AN OBJECT ORIENTED APPROACH FOR CREATING AN AUTOMATED AND EFFICIENT AUCTION MANAGEMENT SYSTEM (AMS) FOR FRANCHISING SPORTSMEN

N.Vignesh[1], S.Sowmya[2]
[1]Research Associate, IIM Ahmedabad
[2]SDE, ACS Oracle India

Abstract: Indian Premier League and similar cricket tournaments are getting more popular and interesting now. As the edition goes on tournament organizers and spectators are in need of a system which helps to manage all fields in the tournament. This system helps auction managers to save their progress and also assists them in taking certain decisions. This System reveals that when it is implemented there will be no need for managers to maintain the work done and also help managers to choose venue, team etc automatically when the sufficient data is fed. Now the implementation of this system will lead to managers to fed data and conditions and the system will process everything and give the suggestions. This will avoid situations like two matches being fixed in same venue at same time. The actors in this system are bid manager, game manager, transaction manager, site manager, system, data manager, sponsor manager and ticket manager. Managers will contain separate profile for each. Each one will have own username and password which will be required while logging in. The system will take care of all the activities to be done for conducting a tournament.

Keywords: **AMS, auction, tournament, management**

I. FUNCTIONAL REQUIREMENT

LOGGING IN: Each manager should register to the floor. They are given a username and a password. When any record has to be updated in main server these details will be counter checked by the server. Users who are visiting the site has to register and are given a username and password.

BID MANAGEMENT: Bidding should be maintained by the system according to the constraints given by the bidding manager. Online bidding is done. The other constraints are the team member limit of a team and the Maximum amount of Bidding for the whole team. No single player must be sold to two teams.

MATCH SCHEDULING: The match should be scheduled accurately. It is to be noted that no teams has two matches on the same day. It should also be noted that each team should play two matches in their home ground. There should be an alternative date for the matches that were cancelled due to rain or other circumstances if any.

TEAM MANAGEMENT: The team should be selected by the software based on the statistics provided by the game manager. The system should select a maximum of eight batsmen, five bowlers and one wicket keeper within the 11 players. The coach for each team is assigned using the system.

VENUE MANAGEMENT: The Venue for the each match must be allotted by the system based on the match schedule produced by the Schedule Manager to the Game Manager. The system should take care of the constraint that two matches must not be scheduled on the same venue on the same day.

TICKET SALES: The ticket sales are done in the usual way through website. There is an option of cancelling the ticket before 24 hours of the match after which cancellation is not allowed. The rates for the tickets are fixed by the venue manager. The system must keep account of the ticket amount and forward a copy to the Transaction manager.

II. NON – FUNCTIONAL REQUIREMENT

USER INTERFACE: The home page has several options thus enabling each kind of user (a game manager, data manager) to get logged in individually into their respective accounts. Home page also displays current happenings about the tournament. Whenever an administrator enters into his account he finds functionalities such as match scheduling which aren’t available for other normal users. The next page that gets displayed to a normal user whenever a user logs in is the page that contains his own profile information’s and options for booking tickets for any particular match. The user is also posed with a query area where he can post his queries to the site and also a feedback form for providing his feedback about any particular match or player. Validation is done on any user provided data to check against redundancy and spam and then only it gets updated to the user interface.

PERFORMANCE AND MAINTAINABLITY: To provide reliable service Backup server is needed this can act as a standby on the event of failure. As soon as the failure is corrected the main server will get new changes in the backup as a packet. Thus the system is maintained in such a way that no failure will occur for managers while scheduling and every transaction will be atomic.

ERROR HANDLING: Though there will no failure will occur in the system, in case any failure occurs in transaction function while pay function is called, roll back should be done so that the ticket selected has to be unselected.

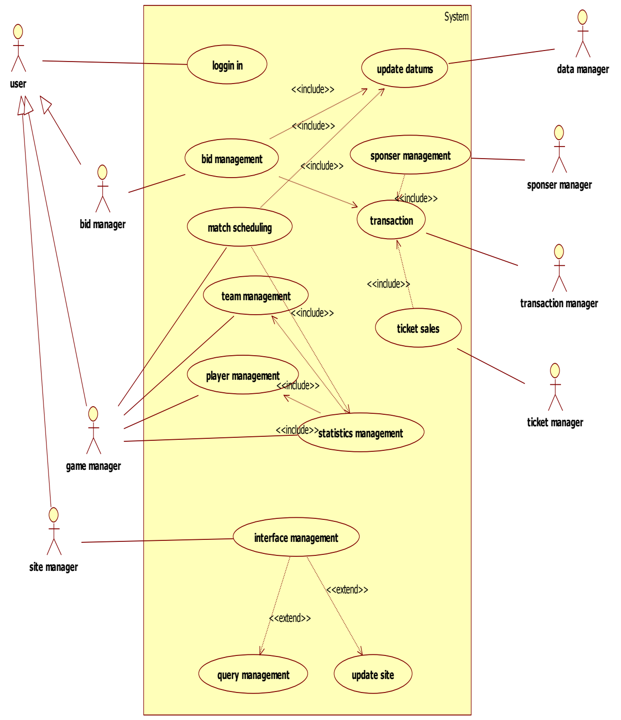

Fig. 1. Use case diagram for the system

The above picture describes the various use cases and actors involved in this system.

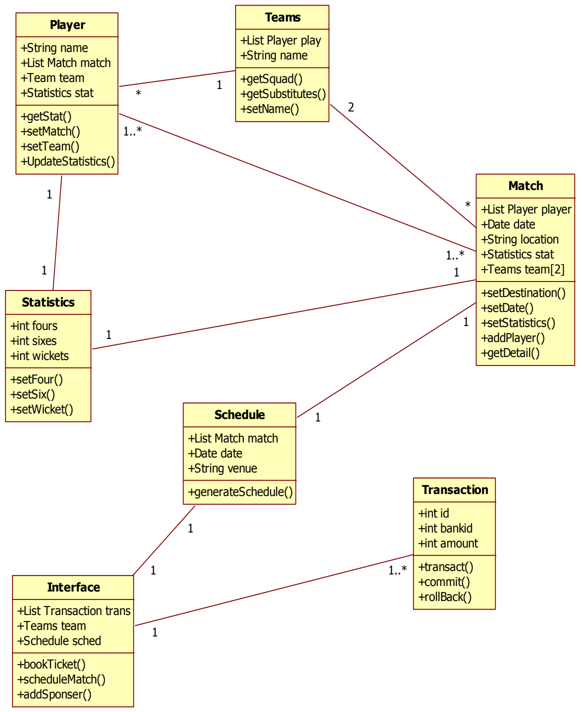

Fig. 2. Class diagram for the system

The above picture gives us an overall domain model and interaction between various objects contained in the Auction Management System (AMS).

III. DESIGN AND IMPLENTAION

This section gives a brief overview of object-oriented design aspects of the AMS with the help of which unknown functionalities of several components is brought under light and it enables us to visualize the components as real world entities which are the major advantages of Object oriented modeling approach. This can be brought by representing several components of the system using various diagrams such as class modeling, sequence diagrams, state activity representation and flow descriptions etc,. The requirements of such diagrams vary over one system to other. This section contains such diagrams which help us in understanding the underlying functionalities of each component in our AMS to a better extent than before.
The various classes available in our project and the descriptions about them are specified below.

MATCH: This contains information related to a particular match and it is associated to other classes such as Statistics, Teams and Schedule since this class uses the information provided by those classes.

PLAYER: This class contains information about each and every player enrolled in the league and it provides several other information regarding the statistical aspects of any individual. This class associates with Match, Statistics and Team to gather and share in formations.

STATISTICS: This class contains both statistical details of both the player and any particular match and it is associated to Match and Player class.

SCHEDULE: This class contains several scheduling information gathered from associated classes and performs several scheduling functionalities. It is associated to classes like Interface and Match.

INTERFACE: This class contains all the presentation tier elements of the AMS system. This receives all the input data from the user and sends them to the controller classes beneath them. This is the VIEW class of our system and it has associations with Schedule and Transaction classes.

TRANSACTION: This class provides the transaction particulars required for the system and it is solely responsible for the financial exchange between our system and external environments. It has association with the Interface class thus enabling the Interface class to display the transactional processing to its users.

TEAMS: This class is plainly the abstraction of several players grouped together based on the match to be played and this class also enables choosing the squad and substitute player for a particular team. It has associations with Player and Match class.

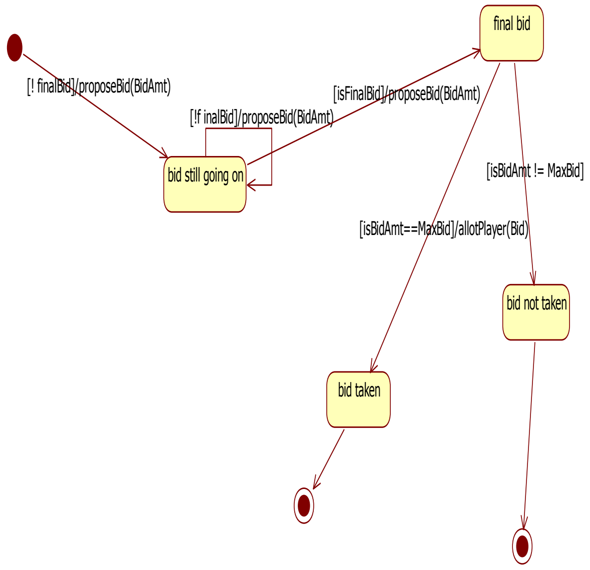
final bid
[! finalBid]/proposeBid(BidAmt)
[isFinalBid]/proposeBid(BidAmt)
[!f inalBid]/proposeBid(BidAmt)
bid still going on
[isBidAmt != MaxBid]
[isBidAmt==MaxBid]/allotPlayer(Bid)
bid not taken
bid taken

Fig. 3. The above picture depicts the state transition diagram of the proposed system.

REALIZATION OF USE CASES IN CLASS DIAGRAM

LOGGING IN: This use case's functionality is implemented using the class interface where several sub functions like getting the user name and password from the user are handled by methods of this class.

TRANSACTION MANAGEMENT: The functionality of this use case is implemented completely in the transaction class which supports the complete financial exchange using its secured methods.

BID MANAGEMENT: This use case's functionalities are visualized using two of the classes.
Team class: which helps the teams to be organized based on the bidding facts.
Transaction class: with the help of which the bidder plays the amount for the particular player under auction.

SPONSER MANAGEMENT: This functionality is brought out using the interface class which contains several methods for managing sponsor requests.

MATCH SCHEDULING: The classes match and schedule work towards achieving this functionality .The scheduled matches are stored in the persistent repository for references.

TEAM MANAGEMENT: The classes' team and player are used to bring out the functionalities provided by this use case and several methods aids in managing the team.

TICKET SALES: This functionality is brought out using the Interface class which has several methods defined in it so as to process and manage the sales of tickets.

QUERY MANAGEMENT: This functionality is also provided by using interface class which has methods defined in it for query processing and management.

PLAYER MANAGEMENT: The player management is taken care of using the classes' player and statistics which has functionalities from bring about the same.

IV. OBJECT ORIENTED APPROACH

Requirement Analysis: This is the first step of the actual project planning which will define all the classes, their attributes and functions, and relationships among them. This will be the base for any further step of the project (modeling or coding).

Design: The next step will be the designing of the system. For this we need to design use case models, interaction models, class and state models which aids in design process.

Interface design: The next step lies in developing application logic for our system and network security for the system under development.

Interface Development: Then we move into the front end of the project. The Interface development refers to the creation of html pages that constitutes the website. This can be done in parallel with the database creation.

Database connection: This phase can start only after the database creation and interface development is over, since it aims at linking the two. Once the connection is given successfully and in accordance with the semantics of the project, the system's development stage is completed. The database here refers to the persistent data store.

Testing Software: Once the development is over, it is important that we check the system for any possible bugs or inconsistency. Hence the testing is done by majority of the resources to make it fool proof. The various testing done includes unit testing, integration testing, system testing and validation.

Documentation: After the system is thoroughly tested, it needs to be documented as they provide interactive customer usage and this documentation also aids the developer to develop coding from documented designs.

V. CONCLUSION

Our Auction Management System when implemented will be an efficient one in conducting tournaments. It could also be used for conducting many other tournaments other than IPL. It makes the work of the Tournament conductors easy. It also provides security by providing username and passwords for each manager of the tournament which makes all the details confidential.

Thus the proposed system has the following advantages such as:

a. Cut costs and improve efficiency
b. Reduce the time spent in preparing an auction sale
c. Free up staff for more productive work in other areas
d. Eliminate double, triple and sometimes quadruple handling of information
e. Automate the invoicing function and do away with hand-written invoices
f. Have total accuracy of records
g. Get instant summary reports by Buyer, by Vendor, by Auction...
h. Automate the Vendor accounting process
i. Work with a software system that has been developed *by* people closely associated with the Auction Industry over many years
j. Work with a software system that has been developed for the people working in the Auction Industry

REFERENCES:

[1] René Caldentey and Gustavo Vulcano “Online Auction and List Price Revenue Management” in *Management Science* May 2007vol. 53 no. 5 795-813.

[2] Roger B. Myerson “Optimal Auction Design”

[3] Hung-Bin Chang, and Kwang-Cheng Chen, “Auction-Based Spectrum Management of Cognitive Radio Networks” in IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 59, NO. 4, MAY 2010 1923

N.Vignesh is working as a Research Associate at IIM, Ahmedabad. He received his B.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Chennai in 2013. His research interest includes Natural Language Processing and Computer Networks.

S.Sowmya is working as a Software Development Engineer at ACS, Oracle India. She received her B.E degree in Computer Science and Engineering from College of Engineering Guindy, Anna University Chennai in 2013. Her research interest includes Natural Language Processing and Computer Networks.