verify Design	1 day	11/18/16 8:00 AM	11/18/16 5:00 PM		;Abdul Rehman
43		Review	1 day	11/21/16 8:00 AM	11/21/16 5:00 PM		;;Abdul Rehman
44		Design Phase Complete	0 days	11/22/16 8:00 AM	11/22/16 8:00 AM		
45		**Code**	17 days	**11/23/16 8:00 AM**	**12/15/16 5:00 PM**		
46		**Prepare for coding**	6 days	**11/23/16 8:00 AM**	**11/30/16 5:00 PM**		
47		Choose appropriate langu	5 days	11/23/16 8:00 AM	11/29/16 5:00 PM		;;Abdul Rehman
48		Create a set of unit tests	1 day	11/30/16 8:00 AM	11/30/16 5:00 PM		IDE (Programming Tool);;Ab...
49		**Start Coding**	10 days	**12/1/16 8:00 AM**	**12/14/16 5:00 PM**	**46**	
50		Write code (Create front	7 days	12/1/16 8:00 AM	12/9/16 5:00 PM		IDE (Programming Tool);;Ab...
51		Properly document code	1 day	12/12/16 8:00 AM	12/12/16 5:00 PM	50	IDE (Programming Tool);;Ab...
52		Connect Database	2 days	12/13/16 8:00 AM	12/14/16 5:00 PM	51	IDE (Programming Tool);;Ab...

Figure 1.2: Scheduling of task and sub tasks for achieving milestone

According to this table, the process of design has been started after making the domain model and user requirements. In this process, the software interfaces are built by using tools which shows colors schemes and forms with respect to user requirements. After designing next process is coding and setting the tools like WAMP server for development of project.

53		**Validate**	**1 day**	**12/15/16 8:00 AM**	**12/15/16 5:00 PM**	**49**	
54		Review & Refactor code	1 day	12/15/16 8:00 AM	12/15/16 5:00 PM		;Abdul Rehman
55		Run Unit tests to check c	1 day	12/15/16 8:00 AM	12/15/16 5:00 PM		;Abdul Rehman
56		Coding Phase with Static Tes	1 day	12/16/16 8:00 AM	12/16/16 5:00 PM		
57		**Test**	**2 days**	**12/16/16 8:00 AM**	**12/19/16 5:00 PM**	**53**	
58		Execute code (dynamic tes	1 day	12/16/16 8:00 AM	12/16/16 5:00 PM		IDE (Programming Tool);;Ab...
59		Remove errors	1 day	12/19/16 8:00 AM	12/19/16 5:00 PM	58	
60		Project Complete	1 day	12/19/16 3:00 PM	12/20/16 3:00 PM		

Figure 1.3: Scheduling of task and sub tasks for achieving milestone

According to this table, the whole project is implemented at that time in the previous table. But now we start testing our product whether it fulfill user requirements or not. For this process, we have to write some test case for validating the system behavior on different inputs. As the testing reveals errors or bugs in the system in specific modules. So, if any error appear during running the system it will be fixed.

In the next chapter we will discuss the functional and non-functional requirements, attributes and features of system. User interface and system interface will also discussed. Software constraints like reliability, accessibility, portability, performance and security features also discussed later. At the end user constraints, use case and domain model for showing association between real-world entities.

"The hardest single part of building a software system is deciding what to build."

By: Fred Brooks (1931-Todate)

Chapter 2:

Requirements Gathering and Analysis

This chapter contains description of functional and non-functional requirements and user interfaces. As this system works on the role of client server so some communication protocols details involves in this chapter. Product or system features and attributes plays a vital role in an organization for which system is build. The software system attributes such as reliability, security, accessibility, portability and performance are discussed. At the end of chapter constraints, use cases, use case diagram and system sequence diagrams.

2.1) Specific Requirements

The purpose of this Specific Requirement Specification is to clear the requirements of the system and to decide what the system should do and what the system should not do. Members, end users and administrator are interact with system. Three types of users will interact with system.

1. Members who are registered from club and they have member id and password assigned from club
2. End users who interact with system for viewing information about match, previous awards records and upcoming match events and they do not possess the technical understanding.
3. The administrator which plays a vital role. He has rights to change the contents of each records

2.1.1) Functional Requirements

The functional requirements for a system describe what system should do [1]. The system allow end users and members to send request for booking of cricket ground if administrator accept or reject that request then notification will send through email. The System allow end users to apply for membership. The system also allow administrator to approve or disapprove that membership request, in both case notification will send to that user through email. The system allow administrator to maintain data of members, teams and their monthly financial records of club expenditures. The system allow administrator to perform basic operation on the details of member such as add, update, search, view and delete. The system allow all users to perform Duck-Worth Lewis computations. The system allow members to login. The system allow administrator to select the team and to schedule or reschedule the match. The system allow users to send feedback to club. The administrator can add payment record of employees and registered members on every month. The system allow administrator to add or remove a member from system. The system allow administrator to accept or reject the requests regarding to membership or ground booking and a notification will send to that user through email. Administrator can add, remove or update the employee details like name, type of employee and his salary. Administrator can allocate or deallocate scheduled match tickets. By which client book tickets online by entering their information. If someone physical pay amount to administrator then he will purchase ticket. If someone first book his ticket then its ticket status will remain booked until he gave amount to administrator.

2.1.2) Non Functional Requirements

Non-functional requirements that are not directly concerned with the specific services delivered by the system to its users. They may relate to emergent system properties such as reliability, response time and store occupancy [2]. Only the registered members can login to system. We supposed we have n number of registered member (n=200). Any member can play role of umpire assigned from the administrator. One team contains 15 members including captain of team. For ground booking free end user must have the valid email address. For membership request user must have valid email address. The notification will send to every member at the end of each month with the permission of administration. Member or other user can request for ground booking. Ground booking requests and membership requests only for the people of RWP and ISB. As the Diamond Cricket Academy is out of scope in this project so that there is no role of coach.

2.2) Software Interfaces

Software interface specify the use of other required software products and interfaces with other application system.

1. Web Browser
    - Chrome
    - Baidu
    - Mozilla Firefox
2. Windows Operating System

- Linux
- Windows
- Linux
- MacOs -Apple
- Unix

3. Mobile Operating System (for mobile users)
   - Android OS
4. WAMP Server
   - Version 2.4

2.4) Communication Protocols

In this application, three communication protocols play an important role for send data from client to server and access information in encrypted form which provides us reliability and security of data.

1. HTTP (Hyper Text Transfer Protocol)
2. SSL (Secure Socket Layer)
3. SMTP (Simple Mail Transfer Protocol)

HTTP has four methods (GET, PUT, DELETE and POST) which are used to access the contents of information resource through web. SSL (Secure Socket Layer) is the standard security technology for building an encrypted link between a web server and a browser. Simple Mail Transfer Protocol is use to send notification through email address.

2.5) Software Product Features

This system used to automate basic operation of Diamond Cricket Club. It provides an automate system for maintaining member profiles data, their financial records, team selection tool for selection of members and ground reservation details. The system allow new client to get membership in club after approval from administrator staff. It keep track of winning team details and upcoming matches. It allow users to book match tickets and maintain match tickets payment details. Administrator can add, remove or update the member payments records and club employees are manage by this system. At last it allow users to send feedback to club, also system send notifications to members.

2.6) Software System Attributes

2.6.1) Reliability

Reliability is like a trust on system that it will return right response. This system is reliable, it always shows the correct response to user in specified amount of time depend upon the entered query. Our system is 80% reliable. Reliability can also loss by changing in code and internet speed.

2.6.2) Availability

Availability means the system will able to use any time. And any user can use or access the system.

2.6.3) Security

The system can secure in some ways for the improvement of services of club. System provides authentication and authorization process for security. Authentication means giving access rights of resources anyone while the authorization is the process of verifying that you have access to something. The member have member id and password and they have right only to access their profile allowed from administrator. System perform authorization when the member logged in. More than one administrator can access system. Only the registered member can access his profile. If he doesn't follow club rules.

2.6.4) Maintainability

The ease and speed of system with which system return to its operational status after a failure occurs. The system has database for storing and retrieving records if any query failed then it will automatically maintain itself. If user enters wrong query then system will prompt user to enter correct input. E.g. if someone try to use same email or phone number two times. Then on second time, form will not submitted and system will prompt user this phone number or email is already taken.

2.6.5) Performance

Performance measures the speed of system by performing operations. The system have the backup server which update when the server startup if system crash due to virus or SQL injection then it will automatically recover data from backup in few seconds. Login operation takes three mili-seconds. Performance is dependent on the network condition and size of records fetched. More details of performance is discussed in fifth chapter.

2.7) Constraints

It is assumed that the user have internet connection and web browser for interacting system. He can access system through Wi-Fi.

2.8) Use Case Description

A use case is a text narrative or template that describes a system function or feature from the user's point of view. The description of each use case is given in form of table named as fully dressed template and the role of that every particular use case is also given at the end of each use case table [3].

2.8.1) Login to Profile

Name	UC1: Login to profile
Primary Actor	Member, administrator
Pre-Conditions	User is registered to club.
Post-Conditions	User will logged in.
Main Scenario	1. User enters member id and password. 2. User clicks on sign in button. 3. User will validate member id and password.
Alternative Scenario	1a. User entered wrong member id and password. 1. Enter password less than 10 letters. 2. Wrong id or password. 2. User not clicked on button.
Frequency	Many times in a week.

Table 2.1: Login

This use case says that how the member and administrator will logged in to system. This given actions are performed by user.

2.8.2) View Upcoming Match List

Name	UC2:View upcoming match list
Primary Actor	User, Member
Pre-Conditions	User is successfully opens the main page of site with any web browser.
Post-Conditions	User will viewed given upcoming matches.
Main Scenario	1. The home page will appear. 2. User click on upcoming match button. 3. A list of upcoming matches with details will appeared on screen.

Alternative Scenario	2a. User not clicked on button. 3. User will not able to view list of matches. 1. Database error message.
Frequency	Many times in a week.

Table 2. 2: View upcoming match list

This use case describes that how a member or any client can view the list of upcoming match. This use case will use by any end user or member also.

2.8.3) Approval of Membership Request

Name	**UC3: Approval of membership request**
Primary Actor	Administrator
Pre-Conditions	User is logged in. The clients who are sending request for membership must have valid email address.
Post-Conditions	User will successfully approve membership request after reading member information.
Main Scenario	1. User will approves membership requests after reading his information. 2. User will send notification to that client. 3. Notification goes to email address.
Alternative Scenario	1a. User will rejects client request. 1. Client enter incomplete and wrong information.
Frequency	Many times in a Month.

Table 2.3: Approval of membership Request

This use case says that how the administrator will accept the membership form of client. This use case will used by administrator.

2.8.4) Apply for Membership

Name	UC4: Apply for membership
Primary Actor	End User
Pre-Conditions	User is on the page of membership form. Only end user can apply for membership.
Post-Conditions	User will successfully apply for membership.
Main Scenario	1. User enters required details. 2. User clicks on send button. 3. System will validate all entered inputs fields. 4. After that a message will show on that page about the acknowledgement that you form is successfully send to club.
Alternative Scenario	2a. User not click send button. 1. User click on reset button. 3a. Invalid input fields. 1. Invalid email address. 2. Invalid name.
Frequency	Many times in a week.

Table 2.4: Apply for membership use case

This use case tells the user how to send membership form to club and which actions are need to send successfully.

2.8.5) Send Request for Booking of Ground

Name	UC5: Send request for booking of ground
Primary Actor	End User, Member
Pre-Conditions	User is on the page of Booking of Ground.
Post-Conditions	User will successfully send request for booking of ground at particular date and time.

Main Scenario	1. User will read all instructions and availability of ground with specific time and date. 2. User entered all input fields. 3. User will clicks on send button. 4. System will validate entered inputs. 5. A message will appear on screen for acknowledgement.
Alternative Scenario	2a. User will enter data due to keyboard error. 3a. User will not click on send button. 1. User will click on reset button. 4a. Invalid input fields 1. Invalid email address. 2. Invalid name.
Frequency	Many times in a week.

Table 2.5: Send request for ground booking

This use case tells us how members and users can send request for booking of ground at particular date and time.

2.8.6) Approval of Ground Booking Request

Name	UC6: Approval of ground booking request
Primary Actor	Administrator
Pre-Conditions	User logged in to system.
Post-Conditions	Request of ground booking is successfully approved.
Main Scenario	1. User will opened the end user request. 2. User will read the given information. 3. User will approved that request.

Alternative Scenario	1a. End user Request is not opened. 1. Empty list of request. 2. No request found 2a. User will not read information. 1. Without reading user will reject that request. 3a. User will not approved request due to wrong information.
Frequency	Several times in a week.

Table 2.6: Approval of ground booking request

This use case says that how the administrator will approved the request of end user for ground booking.

2.8.7) Using Duck-Worth Lewis (DL) Calculator

Name	UC7: Using DL calculator
Primary Actor	User, members, administrator
Pre-Conditions	User is on the page of DL calculator.
Post-Conditions	User will successfully use DL calculator.
Main Scenario	1. User entered the required input values. 2. User will clicks on calculate button. 3. System validate inputs. 4. Results will successfully computed.
Alternative Scenario	1a. User will not entered input values. 1. User have one missing value. 2a. User will not clicks on calculate button. 1. User will clicks on reset button instead of calculate button. 3a. Validation error 1. System failed to compute results. 2. Enter the scores of team A.

	3. Wrong input entered. 4. Re-enter all input to compute results. 4a. Result will not come.
Frequency	Several times in a week.

Table 2.7: Using Duck-Worth Lewis Calculator

This use case tells the series of actions done from the end user, members and administrator. It also describes how to compute use the DL calculator in case of weather or any emergency circumstances.

2.8.8) Selection of Team

Name	UC8: Selection of team
Primary Actor	Administrator
Pre-Conditions	User is logged in to system. Few number of teams already exists.
Post-Conditions	User will successfully select team for match.
Main Scenario	1. User will clicks on Select Team button. 2. A list of teams will appear on screen. 3. User will select the particular team for specific match according to with respect to chosen day and time.
Alternative Scenario	1a. User will not clicks on Select Team button. 1. User will clicks another button.
Frequency	Many times in a month.

Table 2.8: Selection of team

This use case describes that how an administrator will able to select team for particular match. It also defined the number team list already stored in database and only administrator have right to select team from them.

2.8.9) Send Notification to End-user

Name	UC9: Send notification to end users
Primary Actor	Administrator
Pre-Conditions	User is logged in to system. Valid email address of clients.
Post-Conditions	Notification successfully send to clients whose send request.
Main Scenario	1. User approve or disapprove the requests of clients regarding to booking of the Diamond Ground and membership in club. 2. A notification sent to that clients through email.
Alternative Scenario	2a. Notification will not send to clients. 1. Email not exist.
Frequency	Many times in a week.

Table 2.9: Send notification to end user

This use case describes how to send notification to clients who has sends requests to club for membership or Ground booking. This actions will performed by administrator.

2.8.10) Scheduling of Matches

Name	UC10: Scheduling of matches
Primary Actor	Administrator
Pre-Conditions	User will logged in to system. Teams list must exists.
Post-Conditions	Match scheduled successfully according to specific time and availability of ground with particular team.

Main Scenario	1. User will clicks on Schedule Match button. 2. User will select two teams. 3. User will select day and time according to the availability of ground. 4. Successfully match will scheduled.
Alternative Scenario	1a. User will not clicks on Schedule Match button. 2a. The second team not exists. 1. A message will display on screen first add second team to database. 3a. User will not select day because that day ground is not available.
Frequency	Many times in a month.

Table 2.10: Scheduling of matches

This use case describes that how administrator will schedule the match between two teams which must have to exist in database.

2.8.11) Send Feedback to Club

Name	UC11: Send feedback to club
Primary Actor	User, member
Pre-Conditions	User must have on the page of feedback on browser.
Post-Conditions	User will successfully send feedback to club.
Main Scenario	1. User will enter all input fields. 2. User clicks on send button. 3. System will validate all input fields. 4. Feedback will successfully send to club and a message will appear on screen for acknowledge.

Alternative Scenario	1a. User will not enter all inputs. 1. User entered few input fields. 2a. User will not clicks on send button. 1. User will clicks on reset button.
Frequency	Many times in a week.

Table 2.11: Send feedback to club

This use case describes how to send feedback, suggestions and comments to club. Any end user can send notification to club after that administrator will process these comments for the improving the services of club.

2.8.12) View Member Profile

Name	UC12: View member profile
Primary Actor	End user, member, administrator
Pre-Conditions	User is on the page on member profile.
Post-Conditions	User will successfully view member profile.
Main Scenario	1. User viewed the member profile. 2. User will access particular member's information like his records, centuries and other relevant details.
Alternative Scenario	1. User will not viewed any member profile.
Frequency	Many times in a week.

Table 2.12: View member profile

This use case describes how the end user or member can view details of members whose are registered to club. Administrator can view the profile of registered members.

2.8.13) Log Out

Name	UC13: Log out
Primary Actor	Administrator, member
Pre-Conditions	User is logged in.
Post-Conditions	User successfully logged out.
Main Scenario	1. User clicks on log out button. 2. A message will appear on screen that user is successfully logged out.
Alternative Scenario	1a. Users will not clicks on log out button.
Frequency	Many times in a day.

Table 2.13: Log out

This use case describes that how the administrator and member will logged out from the system. A message will appear for user that he is successfully logged out from the system.

2.8.14) Search Member Profile

Name	UC14: Search member profile
Primary Actor	End user, member, Administrator
Pre-Conditions	User is on the page on member profile.
Post-Conditions	User will successfully searched the desired member profile.

Main Scenario	1. User viewed the member profile. 2. User will access particular member's information like his records by choosing the player role. Such as Batsman, Bowler, All rounder 3. A list of relevant name will appear on screen.
Alternative Scenario	1a. User will not viewed any member profile. 1. Administrator will not allow end user to access member details. 2a. Invalid member id or name. 1. Re-enter member id or name.
Frequency	Many times in a week.

Table 2.14: Search member profile

This use case describes how the end user, administrator, member can search a particular member from list of members.

2.8.15) Edit Member Profile

Name	UC15: Edit member profile
Primary Actor	Member
Pre-Conditions	Only registered member can edit profile. User must be logged in to system.
Post-Conditions	User will successfully edit his profile.
Main Scenario	1. User edit the desired details to his profile according to given rights. 2. User can update his account password by entering old password.
Alternative Scenario	1a. Profile will not edit due to wrong old password.
Frequency	Several times in a month.

Table 2.15: Edit member profile

This use case describe the functionality of member profile which can be edit by administrator and member. Both the users can access profile and everyone must edit information according to given rights.

2.8.16) Delete Match Schedule

Name	UC16: Delete Match Schedule
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system.
Post-Conditions	User successfully delete match schedule.
Main Scenario	1. User clicks on match schedule. 2. A list of all match schedule appear. 3. After selection of particular match schedule or he can search a match schedule. 4. User will clicks on delete button. 5. A message will appear for acknowledgement.
Alternative Scenario	1a. User will not clicks on match schedule. 2a. List of match schedule will not appear. 1. Schedule list is empty. 3a. User will not clicks on delete button.
Frequency	Several times in a month.

Table 2.16: Delete match schedule

This use case describe that how a particular match schedule will delete from the list of match schedule. Only administrator can delete match schedule.

2.8.17) Reject Membership Request

Name	UC17: Reject membership request
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system.
Post-Conditions	User will successfully reject membership request.

Main Scenario	1. User clicks on membership button. 2. All the membership requests will appear. 3. User select a particular request and he will read entered information. 4. User will reject his request. 5. A notification will send to that end user with valid reason.
Alternative Scenario	1a. User will not click on membership button. 2a. No membership request will appear. 1. Request list is empty. 3a. User will not read entered information. 4a. User will not reject membership request. 5a. Notification will not sent to end user and message will appear.
Frequency	Several times in a month.

Table 2.17: Reject membership request

This use case describe that how administrator will reject the membership request of end users whose are interested to take membership of club. Only the administrator of club have right to accept/reject requests.

2.8.18) Reject Ground Booking Request

Name	UC18: Reject ground booking request
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system.
Post-Conditions	User will successfully reject ground booking request.
Main Scenario	1. User clicks on ground booking button. 2. All the ground booking requests will appear. 3. User select a particular request and he will read entered information. 4. User will reject his request. 5. Notification will sent and message will appear.
Alternative Scenario	1a. User will not click on ground booking button. 2a. No membership request will appear. 1. Request list is empty. 3a. User will not read entered information. 4a. User will not reject ground booking request.

	5a. Notification will not sent to end user.
Frequency	Several times in a month.

Table 2.18: Reject ground booking request

This use case describes how administrator will reject the ground booking request of user. The notification will successfully send to user on his entered email when administrator will reject his request.

2.8.19) Allocate Match Tickets

Name	UC19: Allocate match tickets
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system.
Post-Conditions	User will successfully allocate tickets of scheduled match.
Main Scenario	1. User clicks on schedules matches. 2. All the scheduled matches will appear. 3. User will click on allocate tickets for match. 4. A form will appear which takes scheduled matches, number of tickets and price of ticket. 5. A message of successfully allocated ticket will appear after entering the details and submitting.
Alternative Scenario	1a. User will not click on allocate tickets for match. 2a.No match schedule is found. 3a.User will not allocate tickets for scheduled match.
Frequency	Several times in a month.

Table 2.19: Allocate match tickets

This use case describes how the tickets are allocated for the scheduled match. Client will view match lists with match details. Then they can also book tickets to watch a match at specific time, date and

venue. Administrator can allocate or deallocate tickets for clients depend upon the number of stakeholder and demand of clients.

2.8.20) Add Employee Detail

Name	UC20: Add employee detail
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system.
Post-Conditions	User will successfully add employee details in system.
Main Scenario	1. User will click on employee management button. 2. At that page, he will click on add employee button. 3. A form will appear for taking the information of employee. 4. User will enter employee name, contact number, salary, club joining date, date of birth and his type. 5. After entering his basic information he will click on add button.
Alternative Scenario	1a. User will not click on employee management button. 2a. User will not click on add employee button. 4a. User will not enter employee details.
Frequency	Several times in a month.

Table 2.20: Add employee details

This use case describes how administrator will add an employee to club. Administrator can add, delete, and update the employee information.

2.8.21) Add Member Payment

Name	UC21: Add member payment
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system. The club must have registered members.
Post-Conditions	User will successfully add payment record of registered member.

Main Scenario	1. User clicks on add member payment. 2. Then form will appear in which a list of all registered members is shown then user will select a member. 3. After that, user will select month and year of payment. 4. He will click on save record button. 5. The payment record selected member have successfully added.
Alternative Scenario	1a. User will not click on add member payment. 2a. The registered members list will not appear. 3a. User will not choose year and month of payment. 4a. User will not click in save record button. 5a. Payment record will not save.
Frequency	Several times in a month.

Table 2.21: Add member payment

This use case describes how administrator will add payment record of registered member by selecting specific month and year. Beside this, administrator can view and delete the payment record of members.

2.8.22) Add Employee Payment

Name	UC22: Add employee payment
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system. The club must have employees.
Post-Conditions	User will successfully add payment record of club employee.
Main Scenario	1. User clicks on add employee payment. 2. Then form will appear in which a list of all employee name and type of employee will show then user will select a type of employee. 3. After that, according to that employee type the name of all employees show who have the specific chosen type. 4. He will choose specific month and year. 5. He will click on save record. 6. The payment record selected employee have successfully added.

Alternative Scenario	1a. User will not click on add employee payment. 2a. The club employees type and names will not shown 1) He will not select employee type. 3a. User will not choose year and month of payment. 4a. User will not click in save record button. 5a. Payment record will not save.
Frequency	Several times in a month.

Table 2.22: Add employee payment

This use case describes how administrator will add payment record of club employee by selecting specific month, year and employee type. Beside this, administrator can view and delete the payment record of employee.

2.8.23) Add Monthly Expenditures

Name	UC23: Add monthly expenditures
Primary Actor	Administrator
Pre-Conditions	User must be logged in to system.
Post-Conditions	User will successfully add monthly expenditures of club.
Main Scenario	1. User clicks on monthly club expenditures. 2. Then a table will appear in which a list of all club expenditures with name and type of expenditures will show. 3. After that, user will clicks on add expenditures button. 4. A form will appear with two field, one for expenditures type and other for amount of expenditures. 5. After entering date in two fields. He will clicks on add button. 6. The expenditures record will successfully added.
Alternative Scenario	1a. User will not click on add monthly club expenditures. 2a. The expenditures table will not show on that page. 3a. User will not clicks on add expenditures button. 4a. User will not enter input fields. 5a. Expenditures record will not save.
Frequency	Several times in a month.

Table 2.23: Add monthly expenditures

This use case describes how administrator will add the club monthly expenditures such as total monthly cost of ground maintenance, or electricity, gas bill etc. The administrator can add, update and delete any expenditures records.

2.8.24) Book Match Ticket

Name	UC24: Book match ticket
Primary Actor	Members, user
Pre-Conditions	User must be in view match schedule page.
Post-Conditions	User will successfully add book ticket of specific match.
Main Scenario	1. User clicks on match schedules link. 2. Then all the upcoming or schedules matches will show on that page if administrator have scheduled any match. 3. After that, user will read the details of match with competing teams, names, match overs, match time with date, price of ticket, total number of tickets and number of remaining tickets. 4. He will click on book ticket button. 5. After entering his details he will click on book button. 6. The match ticket will successfully booked.
Alternative Scenario	1a. User will not click on match schedule. 2a. The upcoming match schedules will not appear. 3a. The match details will not show to user. 4a. User will not enter input fields. 5a. Match ticket will not booked.
Frequency	Several times in a month.

Table 2.24: Book match tickets

This use case describes how stakeholder and users will book a ticket for watching a match. After booking they have to pay price of ticket to administrator for purchasing that booked ticket. After receiving the amount the administrator changes the ticket status from booked to sold.

2.9) Use Case Diagram

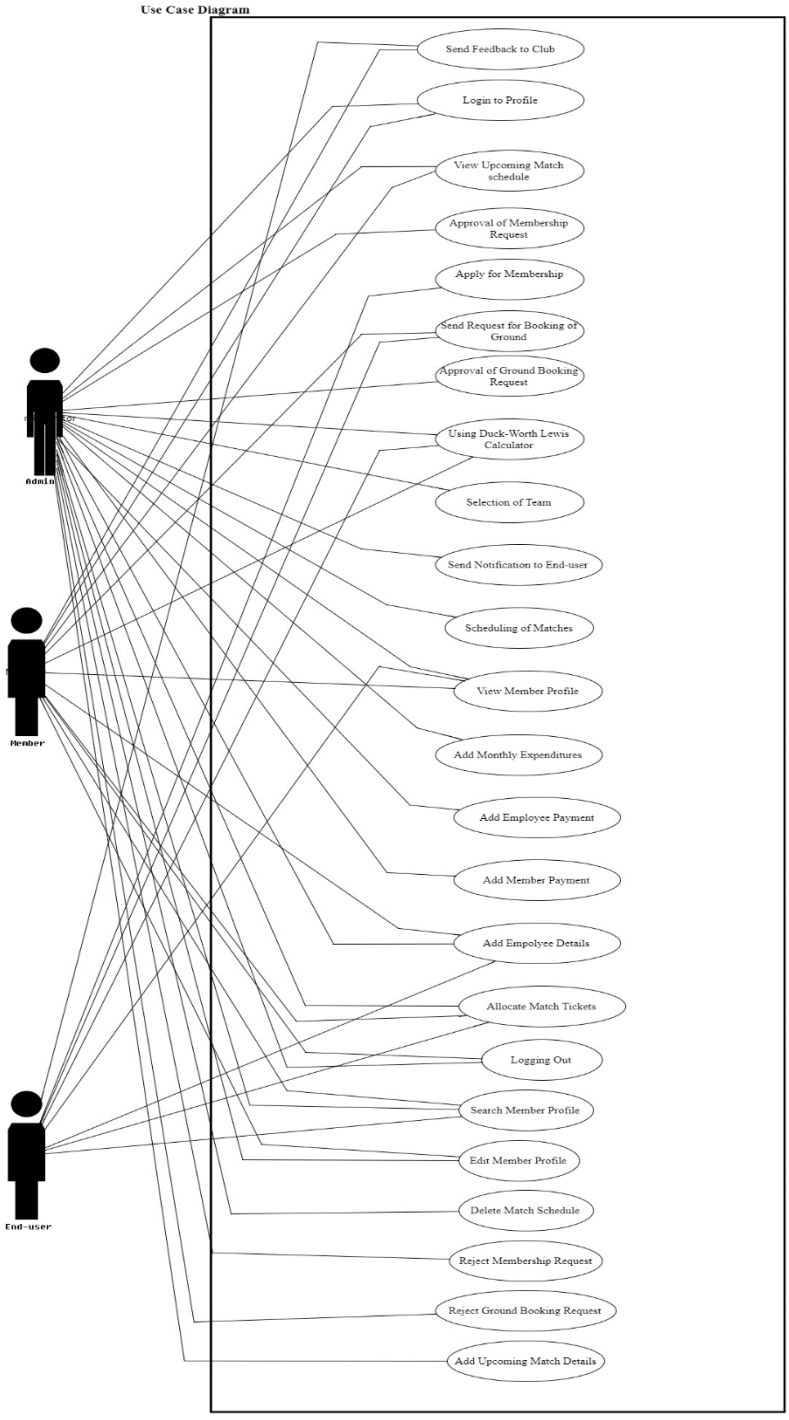

Figure 2.1: Use Case Diagram

2.10) System Sequence Diagram

A system sequence diagram (SSD) describes interaction of an actor and system. It also shows the particular scenario of a use case events are triggered by the external actor. The system to system interaction is possible in SSD. SSD explains behavior of a system. An SSD is constructed by the inception of a use case with main success scenarios.

SSD-1) Login to Profile

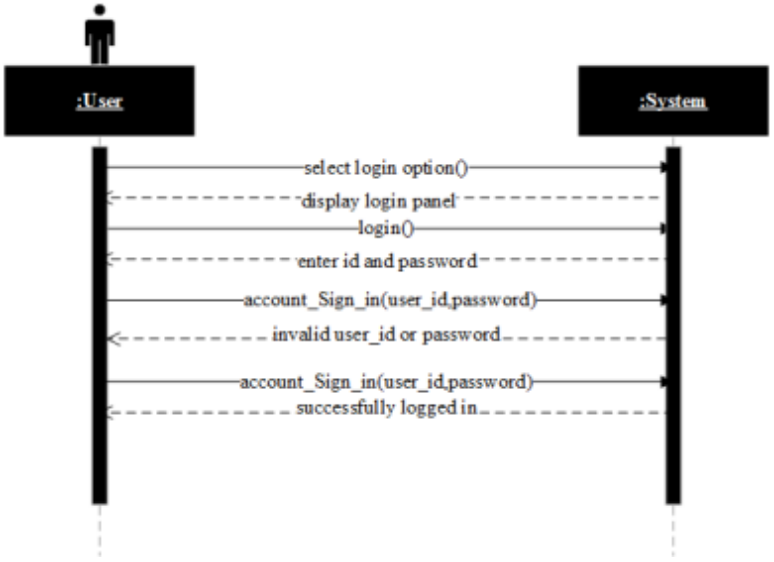

Figure 2.2: SDD for Login

SSD-2) View Upcoming Match List

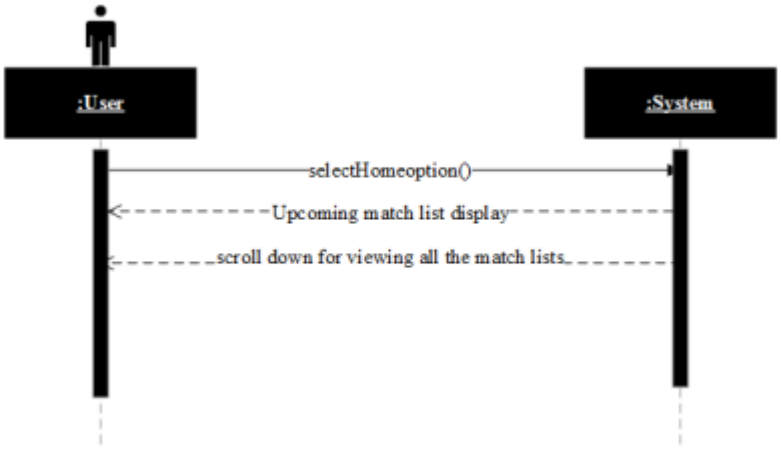

Figure 2.3: SSD for View Upcoming Match List

SSD-3) Approval of Membership

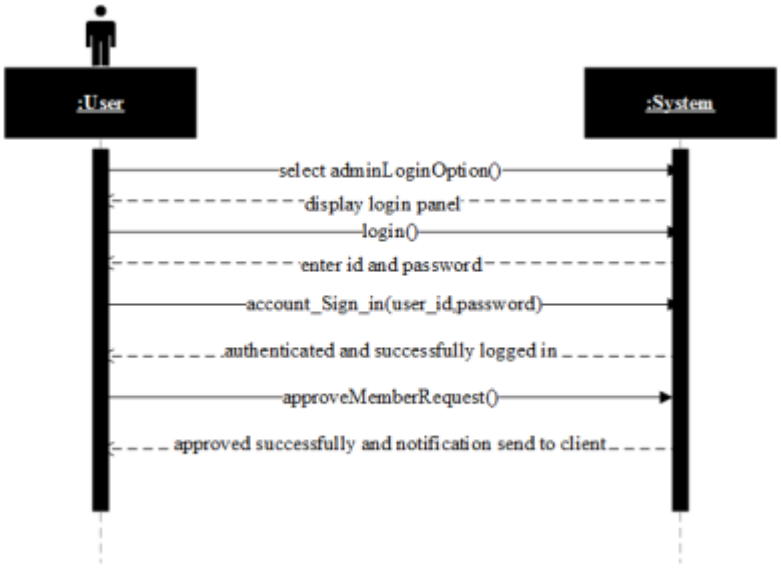

Figure 2.4: SDD for Approval of Membership

SSD-4) Apply for Membership

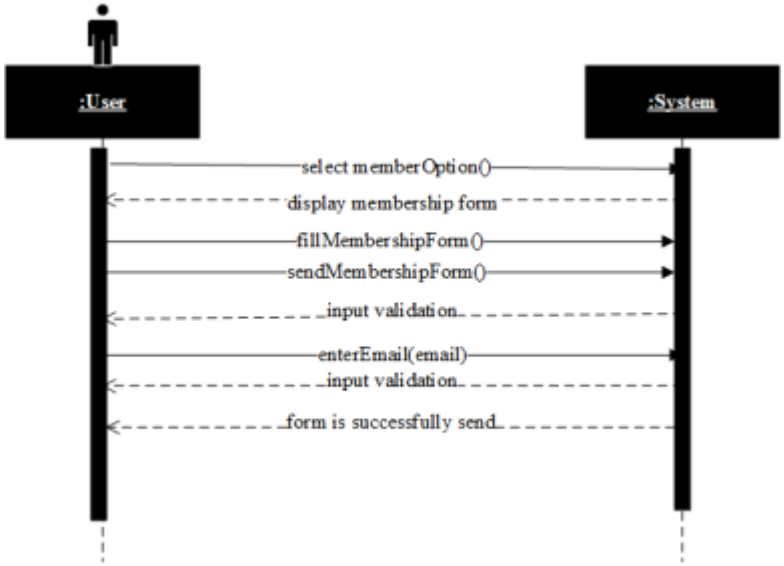

Figure 2.5: SSD for Apply for Membership

SSD-5) Send Request for Ground Booking

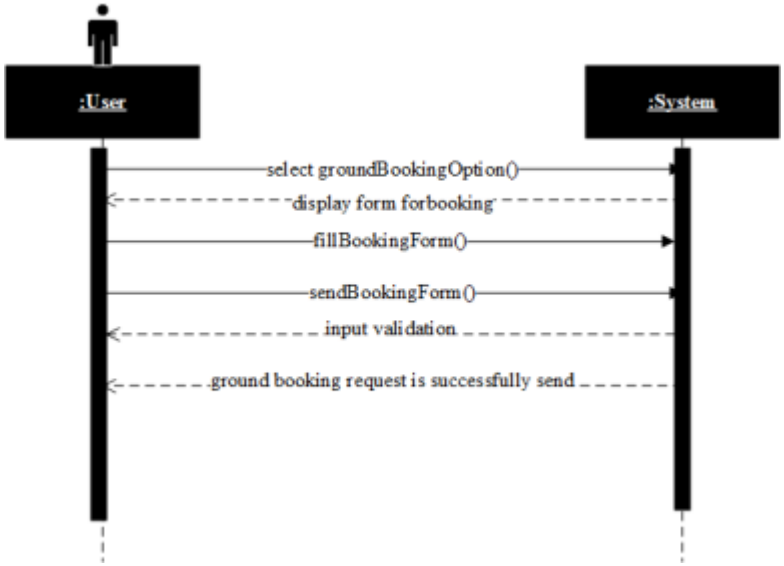

Figure 2.6: SDD for Sending Request for Ground Booking

SSD-6) Approval of Ground Booking Request

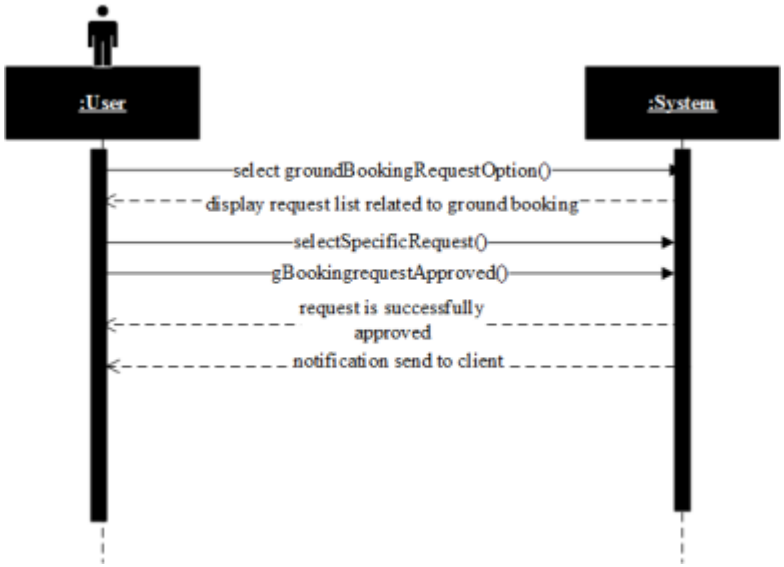

Figure 2.7: SSD for Approval of Ground Booking Request

SSD-7) Using Duck-Worth Lewis Calculator

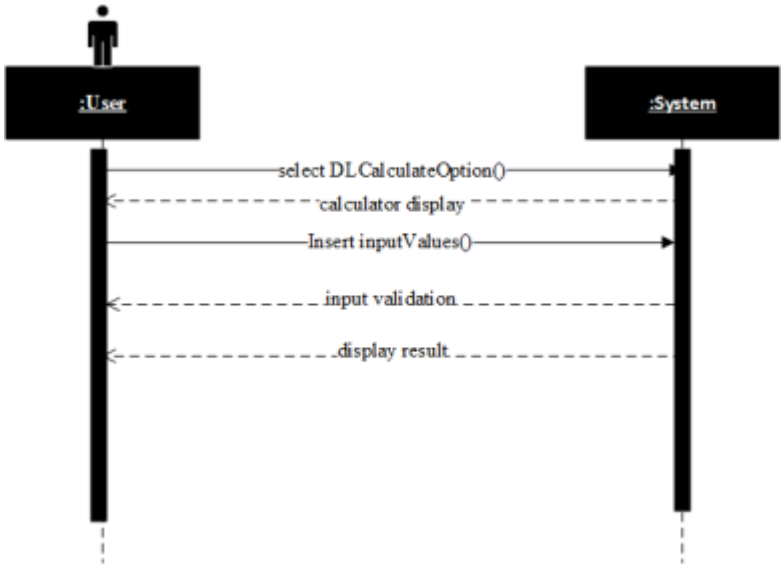

Figure 2.8: SDD for Using Duck-Worth Lewis Calculator

SSD-8) Selection of Team

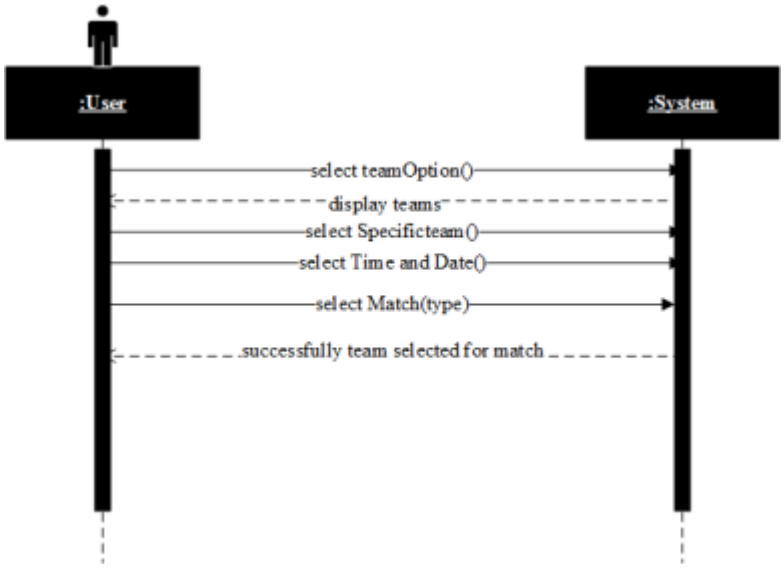

Figure 2.9: SSD for Selection of Team

SSD-9) Send Notification to End-user

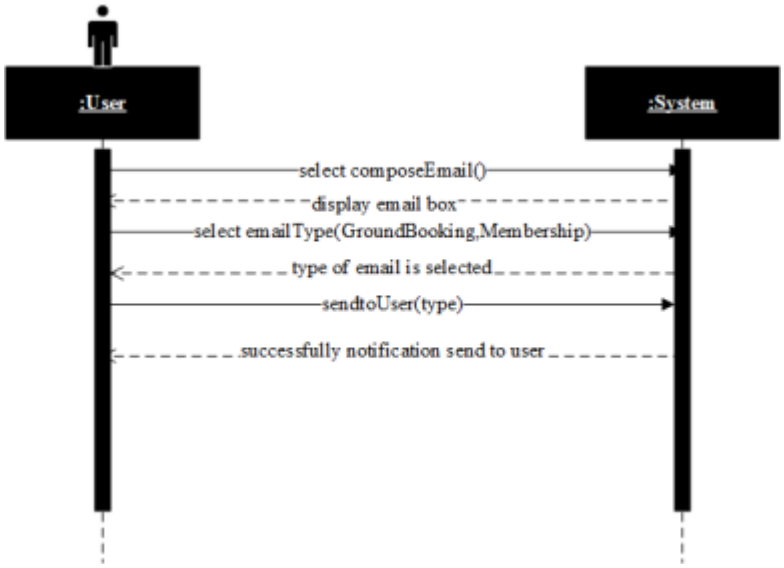

Figure 2.10: SDD for Send Notification to End-user

SSD-10) Scheduling of Match

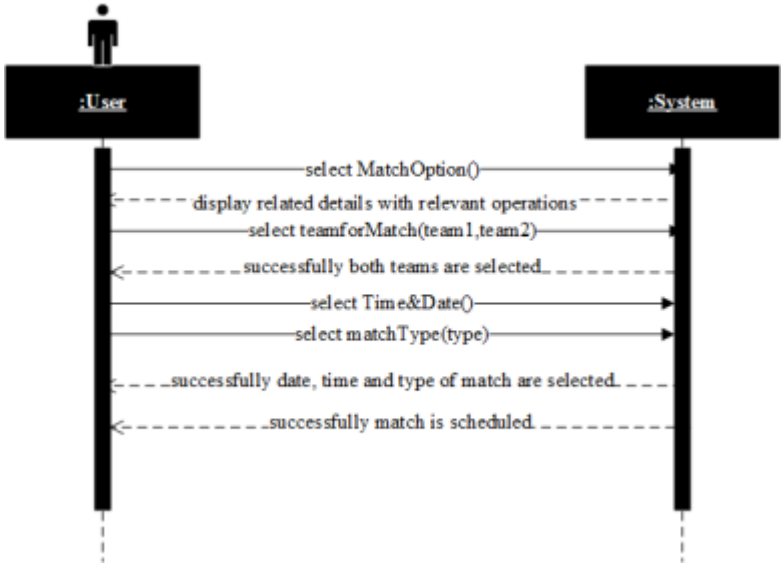

Figure 2.11: SDD for Scheduling of Match

SSD-11) Send Feedback to Club

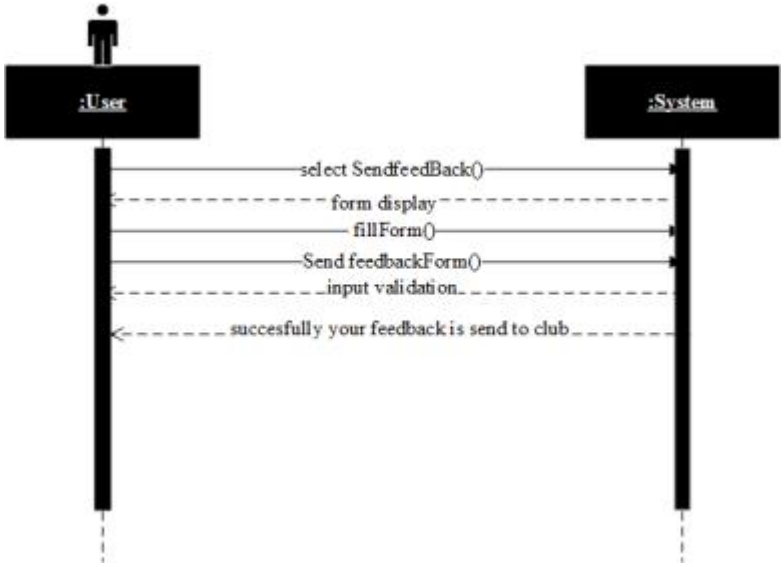

Figure 2.12: SSD for Send Feedback to Club

SSD-12) View Member Profile

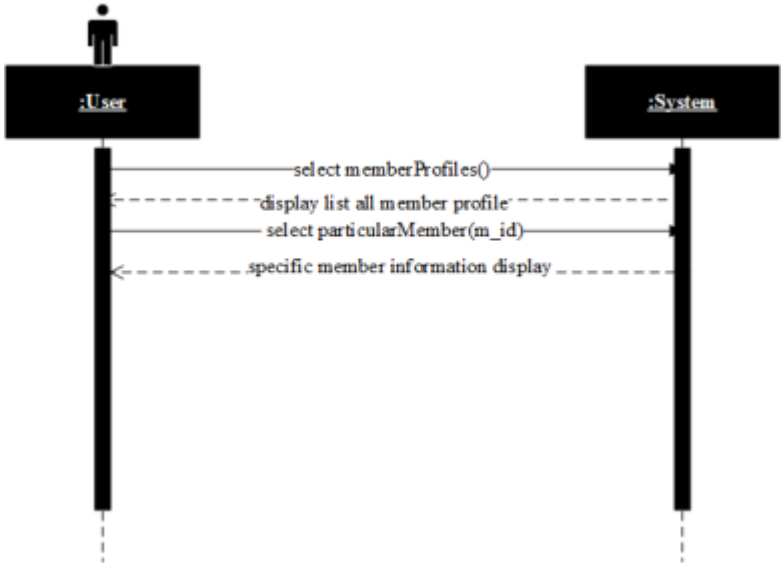

Figure 2.13: SDD for View Member Profile

SSD-13) Logging Out

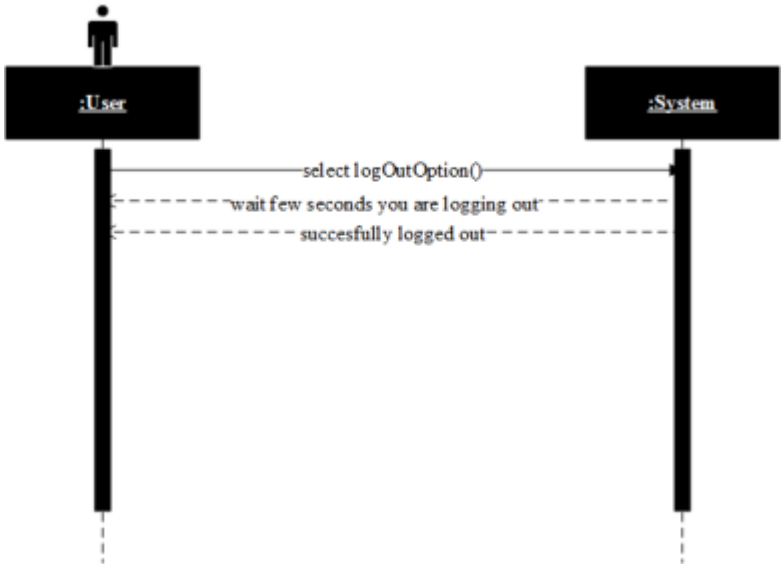

Figure 2.14: SDD for Logging Out

SSD-14) Search Member Profile

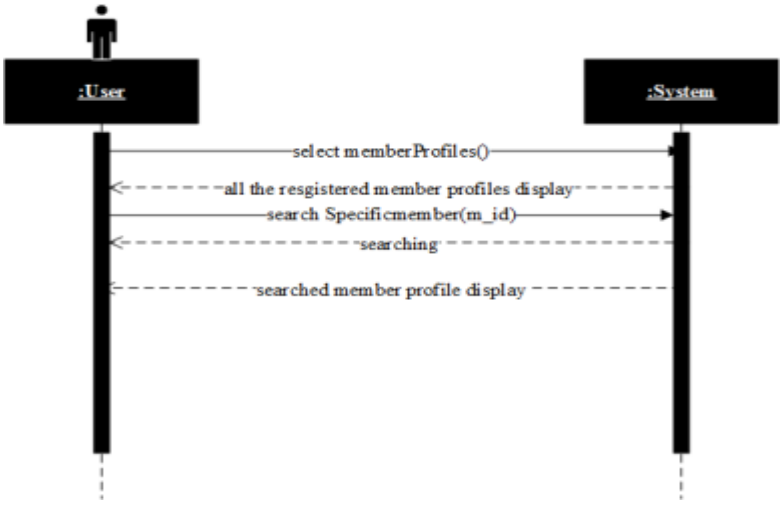

Figure 2.15: SSD for Search Member Profile

SSD-15) Edit Member Profile

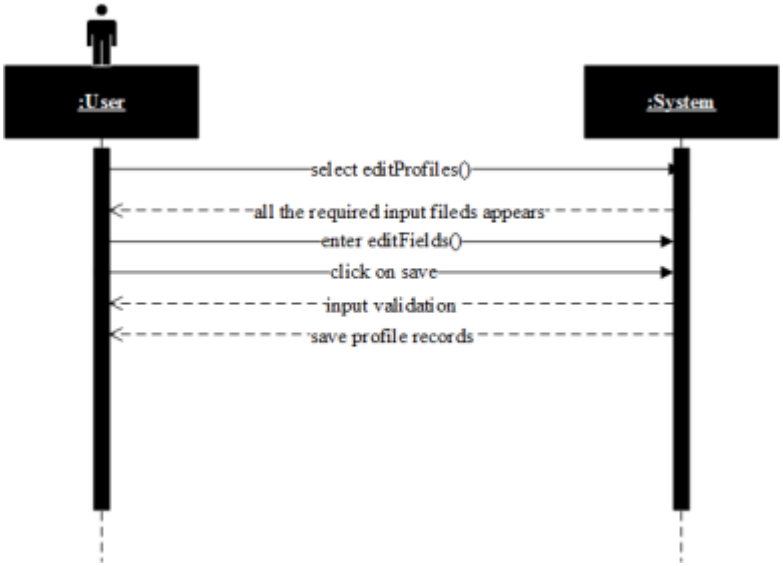

Figure 2.16: SSD for Edit Member Profile

SSD-16) Delete Match Schedule

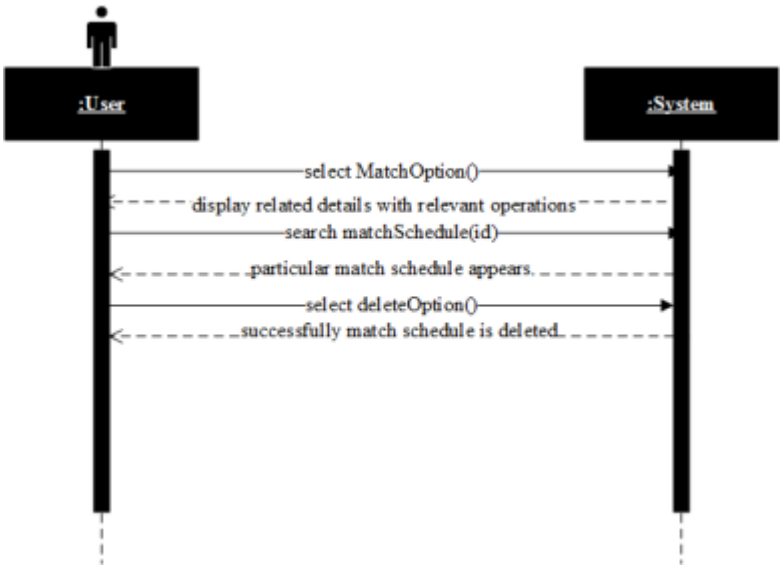

Figure 2.17: SSD for Delete Match Schedule

SSD-17) Reject Membership Request

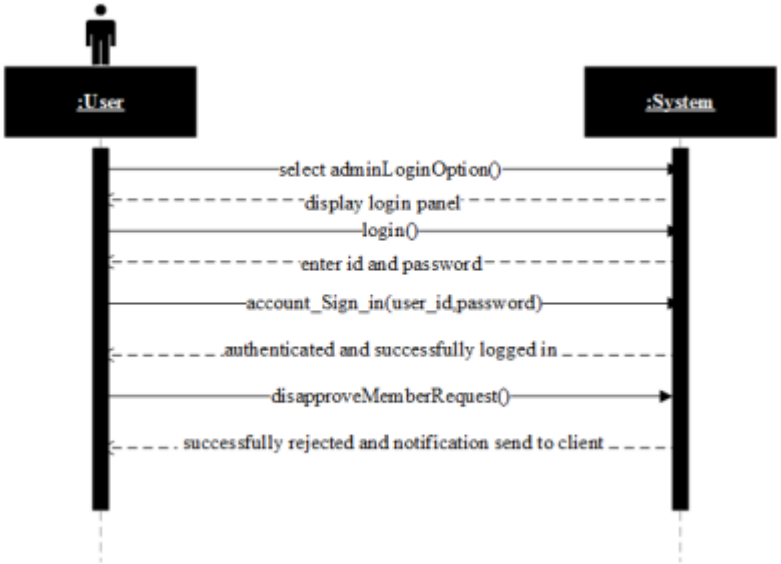

Figure 2.18: SDD for Reject Membership Request

SSD-18) Reject Ground Booking Request

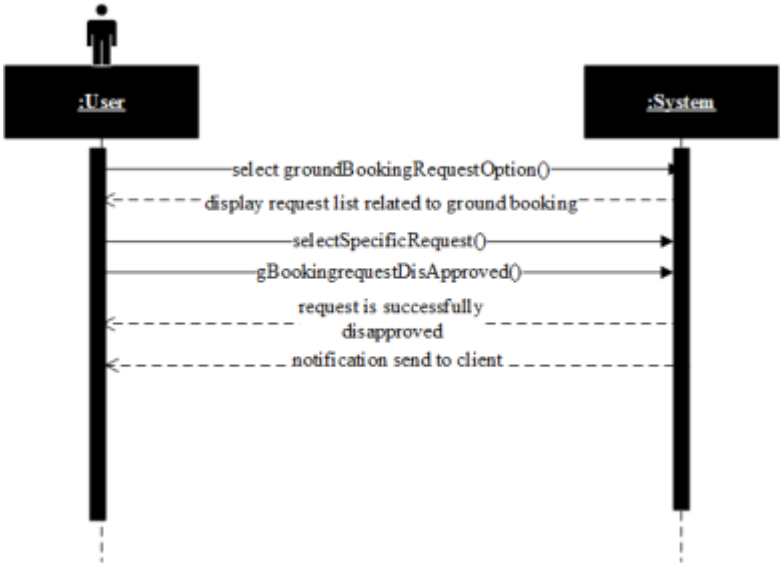

Figure 2.19: SDD for Reject Ground Booking Request

At the end of this chapter, we have discussed functional or non-functional requirements, system interfaces, and type of stakeholders, use cases and domain model. In the next chapter, we will discussed overall system its structure, components and system architecture design. Also, we will discussed system interfaces, screen images and web-pages.

“The really hard problem is discovering what are the right objects [classes] are in the first place.”

By: Carl Argila (1985-Todate)

Chapter 3

System Design

This chapter describes the design of system which includes the architectural diagram, some user interfaces for design. This chapter also includes class diagram which shows the interaction of different classes and structure of system with relationships and attributes.

3.1) Introduction

In this chapter, we briefly discussed all components of system. As we are using v-model software process model which involves modular approach of system. All these modules interacts with each other to fulfill the stakeholder’s requirements. Also, verification and validation of each module is done at the implementation and testing phase. System sequence diagrams provides the information whether a particular use case fulfill is used to meet the user requirement.

3.1.1) Design Overview

In design overview we need to view the system at architecture level by selecting a specific architecture. Architecture design represents the structure of data and program concepts that are required to build a computer-based system. In this chapter we will explain the structure and interaction of each components.

3.1.2) Requirements Traceability Matrix

Requirements Traceability Matrix (RTM) is used to captures all the requirements proposed by client or developer. It is used to track the requirements and to check the current project requirements are met. The main purpose of RTM is to see that all test cases are covered so that no functionality should miss while testing process. It shows the overall defects or execution status with focus set of requirements. If a test case fails, traceability helps determine the corresponding functionality easily. The RTM has six parameters [4].

1. Requirement id
2. Requirement name
3. Test Case
4. Sequence diagram
5. Class diagram
6. Interface (Design specification)

Requirement Id	Requirement Name	Test Case	System Sequence Diagram	Interface
UC:1	Login to profile	T01	Figure 2.2	Figure 3.2
UC:2	View upcoming match list		Figure 2.3	Figure 3.1
UC:3	Approval of membership request	T04	Figure 2.4	Figure 3.4
UC:4	Apply for membership		Figure 2.5	
UC:5	Send request for booking of ground		Figure 2.6	
UC:6	Approval of ground booking request	T04	Figure 2.7	Figure 3.5
UC:7	Using Duck-Worth Lewis Calculator		Figure 2.8	
UC:8	Selection of team		Figure 2.9	
UC:9	Send notification to end-users		Figure 2.10	
UC:10	Scheduling of matches	T02	Figure 2.11	
UC:11	Send feedback to club		Figure 2.12	
UC:12	View member profile		Figure 2.13	Figure 3.6
UC:13	Logging out		Figure 2.14	
UC:14	Search member profile		Figure 2.15	
UC:15	Edit member profile		Figure 2.16	
UC:16	Delete match schedule		Figure 2.17	

UC:17	Reject membership request		Figure 2.18	
UC:18	Reject ground booking request		Figure 2.19	
UC:19	Allocate match tickets			
UC:20	Add employee detail			
UC:21	Add member payment			
UC:22	Add employee payment			
UC:23	Add monthly expenditures			
UC24:	Book match ticket			

Table 3.1: Traceability Matrix

3.3) System Architectural Design

System architecture design is a design paradigm selected for the implementation of computer-based system. The software architecture of a program or computing system is the structures of system, which comprise software components, the externally visible properties of these components, and the relationship among them [5].

The architecture is not the operational software. Rather, it is a representation that enables us to analyze the effectiveness of the design is meeting its stated requirements, consider architectural alternatives at a stage when making design changes is still relatively easy and reduce the risks of the associated with the construction of the software. The architecture highlight early design decisions that will have a profound impact on all software engineering work that follows and, as important, on the ultimate success of the system as an operational entity. It defines how the system is structure and how its components work together. In our project, our system contain six to eight different modules which are named as sub-system which interact to each other for achievement of requirements. There are many different types of system architectures which are totally depend upon the type of system, its requirements or views of each stakeholder. As far as this project is concerned, our project is actually a web-based computerized system which allows all the functionality described in project scope. So we need to select specific software design pattern on which our system is going to build.

3.3.1) Chosen System Architecture

As we are using Core PHP which is commonly used without any frameworks and it has no architecture layers. Core PHP is basic programming which is used to create dynamic webpages. In the chapter#4 we have discussed why we are using Core PHP.

3.3.2) Discussion of Alternative Designs

The other system architecture are also used now-a-days for implementing web-based system. Every architecture has its own advantages and disadvantages. So, the selection of particular system architecture is the most important decision. One of alternative design paradigm is “Two Tier” Client/Server architecture in which business logic is divided into two physical locations. Usually, the clients contains most of the business logic and then Structure Query Language (SQL) allow business

logic to be stored and execute on the database server. This architecture is good if we have small business but the business is meant to grown then this architecture will not work properly. If the business rules are changed then application needs to rebuild and redeployed. With a 3-tier architecture, the business logic is separated from the presentation and database (data source). The security is also improved in 3-tier architecture because the database (Model) is separated from presentation layer (View) and these both layers are interacted through the middle-ware layer (Controller). Controller enables the interaction between presentation layer and database (data source). In our project as we are mentioned future work so that this architecture is helpful and our requirements will changed then this architecture will co-operate for enhancement.

3.4) User Design Interaction

The purpose of user interfaces design is to enable users to interact with your system by communicating meaning. If user can't figure out how our system works or where to go on your website then they will get confused. The user interfaces used in our system are clear, responsive, familiar, consistent, attractive and forgiving. Clarity is an important thing in user design interface everything on interface should clear and understandable. As our system run on computer and mobile devices it allow system to adjust its size and pages according to device screen size this characteristic is known as responsive. Responsive also means fast. Familiar means anything that is naturally understood like Facebook symbol is used instead of only Facebook text on website. Consistent interfaces allow users to develop usage patterns and users will learn how certain elements look like and recognize them in different context. Attractive means the color scheme of interface should visible and improves usability. Attractive in a sense that is makes use of that interface enjoyable. Forgiving interfaces means it should save users from common mistakes for example if someone delete data then system should show a message of confirmation.

3.4.1) Description of the User Interface

The user interfaces play a vital role during phase of design that how the user will attract with system. Few user interfaces are sketched which provides help to user for communication with system. These interfaces are not fully fixed interfaces they can vary during the phase of implementation. They are roughly designed by using three following tools.

1. Evolus Pencil (Version 2.0.5)
2. iPLOTZ (Version: 4.1.2)
3. Adobe Photoshop 7.0

The first user interface is homepage in which details, history and other relevant information about the Diamond Club are visible. Few photos of club are slide on first page. On this page user can easily go to other pages and this is main page of club. Second user interfaces describes how member of club will logged in to his profile and on the other side all other members with name are visible. The third user interface next step of second interface when the member will successfully logged to profile then member data will appear. Fourth user interface contains requests of those user who has sent request to Diamond Club for membership. Only administrator have rights to accept or reject his request and at this page the administrator first need to be logged in. Fifth interface contains ground booking

request and only administrator can process these request according to club rules of defined criteria. Last interface shows a graph of each player with respect to his performance in a match.

3.4.1.1) Screen Images

The user interfaces are designs of system and its components which interact with each other to fulfill requirements of users and provide assessment of better interaction with system. In this project, there are six to eight system components and some of the prototypes or roughly sketched interfaces are shown.

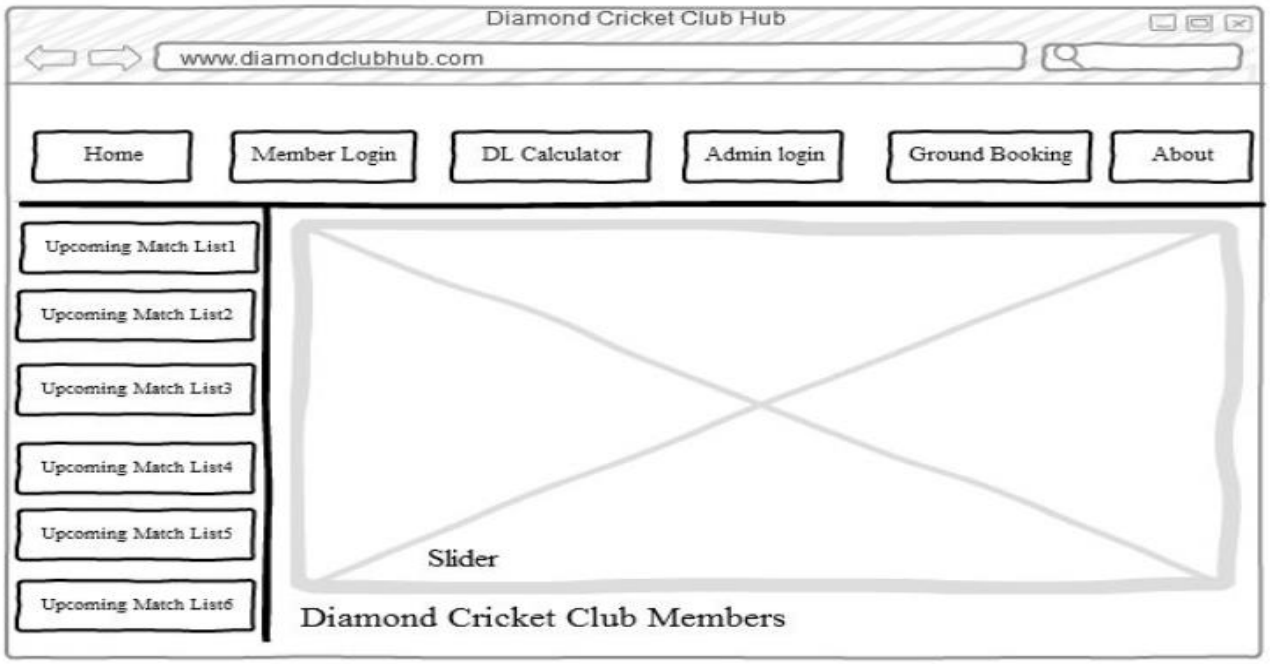

Figure 3.1: Interface for Homepage

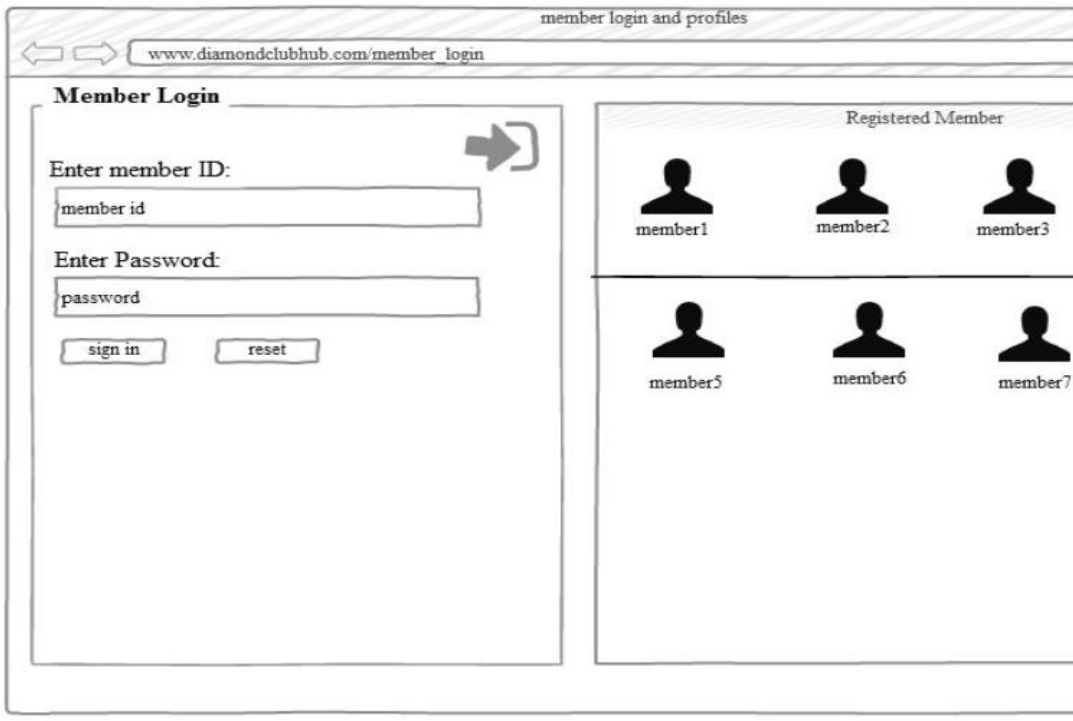

Figure 3.2: Interface for Membership Login

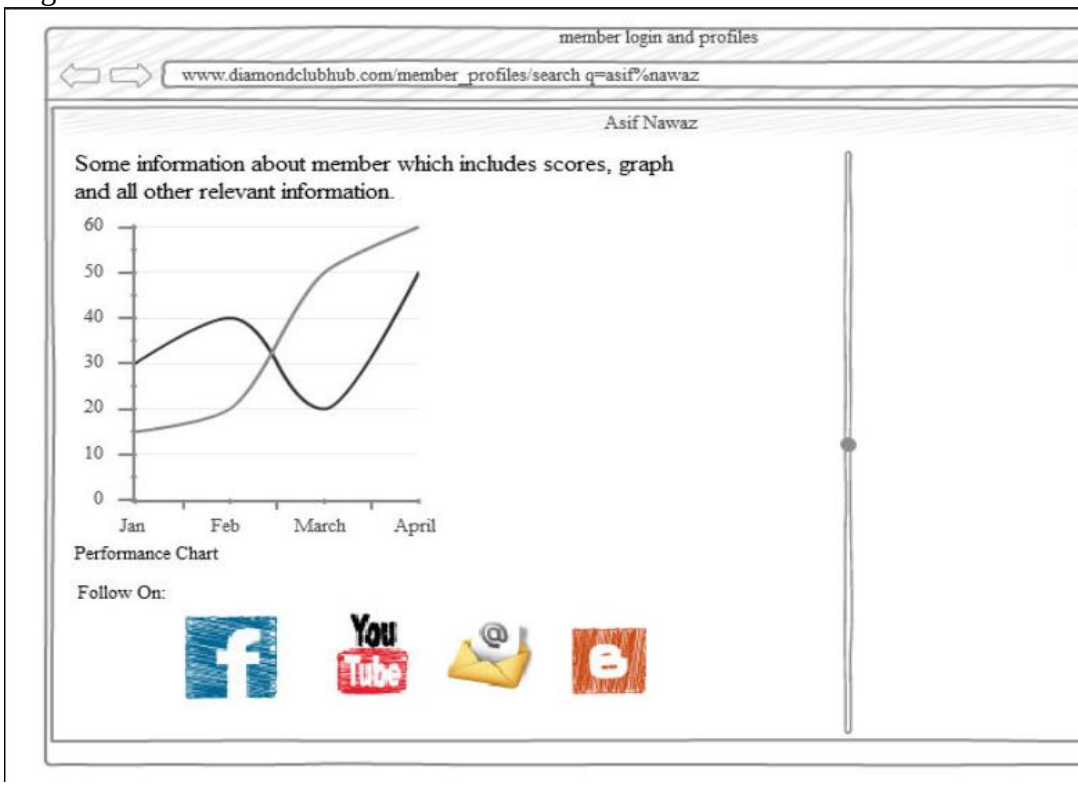

Figure 3.3: Interface for particular member information

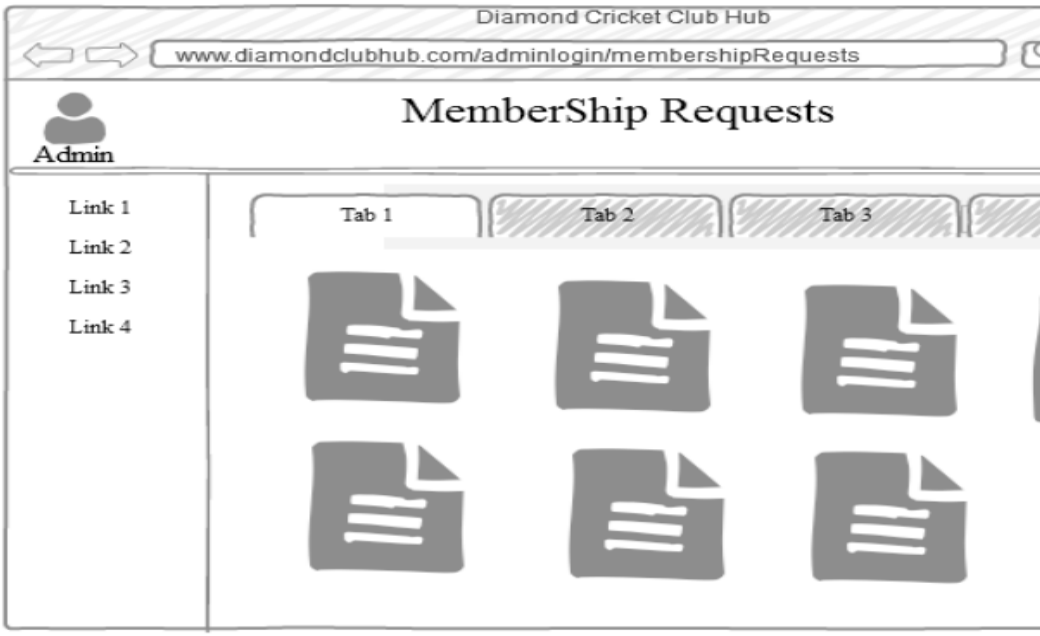

Figure 3.4: Interface for membership request

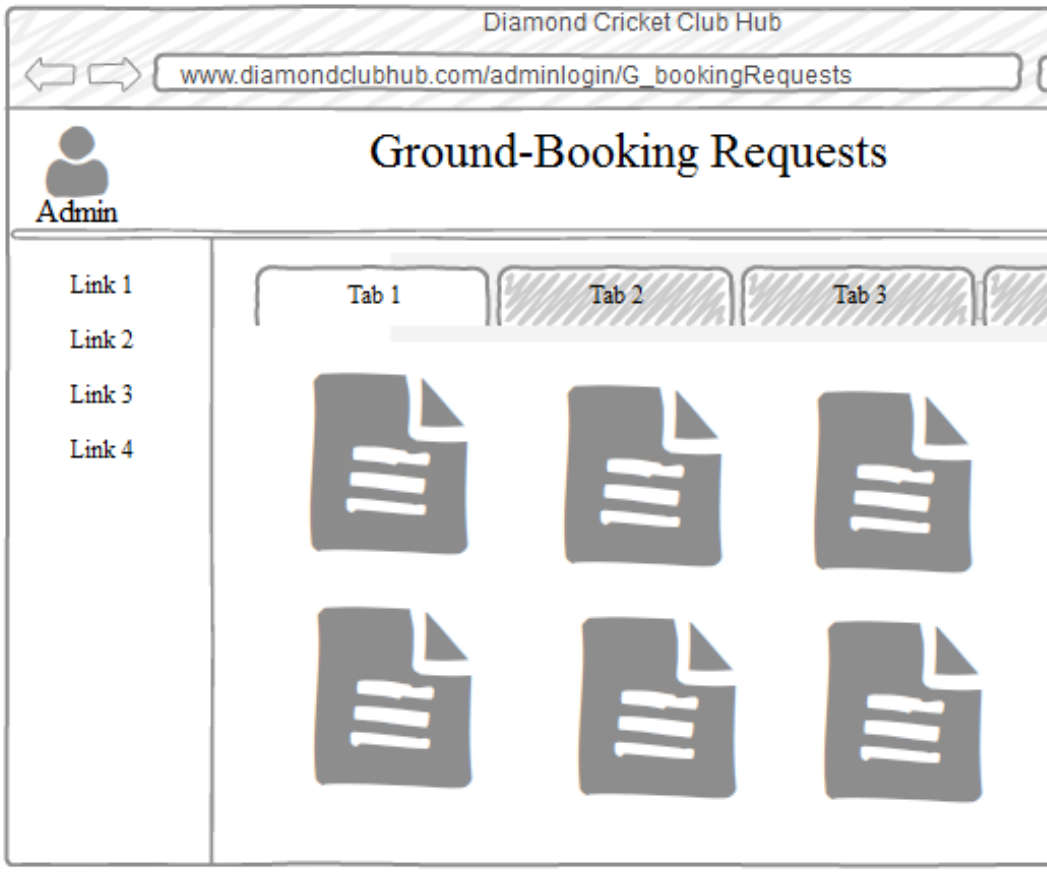

Figure 3.5: Interface for ground booking request

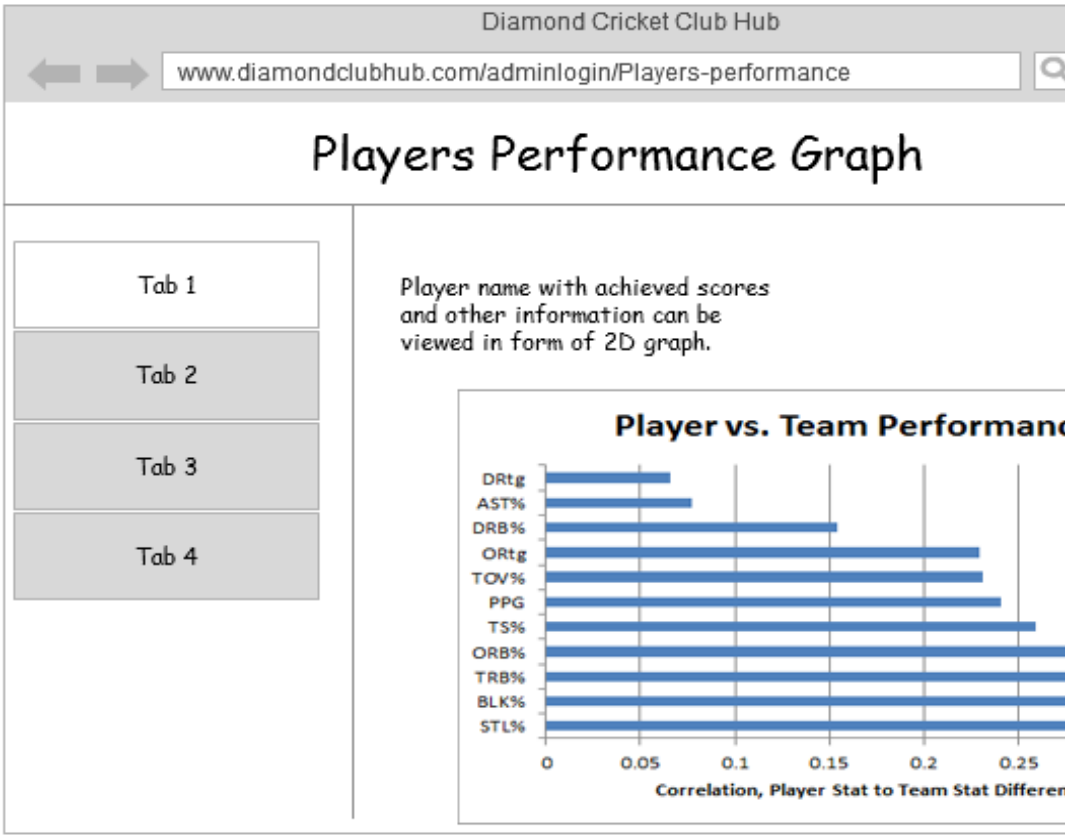

Figure 3.6: Interface for viewing player performance

3.4.1.2) Objects and Actions

Sequence diagram (SD) in Unified Modeling Language (UML) indicates how events cause transitions from object to object. In short, the sequence diagram is short-hand version of the use case. It represents flow from one object to another as a function of time [6]. Our chosen system architecture is based on 3-tier. So, there are three objects view, controller and model on user perform action or to access the system on the base on their requirements. The following SD's are covered few operation performed by administrator, member or end-user. Member and administrator have their profile but end-user have no id or password.

3.5) Detailed Description of Components

Our system is divided in modules in which each module perform set of actions by end-users, members and administrator of club. Diamond Cricket Club have six to eight modules which are integrated and then merged for achievement of stakeholder's requirements. The components are given below:

1. Membership Management System (MMS)
2. Profile Management System (PMS)

3. Match Scheduler System (MSS)
4. Notification Management System (NMS)
5. Feedback Management System (FMS)
6. Ground Booking Management System (GBMS)
7. Employee Payment Management System (EMMS)
8. Monthly Club Expenditures Management System (MCEMS)

The MMS component manages membership requests which from end-users whose are interested to join to club. This whole components only used by end-users and we have already defined in scope of project that "Diamond Cricket Club" accepts membership requests whose only belongs to Islamabad and Rawalpindi. This component save requestor data such as email and other relevant details. Data comes from interface and input validation checks at client side. Interface helps users to avoid from wrong input value entry and store this data to database (Model). On the other side the administrator checks requests and he can accept particular or not depend upon the club rules. If he accepts or reject any request then a notification will send to that requestor through email.

Second component PMS allow administrator and members to login or log out to the system. Only the registered member can access their profiles. It maintains the records of each member like name, age, height, photo, obtained scores, total number of winning awards and other relevant details. End-user can't access it because he was not registered to club. When a user become a member of Diamond Club then admin will send him his id and password through which he will able to access his profile. Member profile content changed by member and administrator both have their rights to change the contents according to rule of club.

Third component MMS allow administrator to schedule the matches with particular team by fixing particular ground name, time, date, rules and other relevant information so that end-user don't face difficulty to find the ground. Administrator can delete, add, update, search match schedule. Each match schedule have a unique id through which later he can delete or update the contents of match according to circumstances. So that users can easily view the information about the fixed matches. This component depend upon PMS component because the contents of match would not change if administrator was not logged in to system. Users and members can search or view the schedule match.

Fourth component NMS allow administrator to send notification to end-users if particular end-user request is accepted or rejected by the administrator then this component automatically access requestor email id and then send him notification to email address entered by user from interface. This component manages both type of requests i.e. ground booking requests or membership requests. As we are also add another functionality to this component for sending notification to all the registered members for the payment of club fee per month. It automatically maintains email address of all registered members. Which is very efficient way to inform the members.

Fifth component ALMS allow administrator to keep track of records of awards. If the club team won a match then administrator will able to add the details of that match through this component. This component can used by end-users and member according to their rights i.e. members or end-users can view or search the particular award list details. Award list contains name of team, members, date, time and all other information. So that if any end-user visit the website of club then

he can easily view the all award lists details. Each list has a unique id through which later administrator can change or delete the details of match.

Sixth component FMS allow users or members to send feedback to club so that club authority can improve performance of club. It will maintains data of each feedback sender and also keep their comments so that administrator can view these comments with specific date and time. Data comes from user through interface and then saved to database. Input validation occurred at client side the responsive or forgiving characteristics of interfaces allow user to avoid from mistakes. Last or seventh component of our system is GBMS which allow end-users and member to book the Diamond Ground. The ground booking is only performed by those users whose are citizen of ISB and RWP. On the clients side, the use first check the availability of ground with timing and date and then he will send request to club. Only administrator can accept or reject this request according to circumstance or weather condition. If he accepts or reject that request then NMS system send notification to user or member through email.

Seven component EPMS allows administrator to keep track of club employees. He can add an employee by entering his personal details like name, phone number etc. He can add, remove or update the employee details with the passage of time. Also, at every month he can add the payment details of employees by entering date of payment and his salary. We can say that this components deals with payment record of employee. If mistakenly, administrator tries to add employee payment record two times then system will prompt and show message that the payment of specific employee is already added.

Eighth component MCEMS allows administrator to keep track of all monthly expenditures which have been used in the maintenance of club and other types of expenditures like cricket balls, bats. At the end of month administrator can add, remove or update the total expenditures of club.

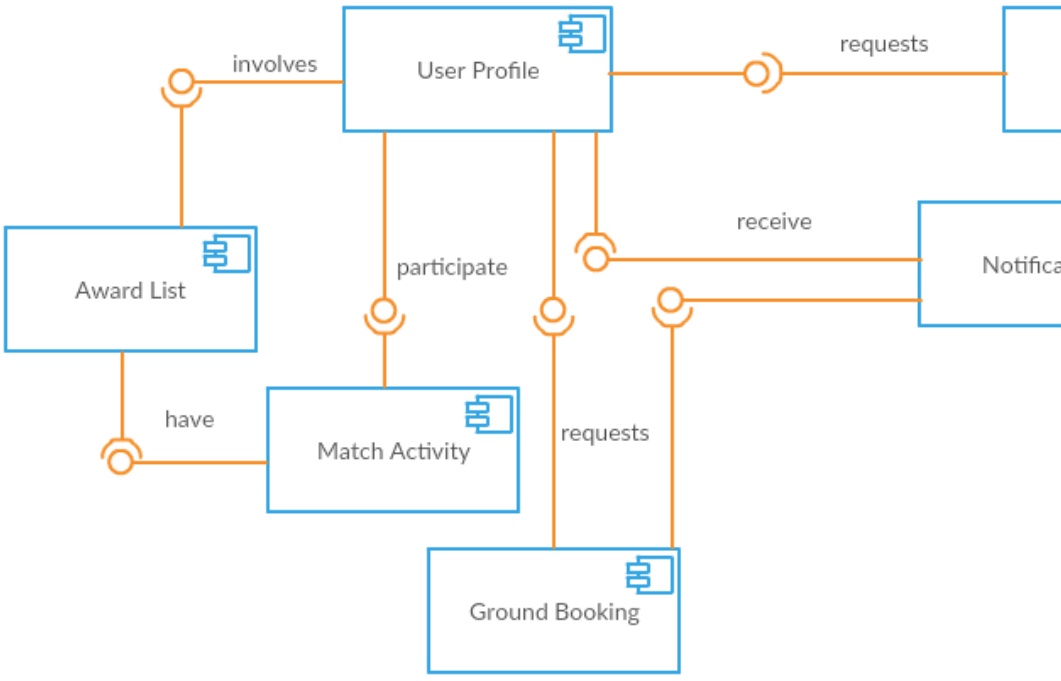

Figure 3.7: Component Diagram

In the end of discussion, our purpose of choosing a modular approach is to divide the whole system into major pieces and then achieve the requirements from these pieces. Different components are dependent to each other some are independent. This approach reduces the chances of errors if any error occurs then recovery easy. As we described in chapter-1, we are using V-software process model which follow this approach because verification and validation of each module is compulsory at this model. So, this is a good way to split the whole project and then merge them.

3.6) Entity Relationship Diagram

As we are moving towards to implementation of our system according to the requirements and scope of this system. For implementation, we need to design or construct a model which is used to managing, retrieving, storing and access the information about club activities, teams, members, ground and membership requests, payments records of members and employees. For this process, we designed an ERD which contains different tables which will be used to storing the information. Each table have different columns and primary key. Few tables have foreign key for retrieving information from two or more tables.

1. Membership
2. Ground Booking
3. Transactions
4. Visitors
5. Team Players
6. Teams
7. Match Tickets
8. Ticket Transactions
9. Contact
10. Match schedule
11. Other expenditures

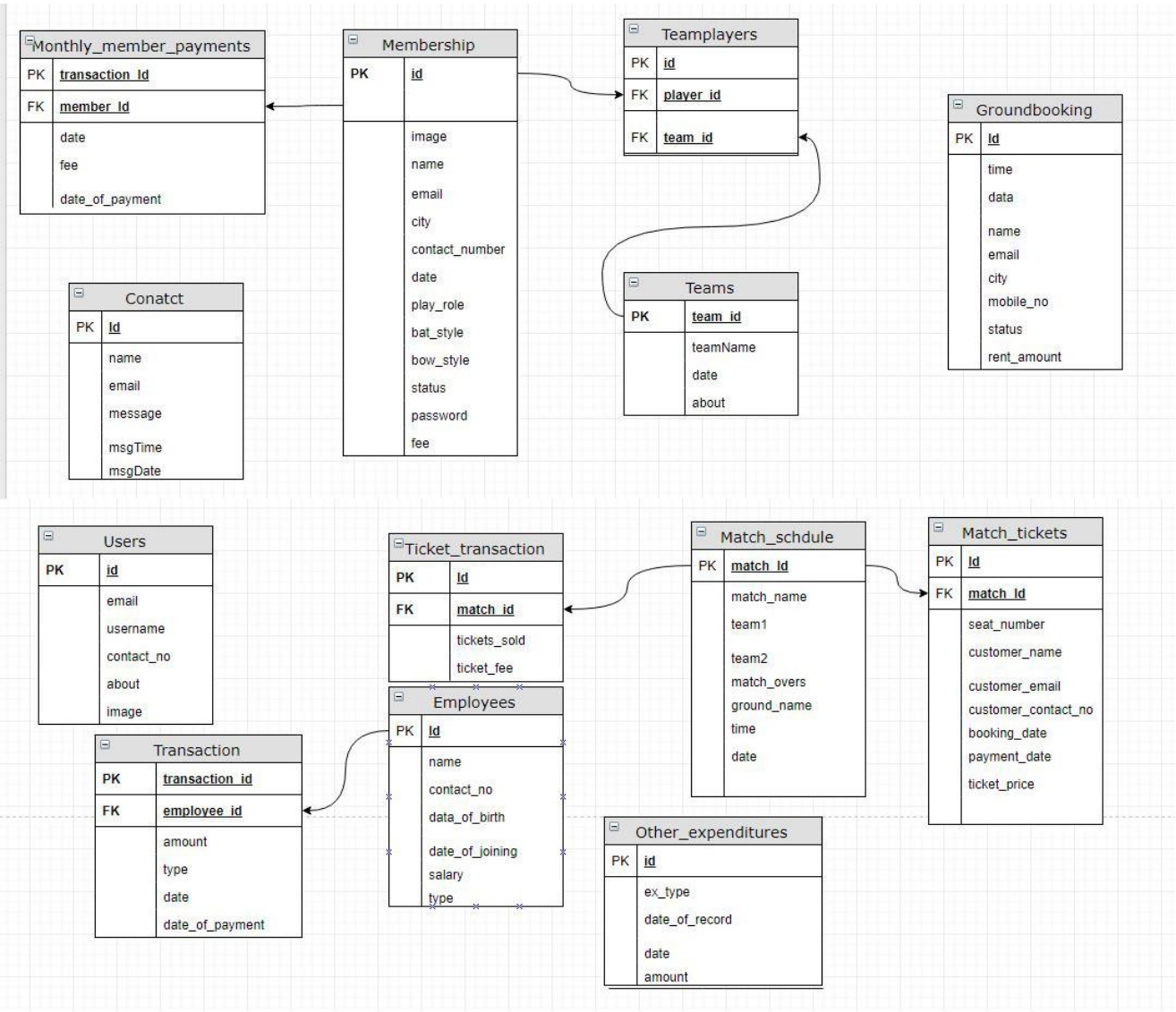

Figure 3.8: Entity Relationship Diagram

In the end of this chapter, as this chapter contains many diagrams of interfaces, SSD's. In short, this chapter was lengthy than other because in this chapter we were talking about the design of interfaces, system design paradigm, system architecture design and then components of systems and their dependencies or functionality. In the next chapter, we will discuss implementation details of our system which allow users to fulfill their requirements mentioned in scope.

"Goals should be broken down into parts, and then think over the plan of their implementation."
Sunday Adelaja [1967-Todate]

Chapter 4

Implementation

In this chapter, we will discuss about the details related to implementation of our system. Previous chapter was describing design paradigm like interfaces, sequence and system sequence diagrams. This chapters includes selection of language, deployment tools which are being used during the development of system, selection of programing languages and Database server.

4.1) System Definition

System is composed of two components first one is web pages which are includes functions and SQL queries to fulfill system functionality on distributed network. Second part if Windows server which includes Data bases. The written script in Hypertext Preprocessor or Personal Home page (PHP) which is embedded with HTML5, other languages and various web content management systems. The PHP interact with Windows Servers like WAMP, XAMP which are locally runnable and executes scripts without hosting a web site on internet. The user interfaces (UI) are used to provide interaction between the user and local server. As our system is runnable on both computer and mobile devices.

4.1.1) Windows Servers

Windows servers are usually freely available on the internet. There are many windows servers which actually designed to allow to create web applications, manages databases and server. They also provides server settings and specific rights on Database. We use WAMP server which is lightweight, easy to use and stable. WAMP makes easy to code PHP and creating Data bases (in MYSQL) in windows platforms and it is available for both 64 bit and 32 bit system.

4.2) Development Tools

4.2.1) Framework

As we are using Core PHP and Core PHP has no framework. Core PHP is very basic PHP. It is used to create dynamic web pages and works without any extra library. As a student, it is quite time consuming to directly jump to a PHP framework so I prefer to learn core or principle PHP

programming to create web applications. After knowing basic concepts of PHP then I'll jump to any framework. PHP can be used on all main operating systems including Linux, Mac OS X.

4.2.2) Language Selection

As I have already mentioned that we are using basic or core PHP which is commonly used to create dynamic pages. So, in for development of our system we use PHP with few others languages which are mostly used in web development. The following languages are used during the development of system.

1. Hypertext Markup Language (HTML5)
2. Bootstrap 3 which is used as a front-end framework
3. Asynchronous JavaScript And XML (AJAX)
4. Java Script (JS)
5. Using JQuery with AJAX

The PHP is used within HTML, AJAX and Bootstrap which is very responsive for mobile applications also. PHP is open source which is developed and maintained by a large group of developers, which helps to make support community. It is relative fast and easy to use because its syntax is similar to C language. We can say that it is easy for them to pick up who are familiar with C. As it is maintained by many developers which means when bugs are founds, it can quickly fixed. Since, many websites are data/content driven, so we will use database frequently. The weakness of PHP is security because anyone can view code for security we can use cryptography techniques like encryption for ensure security. AJAX is very fast we can get response from page without refreshing it.

4.3) Other Software's and Tools

Some other application software's also used during the deployment of system which are given below:

1. Notepad++ (Version: v7.4.2)
2. Sublime Text3 (Build-3126)
3. Google Chrome Browser (Version: 59.0.3071.115 (Official Build 64bit))
4. Filezilla (Version: 3.25.1)

Filezilla is used for hosting the whole system online over the internet. Any user can access our system from anywhere according to its right. A visit counter also included to system by which we can check the unique visitors on the base of IP addresses.

4.4) Coding

As our system is designed for club administrator and free end-users. Administrator will manages activities of club online like adding teams by choosing members, accepting or rejecting of ground booking and membership requests. He can also maintains player profiles review the user comments.

Coding for administrator contains one folder which contains important assets which are linked with other files. Similarly, there is other folder for client which contains client related files and scripts. Only one database will use for the both client and administrator. Which contains 8 to 10 tables which are linked with each other by foreign keys.

4.4.1) Code Structure

The code use in the development of system is based on PDO PHP which is more secure than simple PHP. Few modules are built with core PHP with AJAX and JQuery scripts. JavaScript and other CSS files are included in one file named as master.php, header.php and footer.php. Which are being used in every page. For Database connection we use two database configuration files i.e. db.php and _xonection.php which also use in every page. The one system is for clients and members so that they can perform different queries defined within scope. Another system is built for only administrator who can perform different operation specified in scope.

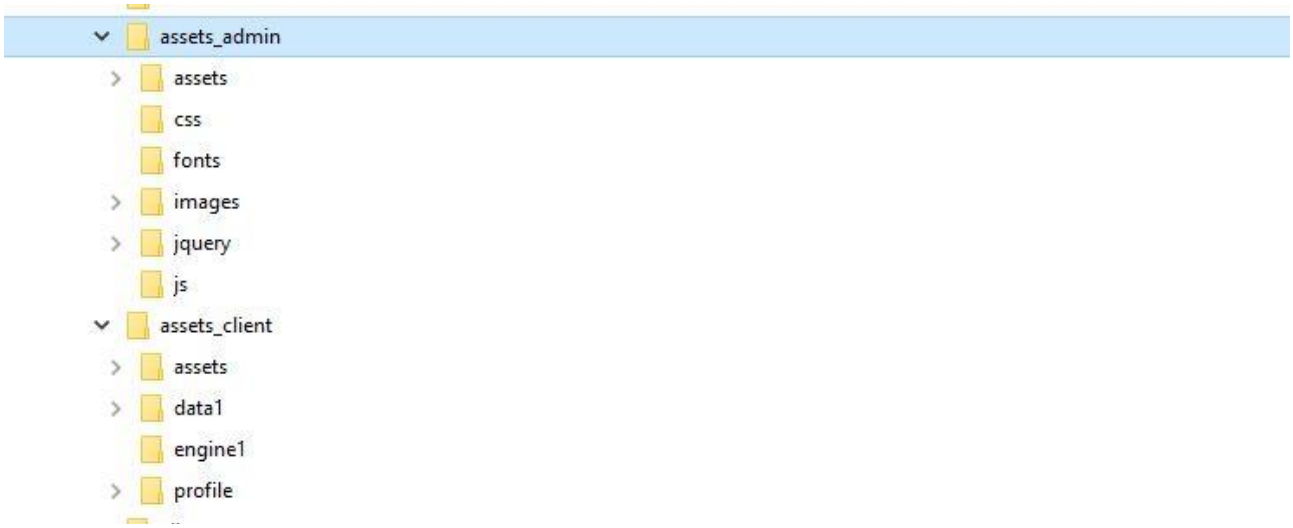

Figure 4.0-1: Assets Hierarchy

This hierarchy is being used in our system. The admin related assets files are enclosed within assets_admin folder. And client related assets are placed within assets_client folder. The code is well-structured.

4.1.2) Code Debugging Tools

As we are using AJAX, JQuery and JavaScript in the development of system. One page contains multiple scripts of all these languages. In our system, we have use a web browser which also used as a debugger during the implementation. In chrome we can see the request of AJAX and its returned response. Ajax send and receive requests at a time in form of JSON.

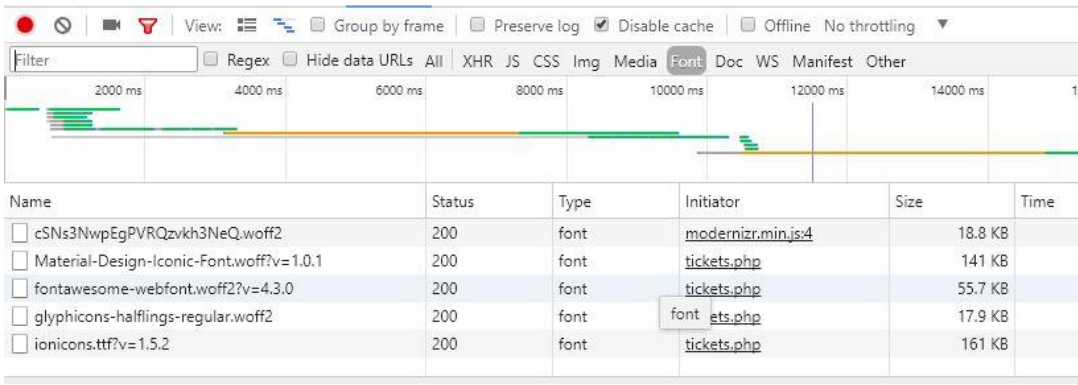

Figure 4.0-1: Page Response on chrome

This screen shot describes which .js file is being used during load a page. By this we can easily see error in .js file with line number. In code, it is hard to catch the error by using this technique we can catch or fix specific error related to AJAX, JQuery and JavaScript.

4.2) Duck Worth Lewis Algorithm

Duck-Worth Lewis is a mathematical formulation for calculating the target score of second batting team if match is interrupted by weather or other circumstances. This method is generally accepted for calculating the target score. Duckworth-Lewis method is an attempt to set a statistically fair target for the second team's innings, based on score achieved by first team, taking their wickets lost and overs into account. If two teams said A, B are playing a match, after the innings of first team, then second team played few overs and loss wickets, in case if any emergency occurs during the match then first we need to enter the team A runs and number of loss wickets. And Team B played overs and their loss wickets. At the end we have to enter total overs of match. The table for the 50 over match is given

OVERS LEFT	WICKETS LOST										OVERS LEFT
	0	1	2	3	4	5	6	7	8	9	
50	100.0	93.4	85.1	74.9	62.7	49.0	34.9	22.0	11.9	4.7	50
49	99.1	92.6	84.5	74.4	62.5	48.9	34.9	22.0	11.9	4.7	49
48	98.1	91.7	83.8	74.0	62.2	48.8	34.9	22.0	11.9	4.7	48
47	97.1	90.9	83.2	73.5	61.9	48.6	34.9	22.0	11.9	4.7	47
46	96.1	90.0	82.5	73.0	61.6	48.5	34.8	22.0	11.9	4.7	46
45	95.0	89.1	81.8	72.5	61.3	48.4	34.8	22.0	11.9	4.7	45
44	93.9	88.2	81.0	72.0	61.0	48.3	34.8	22.0	11.9	4.7	44
43	92.8	87.3	80.3	71.4	60.7	48.1	34.7	22.0	11.9	4.7	43
42	91.7	86.3	79.5	70.9	60.3	47.9	34.7	22.0	11.9	4.7	42
41	90.5	85.3	78.7	70.3	59.9	47.8	34.6	22.0	11.9	4.7	41
40	89.3	84.2	77.8	69.6	59.5	47.6	34.6	22.0	11.9	4.7	40
39	88.0	83.1	76.9	69.0	59.1	47.4	34.5	22.0	11.9	4.7	39
38	86.7	82.0	76.0	68.3	58.7	47.1	34.5	21.9	11.9	4.7	38
37	85.4	80.9	75.0	67.6	58.2	46.9	34.4	21.9	11.9	4.7	37
36	84.1	79.7	74.1	66.8	57.7	46.6	34.3	21.9	11.9	4.7	36
35	82.7	78.5	73.0	66.0	57.2	46.4	34.2	21.9	11.9	4.7	35
34	81.3	77.2	72.0	65.2	56.6	46.1	34.1	21.9	11.9	4.7	34
33	79.8	75.9	70.9	64.4	56.0	45.8	34.0	21.9	11.9	4.7	33
32	78.3	74.6	69.7	63.5	55.4	45.4	33.9	21.9	11.9	4.7	32
31	76.7	73.2	68.6	62.5	54.8	45.1	33.7	21.9	11.9	4.7	31
30	75.1	71.8	67.3	61.6	54.1	44.7	33.6	21.8	11.9	4.7	30
29	73.5	70.3	66.1	60.5	53.4	44.2	33.4	21.8	11.9	4.7	29
28	71.8	68.8	64.8	59.5	52.6	43.8	33.2	21.8	11.9	4.7	28
27	70.1	67.2	63.4	58.4	51.8	43.3	33.0	21.7	11.9	4.7	27
26	68.3	65.6	62.0	57.2	50.9	42.8	32.8	21.7	11.9	4.7	26
25	66.5	63.9	60.5	56.0	50.0	42.2	32.6	21.6	11.9	4.7	25
24	64.6	62.2	59.0	54.7	49.0	41.6	32.3	21.6	11.9	4.7	24
23	62.7	60.4	57.4	53.4	48.0	40.9	32.0	21.5	11.9	4.7	23
22	60.7	58.6	55.8	52.0	47.0	40.2	31.6	21.4	11.9	4.7	22
21	58.7	56.7	54.1	50.6	45.8	39.4	31.2	21.3	11.9	4.7	21
20	56.6	54.8	52.4	49.1	44.6	38.6	30.8	21.2	11.9	4.7	20
19	54.4	52.8	50.5	47.5	43.4	37.7	30.3	21.1	11.9	4.7	19
18	52.2	50.7	48.6	45.9	42.0	36.8	29.8	20.9	11.9	4.7	18
17	49.9	48.5	46.7	44.1	40.6	35.8	29.2	20.7	11.9	4.7	17
16	47.6	46.3	44.7	42.3	39.1	34.7	28.5	20.5	11.8	4.7	16
15	45.2	44.1	42.6	40.5	37.6	33.5	27.8	20.2	11.8	4.7	15
14	42.7	41.7	40.4	38.5	35.9	32.2	27.0	19.9	11.8	4.7	14
13	40.2	39.3	38.1	36.5	34.2	30.8	26.1	19.5	11.7	4.7	13
12	37.6	36.8	35.8	34.3	32.3	29.4	25.1	19.0	11.6	4.7	12
11	34.9	34.2	33.4	32.1	30.4	27.8	24.0	18.5	11.5	4.7	11
10	32.1	31.6	30.8	29.8	28.3	26.1	22.8	17.9	11.4	4.7	10
9	29.3	28.9	28.2	27.4	26.1	24.2	21.4	17.1	11.2	4.7	9
8	26.4	26.0	25.5	24.8	23.8	22.3	19.9	16.2	10.9	4.7	8
7	23.4	23.1	22.7	22.2	21.4	20.1	18.2	15.2	10.5	4.7	7
6	20.3	20.1	19.8	19.4	18.8	17.8	16.4	13.9	10.1	4.6	6
5	17.2	17.0	16.8	16.5	16.1	15.4	14.3	12.5	9.4	4.6	5
4	13.9	13.8	13.7	13.5	13.2	12.7	12.0	10.7	8.4	4.5	4
3	10.6	10.5	10.4	10.3	10.2	9.9	9.5	8.7	7.2	4.2	3
2	7.2	7.1	7.1	7.0	7.0	6.8	6.6	6.2	5.5	3.7	2
1	3.6	3.6	3.6	3.6	3.6	3.5	3.5	3.4	3.2	2.5	1
0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.3: Table for DL calculator

4.3) Code Captions

```php
<?php

        include("../common/db.php");

        $ip = $_SERVER['REMOTE_ADDR'];

        $sql = "SELECT `ip` FROM `visitors` WHERE `ip` ='$ip'";

        $res = mysqli_query($conn,$sql);

        if(mysqli_num_rows($res)==0){
                $query = "INSERT INTO `visitors`(`id`,`ip`) VALUES(NULL,'$ip')";

                $response = mysqli_query($conn,$query);

        }

        $count ="SELECT `ip` FROM `visitors`";

        $num = mysqli_query($conn,$count);

        echo mysqli_num_rows($num);

?>
```

```php
<?php
 include("../common_client/master.php");
 $err="";
if(isset($_POST["submit"])){

     $email = trim($_POST["email"]);
     $email = strip_tags($email);
     $email = htmlspecialchars($email);

     $password = trim($_POST["password"]);
     $password = strip_tags($password);
     $password = htmlspecialchars($password);

     $query = "SELECT *FROM `membership` WHERE email='$email' AND password='$password'";
     $result= mysqli_query($conn,$query);

     $row = mysqli_fetch_array($result,MYSQLI_ASSOC);

     #$active = $row['active'];

       $count = mysqli_num_rows($result);

       // If result matched $myusername and $mypassword, table row must be 1 row

       if($count == 1) {
         # session_register("myusername");
          #$_SESSION['login_user'] = $myusername;
        # echo "succesfully login";
        session_start();
        $_SESSION['member_email']= $email;
         header("location: profile.php");
       }else {
          #$error =
          #echo "Incorrect Login Details";
         $err = "Incorrect Email and Password Combinations!";
       }
   }
```

```
<div class="col-md-6" >
      <form  method="POST" name="login">

                <h3><strong>Members</strong>  Login</h3>

                        <small class="text-muted">
                            <br />
                            Please enter email ID and Password to procced.
                        </small>

                    <div class="form-group">

                        <label>Member ID:</label>

                        <input type="text" class="form-control" name="email" id="email" placeholder="Enter email" required>

                    </div>

                    <div class="form-group">

                         <label for="pwd">Password:</label>

                         <input type="password" class="form-control" id="pwd" name="password" placeholder="Enter password" required>

                    </div>

                    <div>

                       <span style="color:red;" ><?php echo $err; ?></span>

                    </div>

                    <button type="submit" class="btn btn-default" name="submit" id="sub">Submit</button>
                    <button type="reset" class="btn btn-default" id="res">Reset</button>
      </form>
                    <a href="membershipform.php">New User Registration?</a>
</div>
```

```
<script>
   $(document).ready(function(){

                load_data();

                function load_data(query)
                {
                        $.ajax({

                         url:"teams_actions/fetch.php",
                         method:"POST",
                         data:{query:query
                         },

                         success:function(data)
                         {
                          $('#result').html(data);
                         }
                        });
                }

           $('#search_text').keyup(function(){
                        var search = $(this).val();
                        var id = $("#id").val();
                        if(search != '')
                        {
                         load_data(search+id);
                        }
                        else
                        {
                         load_data();
                        }
                       });
   });

   </script>
```

In the next chapter we have discussed about the different type of testing techniques to validate our system whether it is fulfill our requirements or not. We have discussed test cases to validate the each use case.

"Discovering the unexpected is more important than confirming the known."

By: George E. P. Box [1919-2013]

Chapter 5

Software Test

In this chapter we will discussed our system and its components. As this chapter is all about testing the software product whether it fulfill the user requirements or not. Later we will select a testing approach of testing through which we will test each requirement on our system. At the end of this chapter we will show results of testing.

5.1) Test Approach

As our system is not implemented yet. So, we need to choose an approach by which testing can be carried out without product coding or implementation. There are two test approaches first is reactive in which test is not started until coding and design has not completed. On the other hand, second approach is called proactive in which test design process is initiated as early as possible in order to find and fix the defects before the build is created. We are choosing reactive approach. System testing verifies that all elements mesh properly and that overall system function performance is achieved [7].

5.2) Testing Tools and Environment

There are many types of testing depending upon the type of system. As our system is web-based and it is composed on the different modules. In our case, we are choosing black-box testing because which is also known as behavioral or functional testing. Black-box testing enable us to derive set of input conditions that will fully exercise all functional requirements for a program. This testing attempts to find following types of errors:

1. Interface errors.
2. Incorrect or missing function.
3. External database access or error in data structures.
4. Behavior or performance errors.
5. Initialization and termination errors.

In black-box testing the internal functionality or coding is hidden from the external world. Its only concern with the input and output while in the white-box or structural testing internal functionality and code is visible to tester. Our system is web-based so black-box testing is good for handling errors related with interface or other database access errors. By apply black-box testing we obtain test cases that provides us details about the presence or absence of classes or errors.

5.3) Test Cases

The test cases of few important requirements are given with a success or alternative flow in form of tables. We have not implemented this system yet so after implementation we will change the end results of our output status. This table contains test case id, description of test case, tester, instructions, setup, expected result and status. Status may be fail or pass value corresponding to system.

5.3.1) Login to Profile

ID	T01
Description	Member will logged in to system.
Tester	Member
Setup	Registered member enter his email "rehmana578@gmail.com" and password 123.
Instructions:	1. Member will select login button. 2. He will enter member id and password. 3. Press sign in button.
Expected result	Member will logged in to system with his email.
Status	**Pass**

Table 4.1: Test Case for login to profile

4.3.2) Login to Profile (Alternative Scenario)

ID	T01
Description	Member will logged in to system.
Tester	Member
Setup	Registered user registered with id "rehmana578@gmail.com" and password 1234.
Instructions:	1. Member will select login button. 2. He will enter member id and password. 3. Press sign in button.

Expected result	Member will not logged in to system and a message will appear “Wrong email and password combinations”.
Status	**Pass**

Table 4.2: Test Case for login to profile

5.3.3) Schedule of Match

ID	T02
Description	Administrator will schedule the match.
Tester	Administrator
Setup	Administrator enter his email “rehmana578@gmail.com” and password 123.
Instructions:	1. Administrator will select log button. 2. He will enter member id and password. 3. Press sign in button. 4. Then he will enter the details of for scheduling with particular date, time, ground and other relevant details. 5. Match will scheduled successfully. 6. Press logout button.
Expected result	Match will successfully schedule by administrator.
Status	**Pass**

Table 4.3: Test Case for schedule of match

5.3.4) Schedule of Match (Alternative Scenario)

ID	T02
Description	Administrator will schedule the match.
Tester	Administrator
Setup	Administrator enter his email “rehmana578@gmai.com” and password 123.
Instructions:	1. He will enter member id and password. 2. Press sign in button. 3. Then he will enter the details of for scheduling with particular date, time, ground and other relevant details. 4. He choose team1 and team2 which have same name. 5. After clicking on save button an error will appear “team1 and team2 must be different”. 6. Press logout button.
Expected result	Match will not schedule by administrator.

Status	**Pass**

Table 4.4: Test Case for schedule of match

5.3.5) Add a Team

ID	T03
Description	Administrator will add team by selecting 11 players.
Tester	Administrator
Setup	Administrator enter id “rehmana578@gmail.com” and password 123.
Instructions:	1. He will press add team link under the teams. 2. Then he will write name of team 3. He will click on add button. 4. He will select or choose 11 players for team. 5. The team will added. 6. Press logout button.
Expected result	A team is added successfully by administrator
Status	**Pass**

Table 4.5: Test Case for add a team

5.3.6) Add a Team (Alternative Scenario)

ID	T03
Description	Administrator will add team by selecting 11 players.
Tester	Administrator
Setup	Administrator enter id “rehmana578@gmail.com” and password 123.
Instructions:	1. Administrator will select logged in with his email “rehmana578@gmail.com”. 2. He will click on add team link. 3. He will enter the name of team. 4. He will not click on add button 5. Press logout button.
Expected result	Team will not added by administrator.
Status	**Pass**

Table 4.6: Test Case for add a team

5.3.7) Approval of Ground booking Request

ID	T04
Description	Administrator will approve membership request
Tester	Administrator
Setup	Administrator enter id “rehmana578@gmail.com” and password 123. He will accept the ground booking request.
Instructions:	1. Administrator will select logged in with his email and password. 2. He will press ground booking requests option. 3. He will select a particular request. 4. After reading data or requestor he will click on the accept button. 5. Email sends to that user. 6. Press logout button.
Expected result	Ground booking request will accepted by administrator.
Status	**Pass**

Table 4.7: Test Case of approval of ground booking request

5.3.8) Approval of Ground booking Request (Alternative Scenario)

ID	T04
Description	Administrator will approve ground booking request
Tester	Administrator
Setup	Administrator enter his email “”rehman578@gmail.com and password 123. He will accept the ground booking request.
Instructions:	1. Administrator will select logged in with id ad_122. 2. He will enter member id and password. 3. Press sign in button. 4. He will press ground booking requests option. 5. He will select a particular request. 6. After reading data or requestor he will not click on accept button. 7. Press logout button.
Expected result	Ground booking request will not accepted by administrator.
Status	**Pass**

Table 4.8: Test Case of approval of ground booking request

These are the major test cases because without this test our system will not run i.e. without members there is no role of club.

In this chapter, we have discussed software testing approach and features of testing. As testing is used for validating the expected and observed behavior. By this purpose, as we are using modular approach so we have to test each module. In the next chapter, we will discuss future enhancement of this system that which features can be added to this system.

"Software is a great combination between artistry and engineering."

By: Bill Gates [1955-Todate]

Chapter 6

Conclusion and Future Work

In this chapter we will discuss about the future work which can be added in our system and we draw conclusion of this system.

6.1) Conclusion

As our system has been implemented according to the user requirements and its scope. We can conclude that this system is very useful any cricket club but now it is only established for Diamond Cricket Club. As Diamond Cricket Club don't have this system user's face problem to get information of membership form, ground booking and upcoming matches. So, this system is initially build for those users or stakeholders who are interested in sports usually with cricket. The Facebook group of this club is available but people need such a platform in which they can view club activities, registered members profiles and details. So, for fulfilling their need we have built this system so that they can easily send ground booking requests, membership request and match booking. Users or any visitors can send review on club. Beside users, sponsors can also view upcoming matches and player performance in different matches. Registered members have their accounts so they can logged in to their profile. The start of this system is good we can enhance it for adding other modules. Beside members and clients, club administrator can add teams, select players in a team, scheduling of matches, payments records of members and employees. The system generate tickets and manages sold and booked tickets records.

6.2) Limitations

There are few limitations of this system which means we have built this system by assuming some constraints according to its scope. There is no role of coach, captain, referee and any sponsors. Administrator is the persistent of club so he know how to use system. There is no mechanism of cancelation of ground booking request at specific date and time in case of emergency leave or weather circumstances. The notification only send on email not on phone number. The membership fee is two thousand and member monthly fee is three thousand. The employee salary is from eight thousand to twenty thousand. Only few types of employees are handled by this system such as ground maintainer, security guard, technician, cleaner and pitch maintainer. The date of payment for members or employees is set to fix date 01-selected month-selected year. The fee of ground booking for three hours is five thousand. The details of matches is only managed by this system. The match awards, previous played matched records with score sheets are not managed by this system. The payment process is not involve any bank or credit card. In future we will add payment module which involves a bank for collecting the amount of club. For match ticket booking only one person can book one ticket. After that he has to purchase his booked ticket by paying amount to administrator. Administrator can only allocate fifty tickets for a match. In future we will enhance it. Maximum over for a match are fifty and a team contains eleven players. The only administrator manages this system. There is no role of financial manager and other departments.

Note: We have not build this system like cricinfo.

6.3) Future Enhancements

For future enhancement, we will include other operations like diamond cricket academy records such as trials details and each learner details. It will allow sponsors to elect team from club for advertisement. Cricket Academy System will also be included in the system. This system will support the live matches for clients. The graphs based records such as performance of members will be shown through multiple graphs. We will manages the user requests for membership in club and ground booking for more than two city. Also notification module will enhanced later by which, administrator will send important messages through Short Message Service (SMS). We will add payment module by which users or members will online perform transactions. We will add modules for maintaining the match details with score sheets online by which user will view score sheets of specific match. Another modules will added for live commentary of match with video on which user will comment. A support inbox will also include if someone wants to ask something about club then he will able to write his text on client side then administrator will response it. Automatic notification generator will built for send notifications to that users who are regularly visit websites though email. We will size number of cities so that the people from different city can apply for membership and details of cricket trials will viewed on client side.

References:

[1] Ian Summerville (2011), Software Engineering. United State, America, US: Pearson

[2] Ian Summerville (2011), Software Engineering. United State, America, US: Pearson

[3] Roger S. Pressman (2001), Software Engineering Practitioner's Approach. Avenue of Americas, New York, NY: McGraw-Hill

[4] How to Create Requirements Traceability Matrix (2017, January 12) Retrieved from http://www.guru99.com/traceability-matrix.html

[5] Roger S. Pressman (2001), Software Engineering Practitioner's Approach. Avenue of Americas, New York, NY: McGraw-Hill

[6] Roger S. Pressman (2001), Software Engineering Practitioner's Approach. Avenue of Americas, New York, NY: McGraw-Hill

[7] Roger S. Pressman (2001), Software Engineering Practitioner's Approach. Avenue of Americas, New York, NY: McGraw-Hill

Additional Material

1. Graig Larman (1998). Applying UML and Patterns. Upper Saddle River, NJ, USA：Prentice-Hall

Appendix A

Domain Model