Table of Contents

Chapter one

1.1 Introduction

In the compound of ASTU there are many activities running rather than the educational process. One of these activities is related to the act of sport. There are 43 universities in our country and these universities compute in football match. In order to search and get the best players that the university can represent there are soccer match that the students of different department involve.

Football cup management system is a professional field that focuses on the football games, scores of the game, the player's position of the game, the information given by the coach of the team, the different type of team in the cup, the schedule of the games and other information related to the federation. In ASTU the football cup management system is done by the student union and sport and entertainment club members.

Today students of the ASTU get the information of this cup by the announcement board. This is done manually by the print out papers. But most of the students can't get the

information about the score of the games because they don't know the schedule of the games. In order to make this information exchange modernize our group tries to make a website that every student can access and get information.

In order to update information's we use web based system in which the administrators update different information about the football games. Students in the ASTU can use their smart phone or can use there pc and get login to our websites.

1.2 Background of the organization

At early establishment of the university, the football game was not in organized way since it is started with only small teams and those students play football for only entertainment purposes which was not for cup. After all when the game become developed and known well within students it is decided to arrange department teams and the game held in organized way and the computation increased to cup game which exist now.

1.2.1 Mission of the organization

The sport games that found in ASTU are the most competitive game as a country. Because student's dissatisfaction regarding the game is considered and the sport club is organized well to bring the students complaint into practical. The main purpose of the game is to provide the place where students spend their leisure time without need of go out of university's compound and to give opportunity for those who want to participate in the sport club. This helps the university to compute with other universities and do well. Beside to this since the time of the game doesn't overlap with the class time there is whole student participation in the club.

1.2.2 Vision of the organization

ASTU is the one of well-known universities by its advanced student in education and in the future the university has a will to produce students who are very talented in sport terry that can make the university a role model. ASTU Sport federation club has a big role to achieve this goal.

1.3 Objectives of the project

1.3.1 General Objectives

The general objective of this project is to design and develop ASTU football cup management system. Our system give information about ASTU football cup for new visitor of the site. It helps to update new announcement by the coach and other administrators. The objective of developing this site is to create the good interaction of the viewers of the site and the football administrators in order to improve the system

1.3.2 Specific Objective

✓ To store the database of teams and each member's

- ✓ Display the information of the team and players when required
- ✓ To register new players and add the qualified ones in to the team
- ✓ It help the coach to create its team and the members
- ✓ To identify all the teams that participate in the foot

1.4 feasibility studies

Feasibility studies refer to the study of discovering the strengths, weaknesses, opportunities and threats of the existing or proposed business thereby aiming at the success. The focused area of the study is the investment and the return. We check the feasibility in order to evaluate the cost and benefits of the system which we are going to develop. The feasibility evaluates economical, technical and operational area of the project. Based on the result of the feasibility study we proceed or cancel the project.

1.4.1 Technical feasibility

Technical feasibility study is the complete study of the project in terms of input, processes, output, fields, programs and procedures. It is a very effective tool for long term planning and trouble shooting. We need our system to be technically feasible so we divide and conquer the work based on our ability's. Hardware and software resources are available for the development and implementation of proposed system. Therefore we can say the project is technically feasible.

1.4.2 Operational feasibility

Operational feasibility refers to the measure of solving problems with the help of a new proposed system. It helps in taking advantage of the opportunities and fulfills the requirements as identified during the development of the project. It takes care that the management and the users support the project. Operational feasibility is how the system that we develop solves the problems of the past system.

- ❖ Process – the main input from the user are simple and easy
- ❖ Evaluation—user of this system be benefited. This will make easy access to the users to see the score of the games.
- ❖ Implementation—
- ❖ Resistance—

1.4.3 Economical feasibility

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study.

1.5 Problem of statement

The main problem faced by the manual system is that it leads to huge wastage of paper to maintain the record. Even it takes time to search a particular record manually. There is a change of misplacing.

Need for the new system

- The actual need of the new system is to reduce time and paper work.
- The system can store details of player, coaches, times …etc.
- The system is to reduce complexity system for the user handling usres handling the system.
- The system is web-based software.

1.6 Scope and limitation

1.6.1 Scope of the study

Scope of the project is all about the range up to which the system perform task. The system developed in order to manage ASTU football cup in a way that users, administrator and each team members can communicate and share information easily. Its scope includes new members registration, team match schedule, every time updated data about the score of the game or users profile, it allows users to see new announcement or any notification from the admin or coach. It paves way for user to give comment regarding any cup, It provide different login in order to protect file from unauthorized access and make a way in which some of private information can be shared between coach and team members. The restriction of the system is that while there are three cups, inter department, half-life and GC cup but all cannot be active at the same time because the season matters the functionality of cup system. Also the cup management system does not include any other sport rather than football cup for only male students. But there is a way in which developer can add the left functional modules any time in the future.

1.7 Limitation of the project

- ➢ Since it is web based system it works by internet and the system is not accessed by the user as long as there is no connection.
- ➢ For the data that is updated by only admin the important information can be delivered late because of weakness of the individual and this makes system's work jog along in nothing special.

➢ People need to computer or access desktop well to use the system

1.8 Constraints and Assumptions

We know that many users can join to the website and if the data is not secured unimportant change can be happening to the stored data. Thus, as a constraint user should not inter to admin Page which means admin and coach login should be locked by password to avoid unauthorized access. Users and team members are not allowed to delete data rather than only register and leave the team.

1.9 significance of the project

This system is designed to help ease the complexities faced by ASTU sport club, however being designed for small cup it can also grow to fit the big sport events.

- allows students who have little time on their hands, with sport passion join just a click of a button.

- the couch can access his team information without calling them in.

- the audience can keep watching once own favorite team's progress.

- university administration can choose a team that can represent them for national universities match by just checking our system.

- players can be recruited for the national's based on their performance throughout the year. And that could be recorded on the system we designed.

1.10   operating environment

1.10.1   hardware and software tools

1.10.1.1 software tools

Tools	Used for
MS-word 2016	For documentation
MySQL	For the database
NetBeans	Coding
Sublime	Html coding
Visual studio	Development
Enterprise Architecture	Diagrams

Tools	Used for
Personal computer	To type documentation as well as coding was done mainly using a laptop.
Flash	To transfer different kinds of files. (documentation and coding)
Note book and pen	To jot down important issues; most of the requirement analysis was collected using note book and pen

1.11 Methodology

Data collection method

We use the following collection methods

- Interview
  We have made interview with students who are member of student union and sport club in ASTU.
- Observation
  We have observed some inter department games.

- Document analysis
  We have analyzed document (proposal) which is stored in the student union main office that shows the over view of the existing system.

*System analysis and system design

In this project we are going to use object-oriented system development methodology.
- Object oriented analysis
  In this phase we apply object-oriented analysis technique approach for analyzing the system by applying object-oriented programming.
- Object oriented design
  In this phase we apply object-oriented design technique approach for design the system by applying object-oriented programming.

Deployment /implementation methods

Agile Implementation Methodology. The **agile** software development **approach** is designed to cope with the complexity of modern software projects and to allow these projects to be designed in a more flexible manner.

*System development method

We choose agile model. Agile approach is considered to be better.it uses incremental approach a sample prototype discussed with customer.

Agile implementation is a form of project management that works in small increments and well it suited to projects that could become irrelevant once delivered, especially useful in software development. The key to the agile plan is that it provides flexibility for changes to the product as it continues to be developed.

Benefit of agile

Agile is flexible, a huge benefit when the customer requirements or priorities frequently change. Agile also gets the product to the market faster and has better team communication

More flexible

Agile adapts to change better Requirements can be added, deleted, or changed at the beginning of each iterative cycle, guiding the project team on the work for the next iteration. The customer can also shift priorities at the beginning of each cycle, allowing the project team to keep up with the customer's business needs.

Product get market faster

The product gets to market faster using agile because the focus is on working deliverables, not finished products. The requirements defined during a particular iteration are built into a working model.

Better communication

Because the project team works closely with the customer to revisit requirements and priority.

1.12 Constraints and Assumptions

1.12 .1Constraints
➢ Shortage of resource: the system does not provide all cups information at the same time because the season determines the availability of cups information.
➢ The users of the system will have limited access to the system.

1.12.2 Assumptions

We assumed that:

- ✓ the user of the system has basic computer knowledge
- ✓ there is full resources like appropriate technologies and full internet access

1.13 Testing plan

The goal of testing was to demonstrate that the program under control contains bugs.

Unit test the objective of unit testing is to isolate a section of codes and verify its correcteness.it may be an individual or procedural.

Black Box Testing is a software **testing** method in which the internal structure/ design/ implementation of the item being **tested** is NOT known to the **tester**.

White Box Testing is a software **testing** method in which the internal structure/ design/ implementation of the item being **tested** is known to the **tester**.

Integration test We are test the performed to expose defects in the interfaces and in the interactions between integrated components or systems.

1.14 Over view of the project phase

The system development model refers to the frame work that is used to structure plane and control the process of the developing the system there are different type of system managements for ASTU football cup management system. We chose agile method. Because agile method intend to develop system more quickly with limited time spent on analysis and design.

There are 3 phases

- ✓ -Requirement gathering
- ✓ -documentation preparation phase

✓ -Implementation

1.15 project plan

1.15.1 Budget

Material	Amount	Unit Price	Total Price
Pen	5	6	30
Notebook	5	13	65
Flash	2	250	500
Total			595

1.16 Team composition

1.16.1 Team members activity and responsibility

Name	Id	Responsibility	Mail Address
Tihitena Mesfin (team leader)	A/UR4921/09	Coordinating, Analysis, Implementation, Programming, Designing, Editor	hannamesfin111@gmail.com
Tigist Mekonnen	A/UR4930/09	Requirement gathering, Analysis, Implementation, Programming	Tg.makec@gmail.com
Tsion Merid	A/UR4884/09	Testing, Programing, Designing, Analysis, Implementation	tsionmerid@gmail.com

Samiya Sultan	A/UR4958/09	Requirement gathering, Designing, Programming, Implementation, Testing.	Sultansamiyah1@gmail.com
Yerosen Birhanu	A/UR4254/09	Requirement gathering, Analysis, programming, Implementation, Testing	Yerobr21@gmail.com

1.16.2 Task and schedule

Phases	Months				
	Nov16 - Nov26	Dec1 - Dec20	Dec21 - Dec30	Dec - May18	May19 - Jun 10
Requirement gathering and Analysis	■				
Design		■			
Implementation			■		
Testing				■	
Maintenance					■
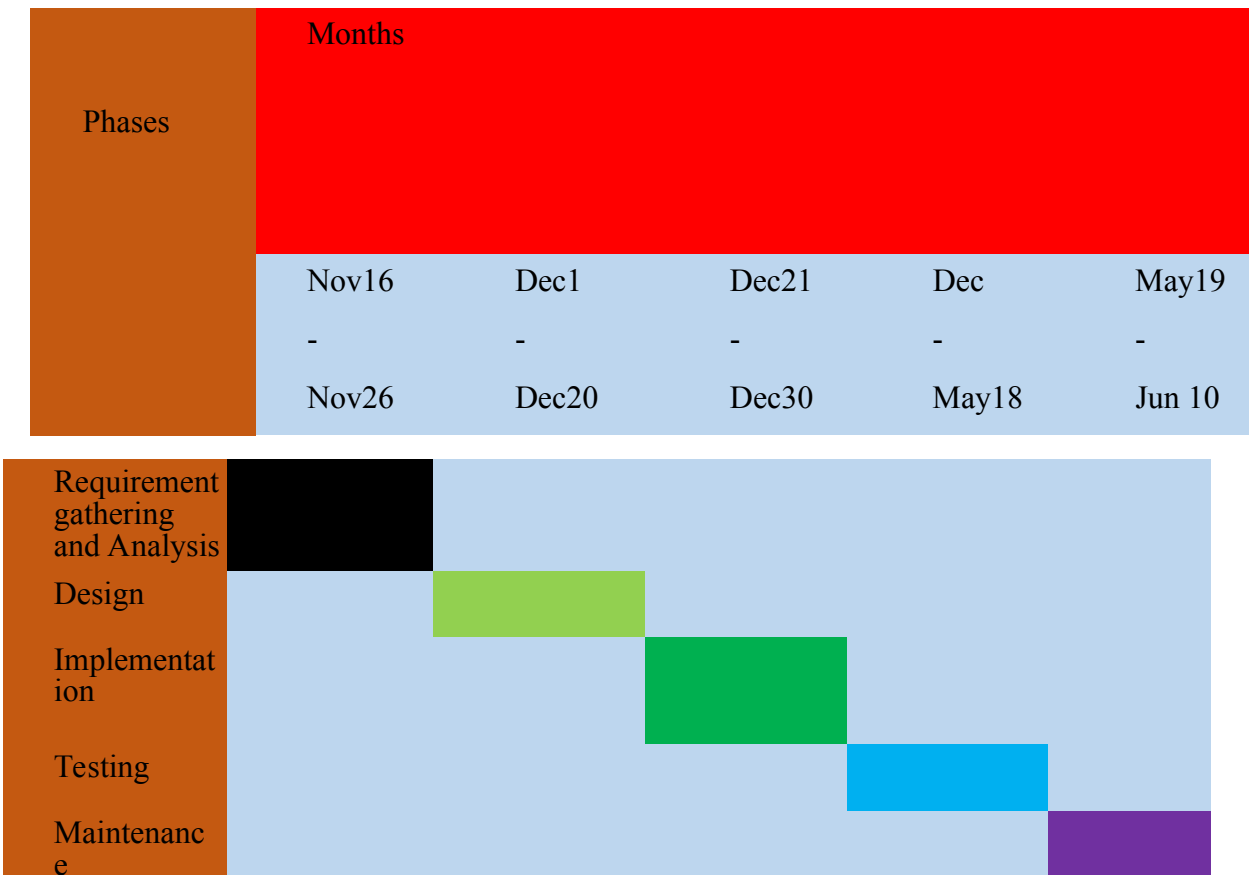
task and schedule

Chapter two

2.1   Description of the existing system

The existing system is very well manual and traditional as well as uncoordinated. Since the football event is mainly managed by students in the sport club and student union, there hasn't been enough budget to computerized it. The current system basically conducts matches and organize group selections.

But in a small extent they notify about the matches that are being held at the time.as a result the target of our system is to provide the football system in the campus with web based computerized system.

2.2   Major function of the existing system

- Register players
- Choose the qualified ones to a team
- Assign groups

- Assign coach to a certain team
- Schedule matches
- Assign a referee to a game
- Conduct matches
- Release notifications
- Award the winner

2.3 user of the current system

- Students
    - Students register for team membership
    - Student see the schedule
    - Students see the score of the game
- Players
    - Players see the training schedule
    - They will see their playing position
- Coach
    - Coach can get the best players
    - Can set training
    - Can tell the players positions easily
- Referee
    - o Can report the card he has given
    - o Can see which game he is assigned to
- Administrators
    - ❖ Can easily announce the schedule
    - ❖ Can set the score of each game

2.4   Drawback of current system

- Manual management system is hard and complicated.
    - Human error

Try as we might, human beings are more prone to error than computer.
While a computer system has the occasional glitch, it generally records data and organizes it in a
more accurate than human beings, with less need for error checking. Manual data processing
requires for more eyes to check and double check data for accurate.

- Speed

It takes a lot of time to enter data manually. Computers are generally

faster than humans.

- Since notification is not published well audience do not attend all the events prepared.
- Fans don't have all the information they need on their favorite or opponent team.
- Since the process is managed by student who are members of student union, they are not very well experienced.

2.5 Software Process Model

Now time it is difficult or impossible to predict how computer based system (e.g. web based app) will evolve as time passes because market condition change rapidly, end users need evolve, new competition threat emerge without warning, a usage scenarios might have to be modified, a list of functions may be extended which means new requirement should be added as needed any time.

In our software process we will use Agile model with different methods that involved in this model. Among those methods we will use combination of extreme programming, Dynamic systems development method together with other methods because of the above and following reason.

Agile software development model encourage our project to:

- ✓ Add any required functionality and modify scenarios any time as needed
- ✓ Work in Small, highly motivated project team
- ✓ Use informal methods (less formal and reduced scope)
- ✓ Customer satisfaction and early incremental delivery of the software
- ✓ Edit written specifications
- ✓ Make software engineering work products minimal
- ✓ Overall development simplicity
- ✓ to deliver working software quickly
- ✓ Speed up or bypass one or more life cycle phase (based on iterative approach to software development)
- ✓ Active and continuous communication between developers and customers

Agile software development model uses different methods. Among those methods XP and DSDM suits for our project.

Extreme programming method

We use XP in our project because this method suits for the projects those have the following behaviors

- For small to medium sized teams developing software with vague or rapidly changing requirements
- Coding is the key activity throughout a software project
- Communication among teammates is done with code and necessary documentation is done by decision as its importance and our project have these behaviors.

Among practice of XP that we prefer using more in our project are:

- ❖ Small releases: put a simple system in to production, then release new versions in very short cycle
- ❖ Metaphor: all development is guided by simple shared story of how the whole system works
- ❖ Simple design: system is designed as simply as possible (extra complexity removed as soon as found)
- ❖ Refactoring: programmers continuously restructure the system without changing its behavior to remove duplication and simplicity
- ❖ Pair programming: all production code is written with two programmers at one time
- ❖ Collectively ownership: any one can change any code anywhere in the system at any time (off the developers)
- ❖ Continuous integration: integrate and build the system many times a day, any time a task is completed
- ❖ On-site customer: a user is on the team and available full time to answer questions
- ❖ Coding standards programmers write all code in accordance with rule emphasis communication through the code.

2.6 Business Rule
The following are business rules of the current system

- ➤ The nominators group and the coaches jointly select player for inter department cup based on their performance on previous matches.
- ➤ Teams are composed of players from the same department.
- ➤ Sport and entertainment club with their members arrange the schedule.
- ➤ Sport and entertainment club post match results on banner around "Anfi" and on telegram channel named "astu arena".
- ➤ Sport and entertainment club reward the champions.

Chapter 3

3. Proposed System

3.1. Overview

The system we are going to develop is intended to replace the manual system of inter department club management system in to computerized and making data storage well organized. The existing system mainly provide match related information but our system will provide additional information like players' profile, current status, etc and these information are easily and timely available for audiences.

3.2. Functional Requirement

Functional requirements define what the system supposed to do.

1. Manage users (privileged accounts are granted for players and coaches).

2. Register applicants.

3. Announce applicants who are accepted.

4. Update players' information.

5. Create schedules.

6. Update schedules.

7. Generate fixtures

8. Create score board

9. Assign Referee

10. Assign Coach

11. Provide information about the entire activity of the Cup accordingly for the users.

- The coach informs training schedules for his players.
- The admin provides schedules of matches and results for the visitors.

3.3. Non-functional Requirement

Non-functional requirements define how the system supposed to be.

A. Usability: the system should be easy and understandable for users by making the steps as small as possible and simple.

B. Authentication requirement: the system allows users to read only without password and username, otherwise authentication is required only authorized users can modify the content.

C. User Interface requirement: should be easy to understand by using familiar and common graphical interface that the user can interact without difficulty.

D. Error handling requirement: error will be handled accordingly. Mostly error will occur due to improper use of the system by the users. This kind of error will be handled by making the system to generate error message that describes the cause and possible solutions if possible.

E. Documentation: every activity of the development, design, implementation and other processes should be documented for future reference and also maintenance during system failure.

3.4. System Model

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. System modeling has generally come to mean representing the system using some kind of graphical notation.

3.4.1 Scenario

Scenarios are real life example of how a system can be used. It can describe the starting situation and normal flow of events.

Uses pre-conditions

1. They must have pc or desktop computers to launch our web
2. The have to enter our address www.ASTUfootballcup.edu.et
3. Then the user can get our homepage

Scenario 1 – see notification

Flow of event

1. Fan must go to our address and then select the cup tab
2. Select the cup that they want to get notification about
3. Then they will get another tab that say notification
4. They can see all the information about that cup, the recently played games, score of the games and other announcement

Alternative conditions

-the game in the ASTU are seasonal games, so they will open in the given time period only

Scenario 2 –Register

Pre-condition – this visitors must be the student of ASTU , they must register on the time period of the registration, must give the correct information about them selves

Flow of event

1. After they chose the type of cup they see the register tab
2. Then they see the register form
3. Fill the form and click ok button

Alternative conditions

-if he fills the form incorrectly then the system generates an error message

-if he has been registered already the system generate already registered message

Scenario 3- login

Pre-condition – both the coach and players must have the correct user id, name and password

given by the admin

- The admin should be the real member of the administration team

Flow of event-first launch to a web

-for admin click the admin button on the home page

-enter the user name and password click ok

-select the cup tab then click ok

-enter user name and password

Alternative condition – if the user name and password they entered is wrong then the system will generate error message

-if they forget the password will click forget password button

Scenario 3-see news

Pre-condition—first the player should register and chosen for the team, and the admin send username and password that the player can access the system

Flow of event

1. First they have to choose the cup that they are member off
2. Then select the login tab
3. Then they have to enter their name and password and press enter
4. Then they click the news tab and get the updated information about their team and other related information

Alternative conditions

- If the username and password they enter are incorrect the system must generate error massage

-If they forget the password they can recover their password using **Do you forget** button

Scenarios 4-view and edit profile

 Pre-conditions- they have to be one of the members of the team,

Flow of event

1. Chose the view profile tab
2. See the form that contain his profile
3. Click the edit button and edit the form
4. Click the save button after you edit
5. Click cancel if you don't want to save the updated form

Alternative condition

-Some of the forms can't be changed by the player or the admin must check for the changes that are made

Scenario 5-choose players

Pre-conditions- the coach have all the list of players that register for team,

Flow of event

1. After choosing the tab the coach
2. Choose the icon that say login as a coach
3. Enter the username and password click ok
4. Click the tab that say view list
5. Select the players that are good for the team and click submit

Alternative conditions – if they enter incorrect password they will receive error message from the system.

Scenarios 6-notify training schedule

Flow of event

1. Login using username and password
2. Write on the newsfeed about schedule of the training
3. Write the time, date and place

Scenarios 7-manage account

Pre-conditions- they have to be chosen for the team, so the admin login and signup them and give username and password to the players. They have to be members of the coach staff

Flow of event

1. Admin go to the home page and select the admin tab
2. They can select the login button
3. Enter username and password and click ok
4. In the next interface click the signup button
5. Fill the signup form for each the players and coach
6. Click save and they have username and password that they can login too
7. Admins can send this username and password by their emails.

Scenarios 8- see fixture and score board

Pre-conditions- login to the web and select the cup

Flow of event

1. Launch our web
2. Select the cup you want to see fixture and score board
3. Select the fixture tab and see the schedule of the match
4. Select the score board and see the result of the recently played game

Scenarios 9-manage schedule

Pre-condition-login to the web, select admin button and login

Flow of event

1. After launching the web and login to the system
2. Select the manage schedule button
3. Select the time and place of the game
4. Assign referee for a particular game

5. Save and post into the fan page

Alternative condition- while setting the schedule chose the available time and date so as there is no overlapping of schedule

3.4.2 Use case mode

Use case model is a scenario-based technique in UML which identify the actors in an interaction and which describe the interaction itself. A set of use cases should describe all possible interactions with the system.

Actor identification

Actors are some people that accomplish use case up on a system

Name; fans / visitors

Description-visitors are students of ASTU that want to get information about ASTU cup and visit the page

Rule-see notifications, see the result of the game, see schedule, and apply to be player

Name; player

Description-student of ASTU that are member of different team in ASTU cup, which has user name and password in order to login

Rule –see announcement made by the coach, see the playing position for the game, view their own personal profile and edit

Name; coach

Description-this are society of sport that can guide and coach other teams. Coaches also have user account to access the system

Rule –select members of team and assign position, announce training schedule,

Name –admin

Description- the top authority of ASTU football cup federation that can grant access to the system by giving user name and password,

Rule – assign referee, assign coach for every team, prepare the game schedule, and conduct the result of each game

Name-System

Description-the system that contain all the database of the players, team, coach, referee and other administration staff members

Rule-save the registration form, has the profile of all the players, coach, and refree. Give the members their own special user name and password

Use case description

Table for use case description for seeing notification, fixture and score board

Use case name	See notifications, fixture, score board
Use case ID	1
Use case description	To see announcements, schedule and result
Actor	Visitors /Fan /player/coach/ any society of ASTU
Pre-conditions	Visitors must launch our system
Post-conditions	The system provide information
Main flow	A, visitor open the home page of our system and chose the cup
	B, the visitor go to the notification page
	C, system display the desired notifications
	D, the visitor select the fixture and scoreboard tab
	E, see the announcement
Exceptional flow	-May be the visitors enter incorrect URL
	-if the visitor request information about a cup which is not held in that moment.
Include	View , comment
Frequency of use	Depend on the visiting of the user

Table for use case description for register

Use case name	Register
Use case ID	2
Use case description	To register and fill the form into the database
Actor	Visitor/fans
Pre-conditions	Visitors must launch our system
Post-conditions	The system register the personal information in to the database
Main flow	A, visitor open the home page of our system and chose the cup B, the visitor click the register button c, system the register form D, the visitor fill all the asked question E, click ok button
Exceptional flow	-May be the visitors enter incorrect information -if the visitor is already registered
Include	Fill the question and comment personal experience
Frequency of use	They are registered only one time but can depend on the user

Table for use case descriptions for players see news

Use case name	See news
Use case ID	3
Use case description	To see announcements made by coach and admin
Actor	Players
Pre-conditions	Players must be the member of a team and have account to the system

Post-conditions	The system provide information
Main flow	A, player open the home page of our system and chose the cup
	B, Players enter their user name and password
	C, enter login button
	D, select news tab and see recently updated news
Exceptional flow	-if player enter incorrect username and password the system will generate error message
	-if the player forget his password he have to recover his password by clicking forget password button
Include	View , comment
Frequency of use	Depend on the visiting of the user

Table user description for view and edit profile

Use case name	View and edit profile
Use case ID	4
Use case description	To see players personal profile and edit
Actor	Players
Pre-conditions	-Players must be the member of a team and have account to the system
	-Players profile database should exist in the system
Post-conditions	The system provide information and edit the information
Main flow	A, player open the home page of our system and chose the cup
	B, Players enter their user name and password

	C, enter login button
	D, select view profile tab and see their profile
	E, click edit button and edit some information's
	F, click save button to save updated information's
Exceptional flow	-if player enter incorrect information the system tell them to correct it
	-if the player can't edit some information's it will be restricted
Include	View , edit, save
Frequency of use	Depend on the visiting of the user

Table for choosing player

Use case name	Choosing player
Use case ID	5
Use case description	Choosing players for the team and for the game
Actor	Coach
Pre-conditions	-visitors must register for the team and come to the training day and show their performance
	-coach select the best players for the team
	- also select players for the given game
Post-conditions	The system provide information about each players
Main flow	A, coach open the home page of our system and chose the admin tab
	B, coach enter their user name and password
	C, enter login button
	D, select view list tab and see the list of players
	E, click submit button

Exceptional flow	-if coach enter more than the appropriate number of players the system tell them to correct it
Include	View , save, submit
Frequency of use	Depend on the visiting of the user

Table for use case description of notifying schedule

Use case name	Notifying schedule
Use case ID	6
Use case description	Setting schedule time and place
Actor	Coach
Pre-conditions	-visitors must be member of the team -coach select the appropriate schedule for training players of the team
Post-conditions	The system provide information about training schedule
Main flow	A, coach open the home page of our system and chose the admin tab B, coach enter their user name and password C, enter login button D, write on the newsfeed page about schedule E, click submit button
Exceptional flow	-if there are redundancy of schedule the system the generate error massage
Include	View , save, submit
Frequency of use	Depend on the visiting of the user

Table for use case description of manage account

Use case name	Managing account

Use case ID	7
Use case description	Granting access to other users
Actor	Administration
Pre-conditions	-visitors must register for the team and they have to be chosen for the team -coach select the best players for the team and submit to the admin - admin signup the users and give them username and password
Post-conditions	The system provide information about each players
Main flow	A, coach open the home page of our system and chose the admin tab B, coach enter their user name and password C, enter login buttoD, select view list tab and see the list of players E, click submit button
Exceptional flow	-if coach enter more than the appropriate number of players the system tell them to correct it
Include	View , save, submit
Frequency of use	Depend on the visiting of the user

Table for use case description of

Use case name	Manage schedule
Use case ID	8
Use case description	
Actor	Admin and system
Pre-condition	-Checking for the time and date for the next game -checking for available referee

Post-condition	-setting the schedule Assigning referee and posting on the player and user page
Main flow	-open the web page -enter the user name and password and login -select the manage schedule tab - set the appropriate schedule time, date and assign referee
Exceptional flow	-there may be overlapping of schedule -search for available time date and place for the games
Include	-Create, view, update, save
Frequency of use	-For each game

b. Use Case Identification

The system includes the following use cases:-

- Login
- Manage Account
  - Create accounts
  - Delete accounts
  - Update accounts
- Release information
  - Fixtures
  - Training schedules
  - Match results
  - Team's general status
  - Match report
- View Content
  - View teams' general status
  - View players' profile
  - View fixtures
  - View Training schedules
  - View new players list
  - View scores
  - View database

- ➢ Update content
  - • Update teams' general status
  - • Update players' profile
  - • Update fixtures
  - • Update training schedules
- ➢ Apply
- ➢ Announce accepted applicants
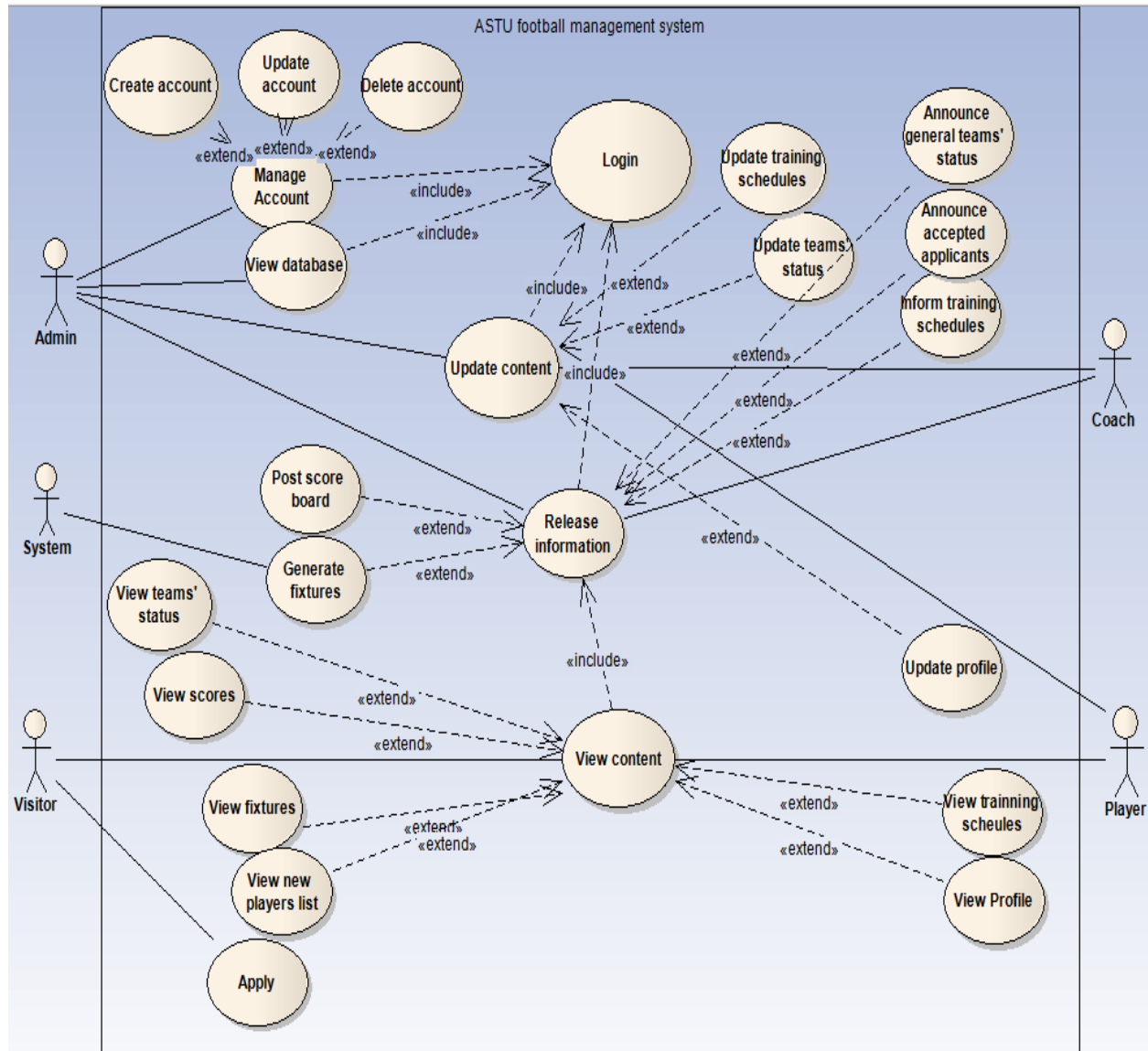
3.5   Object Model

3.5.1   Data dictionary

Classes	Attributes	Operations	Descriptions

Administration	Aname, A_ID, Position	Register() AssignRefree() Notify() AssignCoach() AssignGroups() Remove()	Register, release notifications, assigns coach, referee and groups, also can deny access
Coach	Cname, C_ID, Team	Select() AssignPosition() Notify()	Selects qualified players, assigns positions and notify the changes
Recruited Player	Pname, Team, height, Weight, bdm, age, year, coach	View()	View whether they made the list
Candidate player	Pname, team, height, med_ history, weight Age, school, department, year, Pemail, phoneNo, preferredpostion, status, town	Registers()	Register to be a player
Team	Department,coach,players,rank,points		
Team player	Pname,P_id,goals,ranks,positon,coach	Login() View() Logout()	Logs in, see notifications about changes made by the couch. and logs out
Referee	Rname,Rid,Match	Login() View() Logout	Logs in, views a match they are assigned to and logs out
Group	groupA, GroupB,groupC,groupD		
Match	Team, Match_time,Match_date,result		
Fan page		View()	View notifications made by the administration

3.5.2 Class diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

1. Analysis and design of the static view of an application.
2. Describe responsibilities of a system.
3. Base for component and deployment diagrams.
4. Forward and reverse engineering.

Class model

The comments were to correct the relationships among each class and to carefully anilize the generalized and agggregated classes

class Class Model

team
- list_id: int
- tid: int
+ tname: varchar

+ register() : void

fixture
- fix_id: int
- rid: int
- tid1: int
- tid2: int
+ time: int

+ view_fix() : void

card
- card_id: int
+ color: varchar
+ number: int
- pid: int
- referee_id: int

+ viewcard()() : void

player-list
- list_id: int
- pid: int

+ view() : void

score-board
- fix_id: int
+ result: int
- SB_id: int

+ view()() : void

Registration-form
+/ age: int
+ cup: varchar
+ dept: varchar
+ height: float
- id: int
+ name: varchar
+ position: varchar
+ status: varchar
+ weight: float
+ year: int

+ submit() : void
+ view() : void

player
+/ age: int
+ noOfcards: int
- pid: int
+ pname: varchar
- tid: int
+ weight: float

+ login() : void
+ update() : void
+ view() : void

coach
- cid: int
+ cname: varchar
- tid: int

+ createlist() : void
- login() : void
+ notify() : void
+ updatelist() : void
+ viewlist() : void

referee
- fix_id: int
- rid: int
+ rname: varchar

+ createCard() : void
+ login() : void
+ updateCard() : void
+ view_fix() : void
+ viewcard() : void

authorized-users
- password: int
- Uid: int
+ Uname: varchar

- login() : void

adiminstration
- ad_id: int
+ ad_name: varchar
+ ad_position: varchar

+ assign() : void
+ login() : void
+ update() : void
+ view() : void

3.6. Dynamic model

3.6.1. Sequence diagram

 Sequence diagram shows the order of activities in which the operation is done. It shows flow of events in interaction with the objects.

- ✓ A brief description
- ✓ Precondition
- ✓ Primary flow of event alternative flow of event
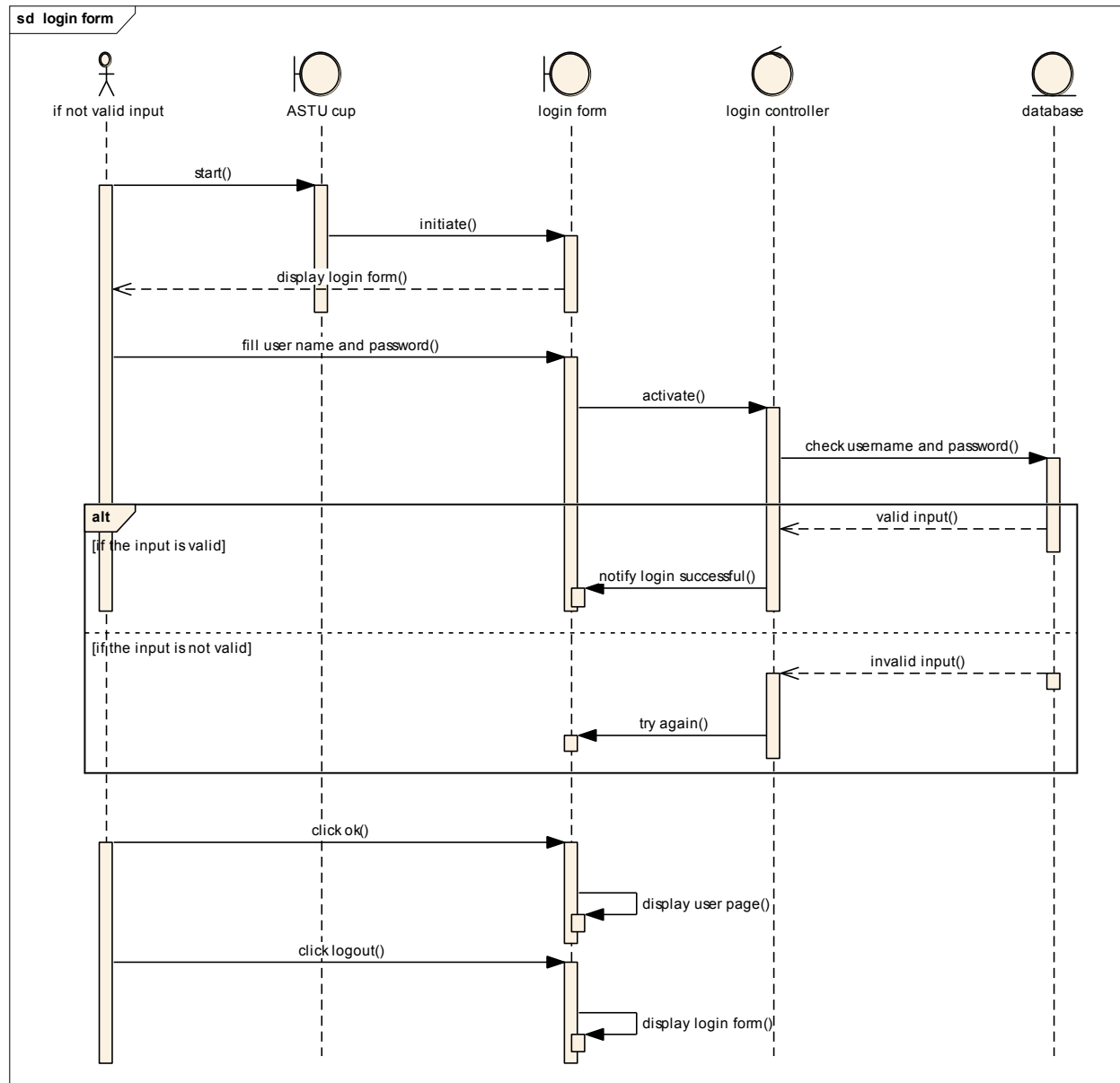- ✓ Post condition

A brief description describes what that use case will do and the precondition for a use case lists any conditions that have to be met before the use case can start at all. The flow of event describes steps to execute the functionality in the use case. Primary and alternative flow of event includes:

- ➢ How the use case starts
- ➢ The various paths through the use case
- ➢ The normal or primary flow through the use case
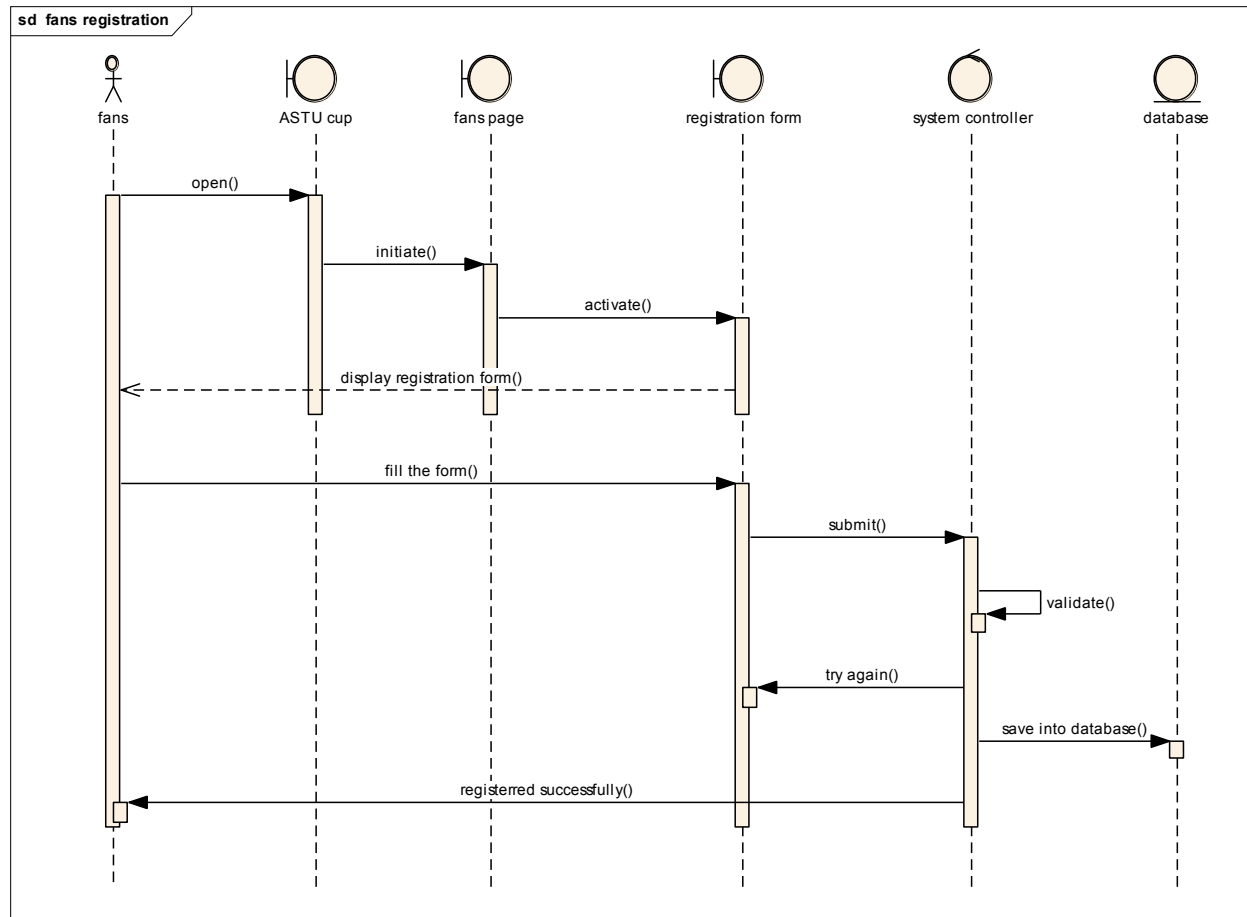- ➢ An error flow
- ➢ How the use case ends

There are three different objects in the sequence diagram. These are:

- Boundary object
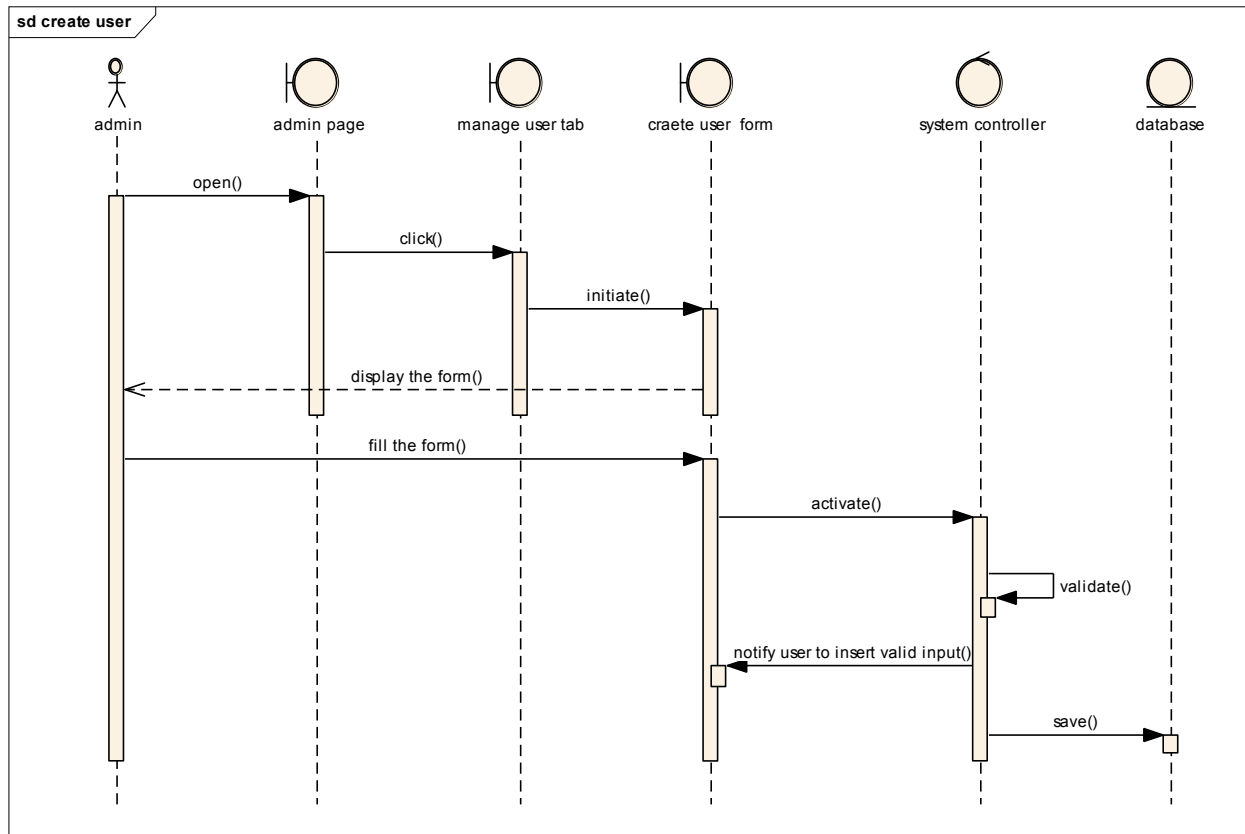- Persistent object and
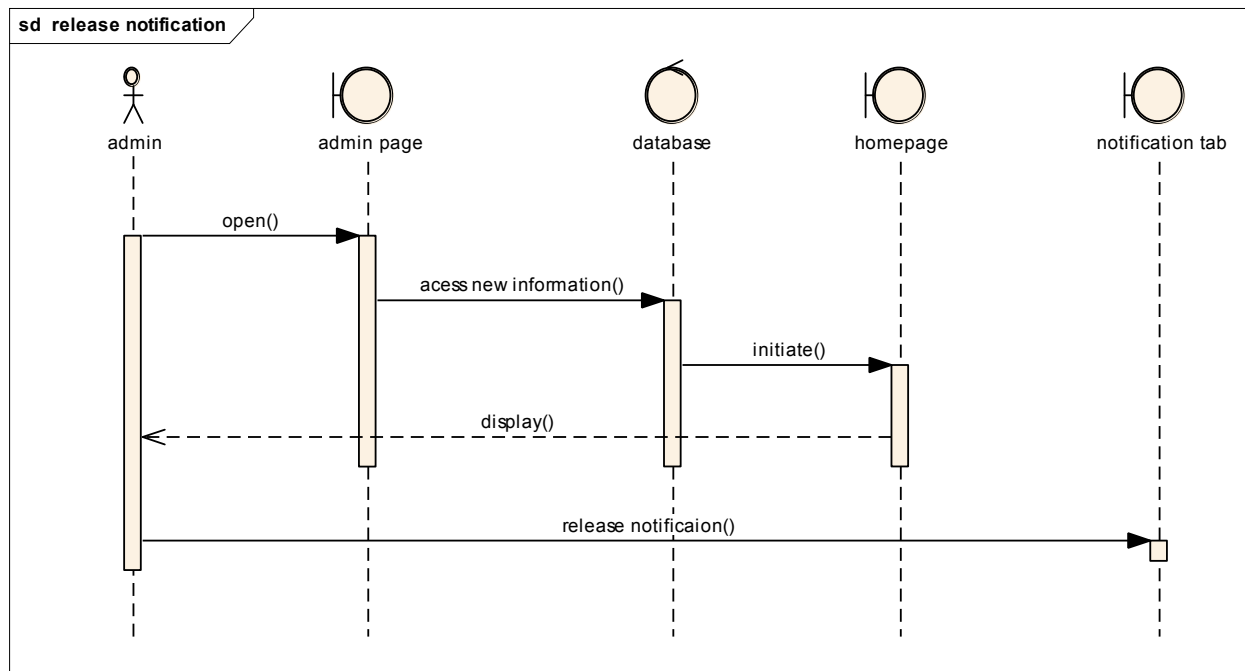- Control object

1. Login for users

sd login form

```
if not valid input        ASTU cup        login form        login controller        database

        start()
   ─────────────▶
                    initiate()
              ─────────────▶
        display login form()
   ◀─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─

   fill user name and password()
   ──────────────────────────▶
                              activate()
                         ─────────────▶
                                   check username and password()
                              ──────────────────────────▶
```
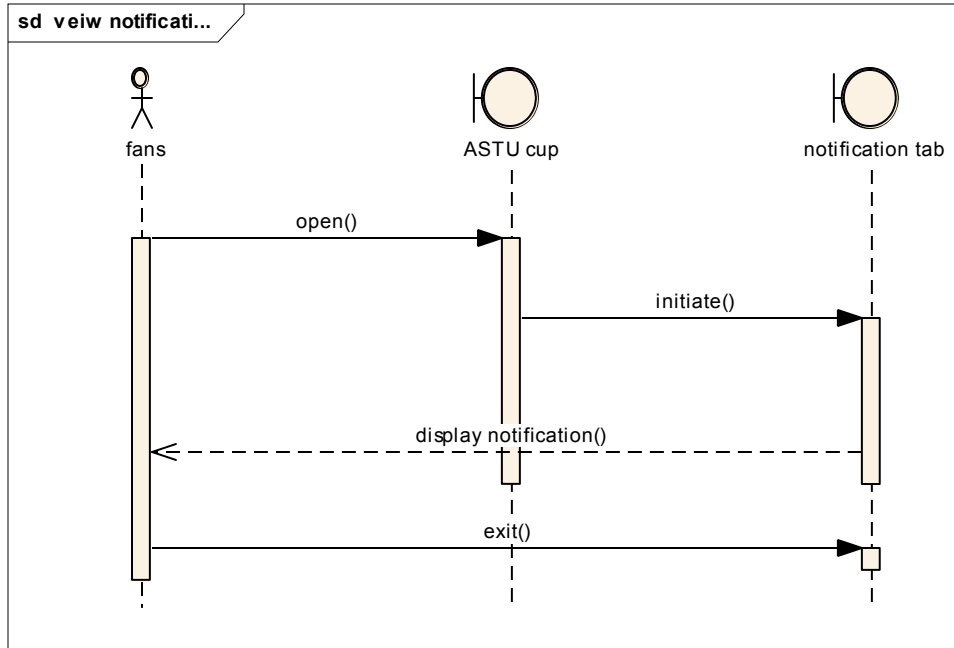
alt

[if the input is valid]

```
                                        valid input()
                                   ◀─ ─ ─ ─ ─ ─ ─ ─ ─
              notify login successful()
         ◀──────────────────────
```

[if the input is not valid]

```
                                        invalid input()
                                   ◀─ ─ ─ ─ ─ ─ ─ ─ ─
                   try again()
         ◀──────────────────────
```

```
        click ok()
   ──────────────────────────▶
                         display user page()
                    ◀──────
        click logout()
   ──────────────────────────▶
                         display login form()
                    ◀──────
```

2. Fans registration

sd fans registration

fans — ASTU cup — fans page — registration form — system controller — database

- open()
- initiate()
- activate()
- display registration form()
- fill the form()
- submit()
- validate()
- try again()
- save into database()
- registerred successfully()

3. Create user

4. Release notification

5. View notification

sd veiw notificati...

fans — open() → ASTU cup
ASTU cup — initiate() → notification tab
notification tab --- display notification() --→ fans
fans — exit() → notification tab

6. Select players

sd choose player

coach — open() → home page
home page — open() → ASTU cup
ASTU cup — select() → specific cup
specific cup — log into() → user page
--- display coach page() ---→ coach
user page — select best players() → player list
player list — save into database() → database
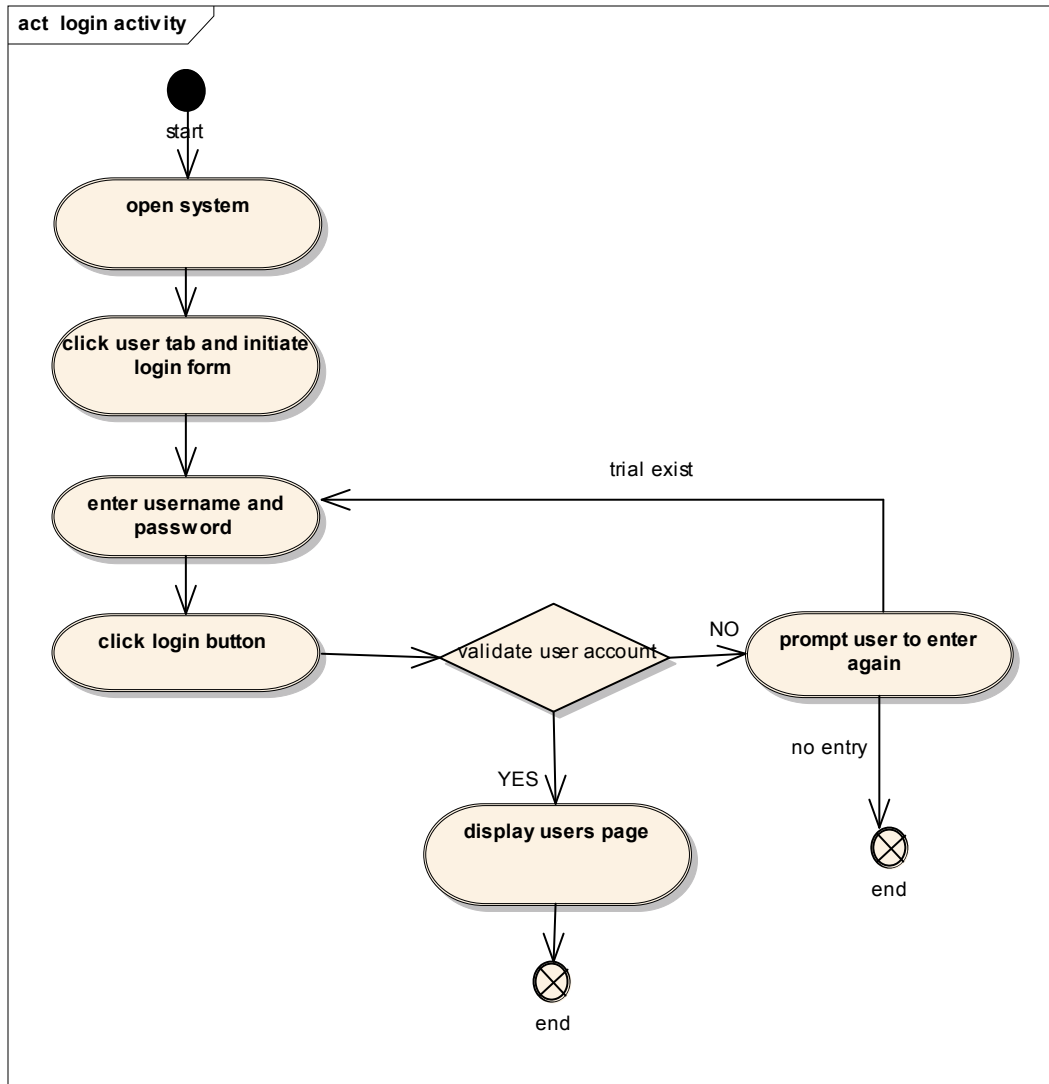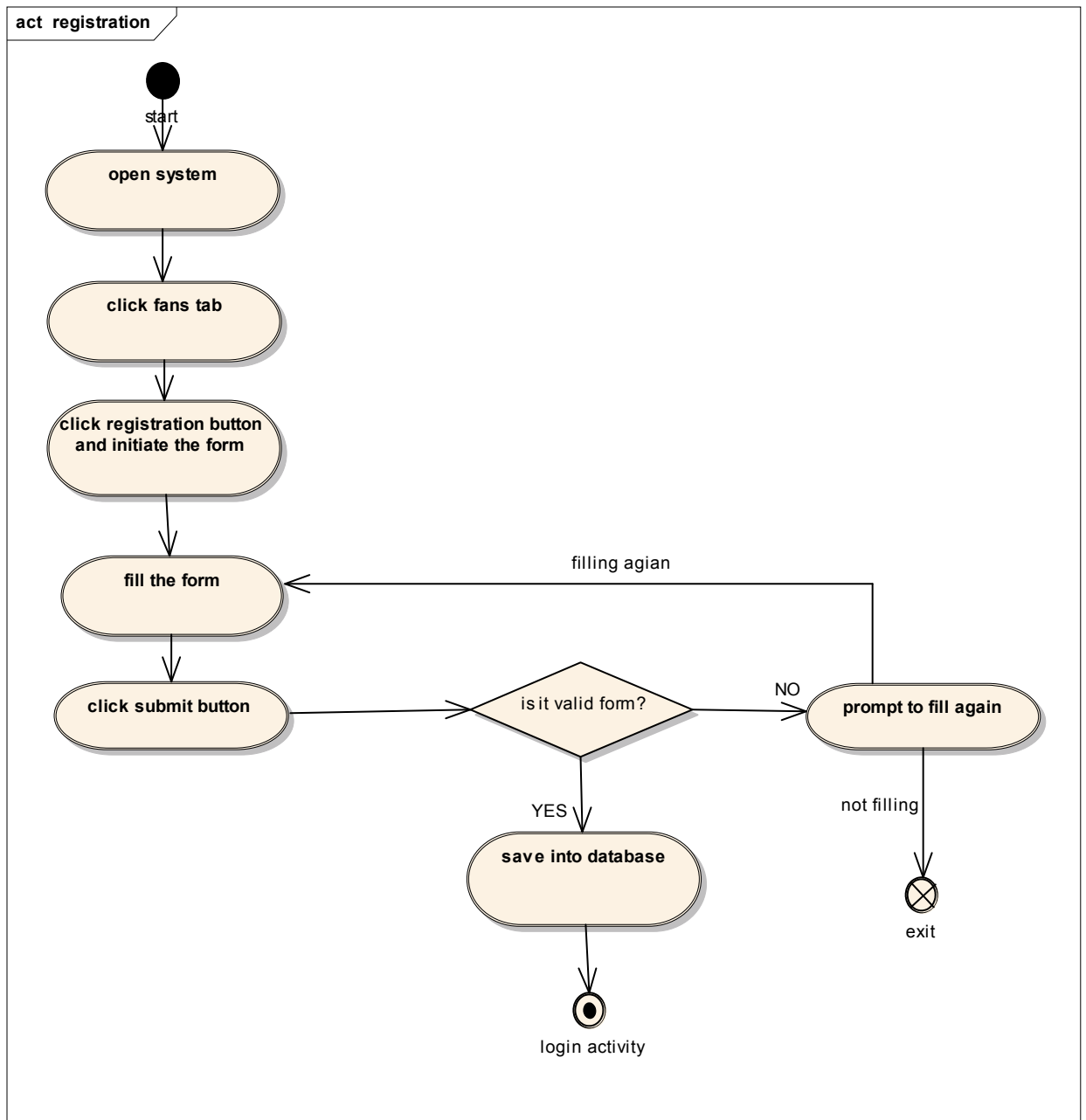— log out() → coach

7. Manage account
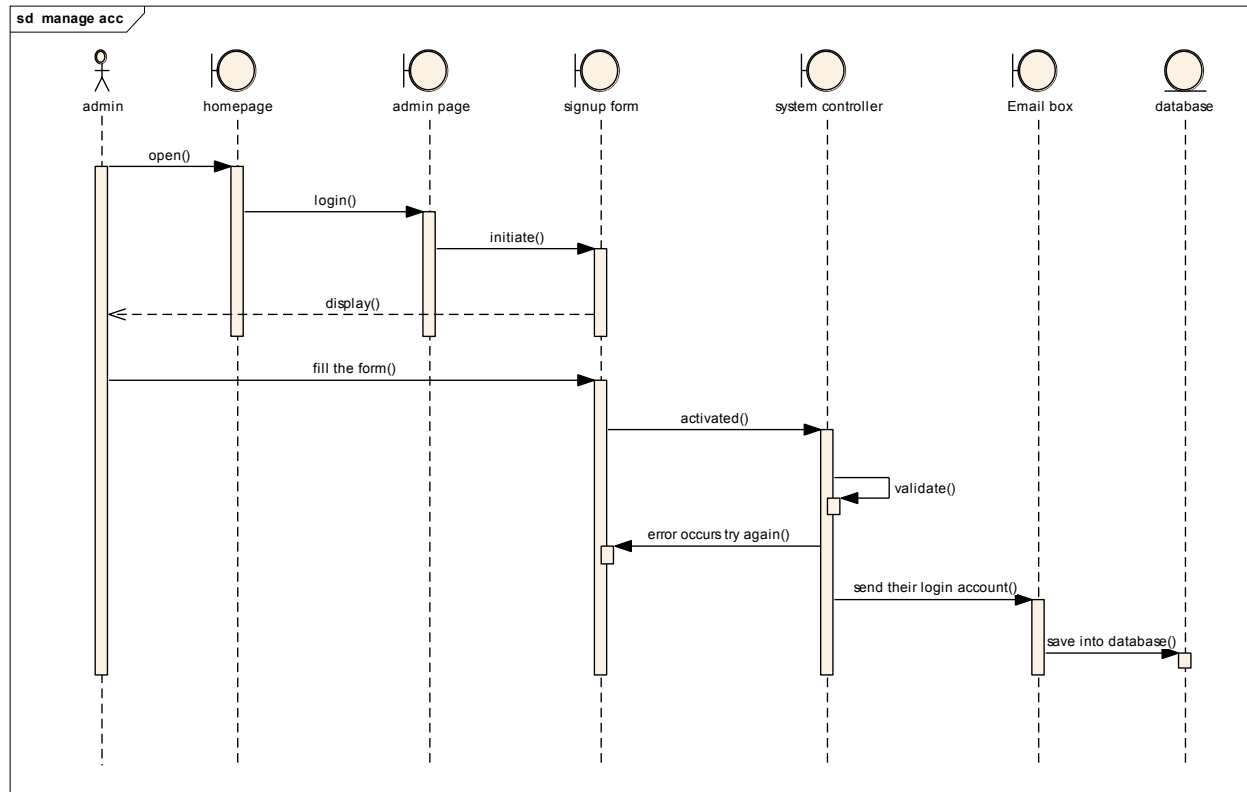
3.6.2 Activity diagram

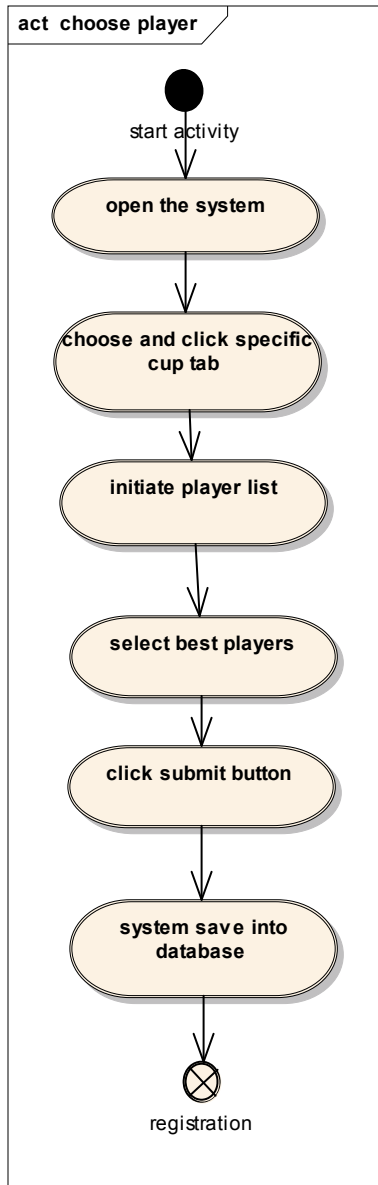Activity diagram depict the sequence of these activities:

- Diagrams are abstract and describe process in general
- They model behavior independent of object
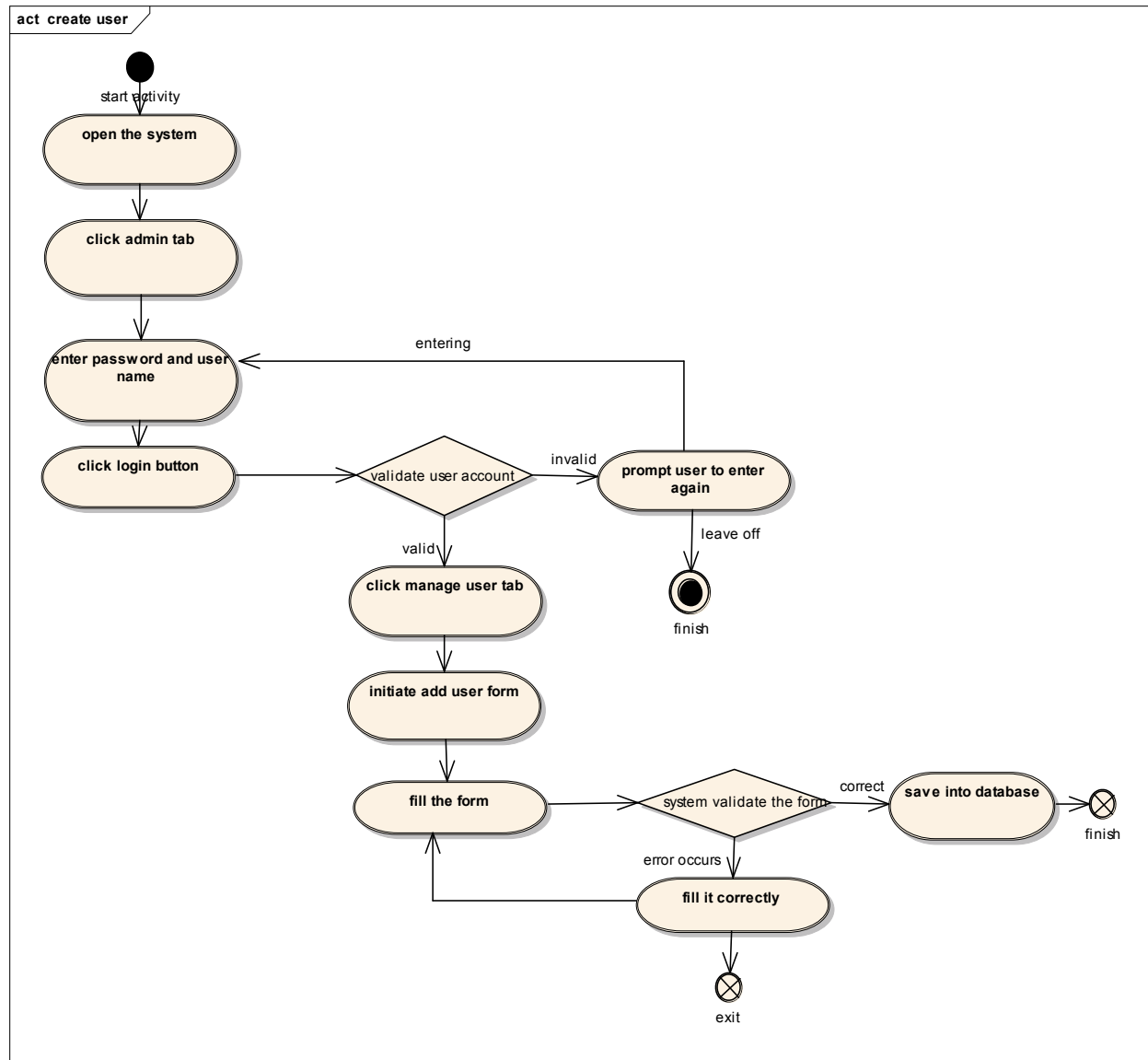- Can be used for any types of process
1. Login

act login activity
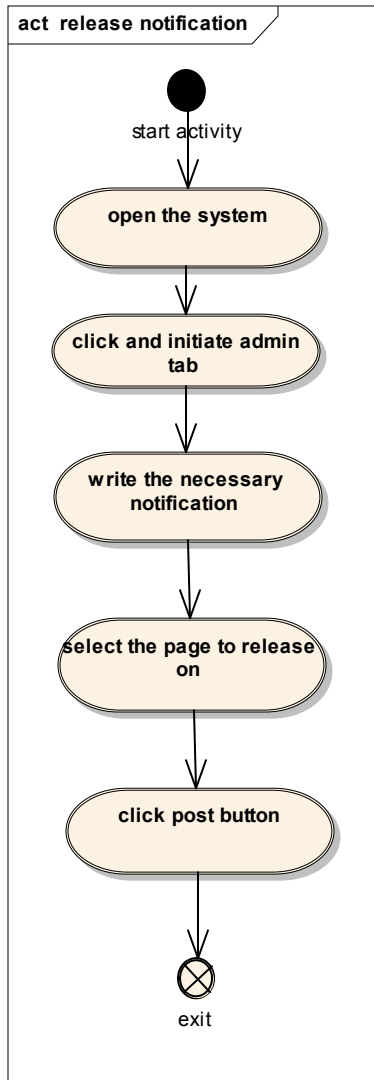
start

open system

click user tab and initiate login form

enter username and password

click login button

validate user account

trial exist

prompt user to enter again

NO

no entry

end

YES

display users page

end

2. Registration

act registration

start

open system

click fans tab

click registration button
and initiate the form

fill the form

filling agian

click submit button

is it valid form?

NO

prompt to fill again

YES

save into database

not filling

login activity

exit

3. Select players

act choose player

```
          ● start activity

    ┌─────────────────────┐
    │   open the system   │
    └─────────────────────┘
              │
    ┌─────────────────────┐
    │ choose and click    │
    │ specific cup tab    │
    └─────────────────────┘
              │
    ┌─────────────────────┐
    │ initiate player list│
    └─────────────────────┘
              │
    ┌─────────────────────┐
    │ select best players │
    └─────────────────────┘
              │
    ┌─────────────────────┐
    │ click submit button │
    └─────────────────────┘
              │
    ┌─────────────────────┐
    │ system save into    │
    │ database            │
    └─────────────────────┘
              │
            ⊗ registration
```

4. Create user

act create user

- start activity
- open the system
- click admin tab
- enter password and user name
- click login button
- validate user account
  - invalid → prompt user to enter again
    - entering
    - leave off → finish
  - valid → click manage user tab
    - initiate add user form
    - fill the form
    - system validate the form
      - correct → save into database → finish
      - error occurs → fill it correctly → exit

5.release notification

act release notification

start activity

open the system

click and initiate admin tab

write the necessary notification

select the page to release on

click post button

exit

6. See notification

act veiw notification

start activity

open the system

initiate user tab

see notification

exit

7. Manage account

3.6.3 State chart diagram

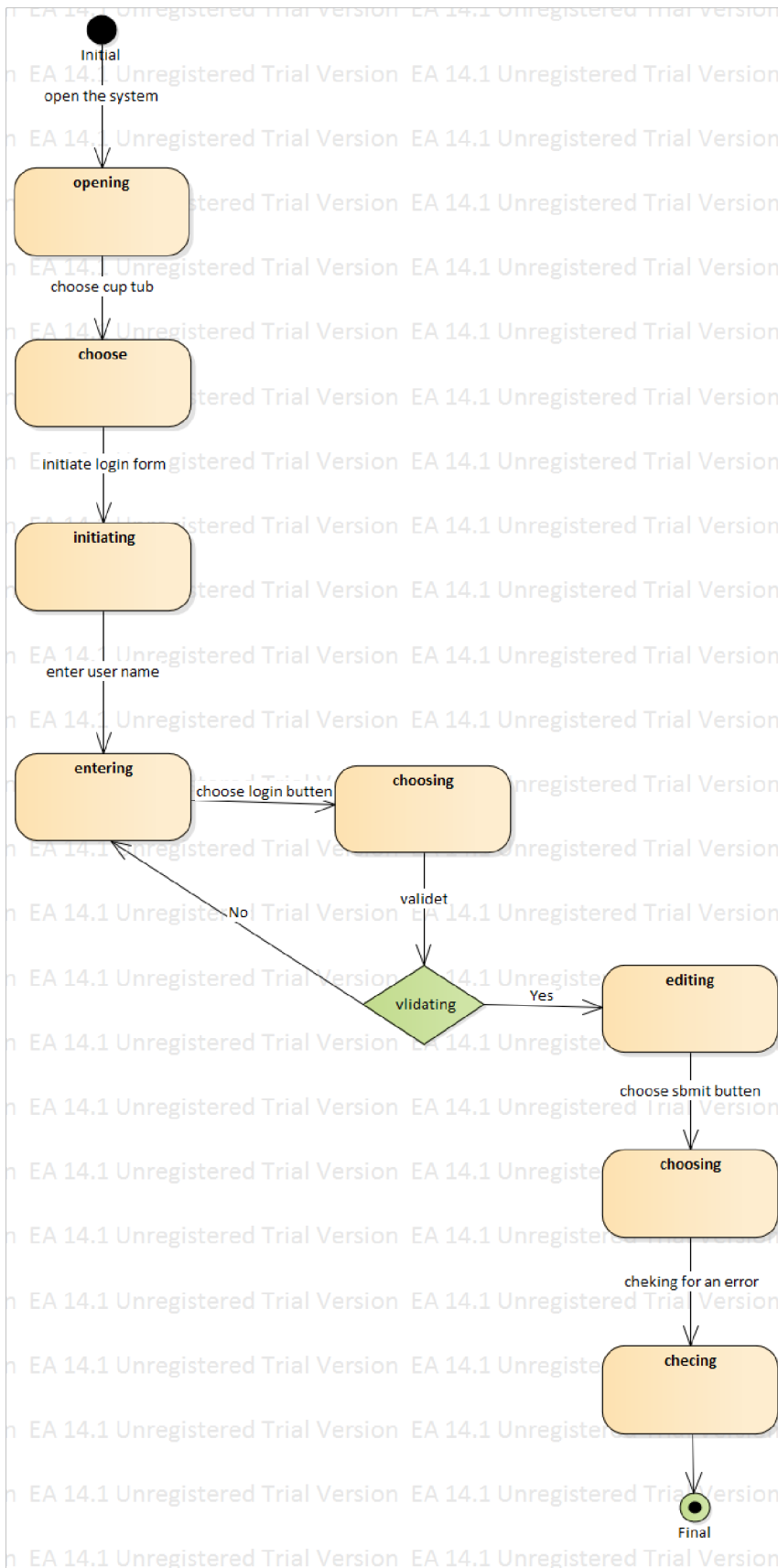State diagram determines how class process events based on their current state.

Event is any internal or external trigger which causes the system to change its state.

Transition is change in state. It starts from source to its destination

registration

Initial

Start system

Starting

Open System

Opening

Initiate form

Choosing

initiate form

Initiating

fill form

Filling   submit form   **Submitting**

check whether form is right

No

Checking   **Save**

Final

Initial

open the system

opening

choose cup tub

choosing

enter username and password

entering — choose login button → choosing

is the login account is valid

validating

No

Yes

initiating

open player list

opening

select player

selecting

save

saveing

Final

Chapter four

Software design

4.1. Introduction

4.1.1. Overview

This is the System Design document is for the ASTU football cup management system. This document includes the design goals, the proposed system Design and the object design. In the next sub-chapter we will see software arch, proposed software arch, hardware software mapping, subsystem decomposition, persistent data management, access control, global software control and boundary condition.

4.1.2. Purpose of the system

The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

4.1.3 Design goal

The design goals are determined mainly based on the nonfunctional requirements. Design goals describe the important system qualities.

- ➢ Performance
- ➢ Dependability
- ➢ Maintenance
- ➢ End user

4.1.4 Performance

Response time: we use simple algorithm with low running and compiling time so that the response time will be less.

Throughput:

4.1.5 Dependability

Robustness: our system let users to interact mostly by providing choices of inputs through check boxes, drop down which makes the system able to survive invalid user input.

Availability: the system is available anywhere with internet connection.

4.1.6 Maintainability

Extensibility: we try to decompose into distinct subsystems as much as possible that they can function independently this makes the system open to accept new functionalities.

Modifiability: as we mentioned above subsystems are loosely coupled so that we can change functionalities without affecting the remaining functionalities of the system.

Readability: we can easily understand the system by reading the code because its methods are clear and commented well.

4.1.7 End User Criteria

Usability: most of the functions are easy to use because they are not complicated and they easily understandable, users get them by using few numbers of buttons. Anyone can use it by following the simple system each step is clear and the steps themselves gives hint how to proceed to the next action.

4.2 Current software architecture

We used four basic layer in this architecture the first one is client side layer which is represented by computer used by user to enter query and receive respond from server. The second layer is server side which accept request from client and give respond only when there is third part which is internet connection between them. Then internet service portal customize website and aggregate information from wide array in a consistent manner and provide necessary information to the server in order to give respond for the request. The fourth layer is database layer to access data from the database to perform operations
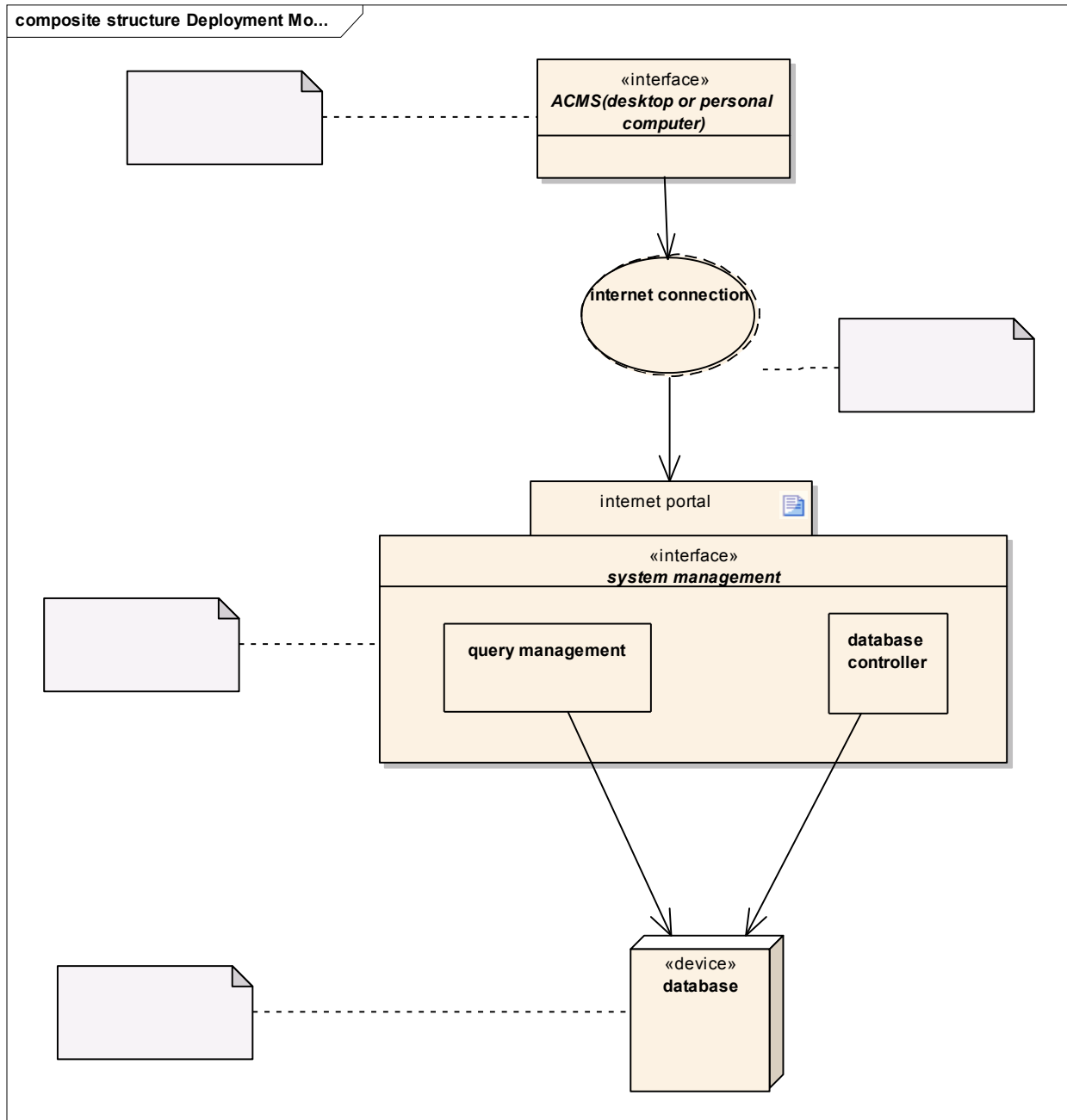
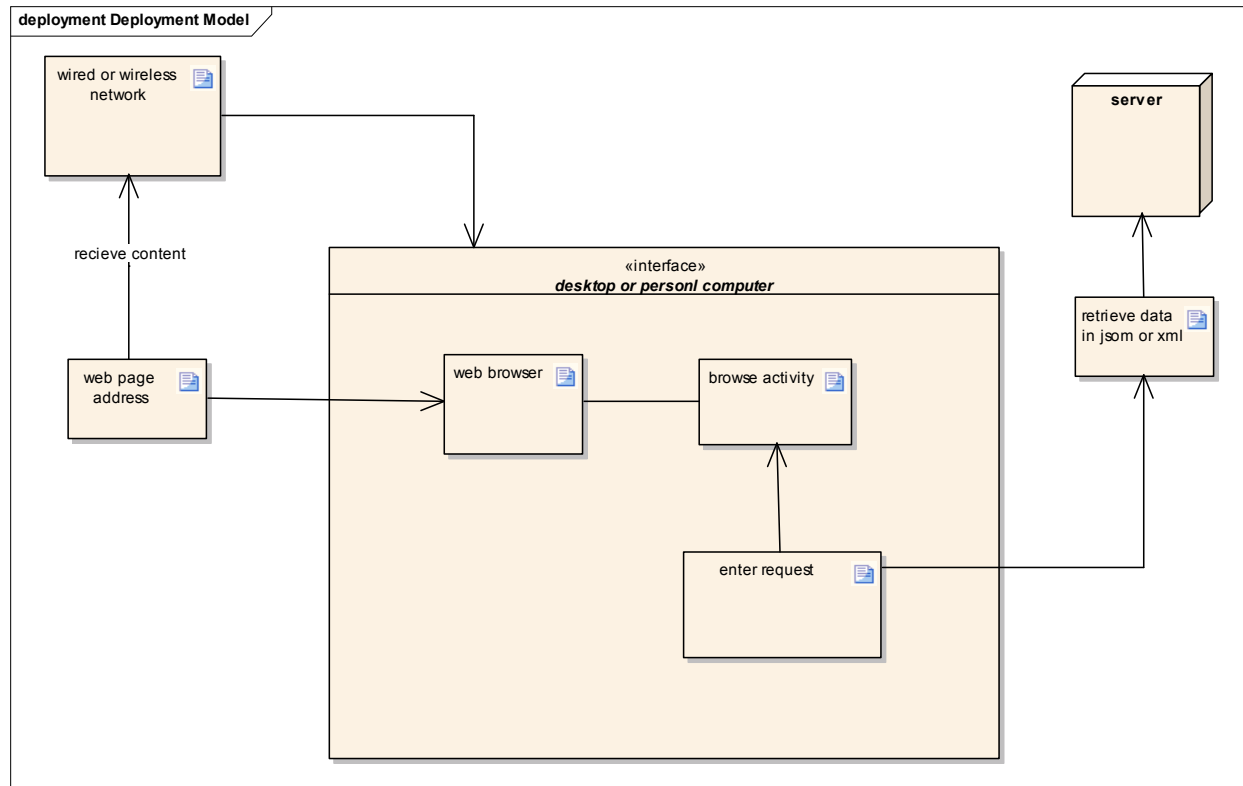*Figure* **4.2.1 True system architecture**

Figure **4.2.2 general system architecture**

4.3 Proposed system

4.3.1 Overview

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

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

4.3.2 Subsystem decomposition

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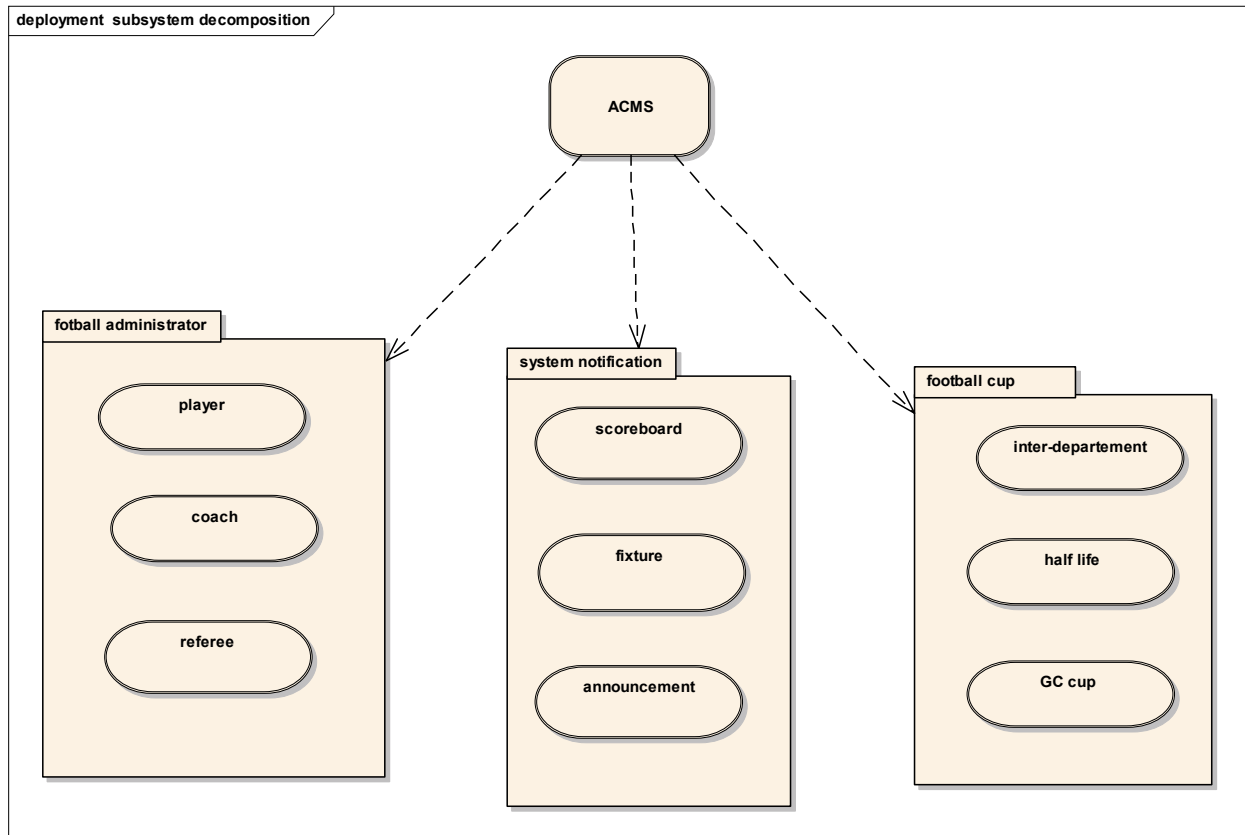

Figure **4.3.1 first level subsystem diagram**

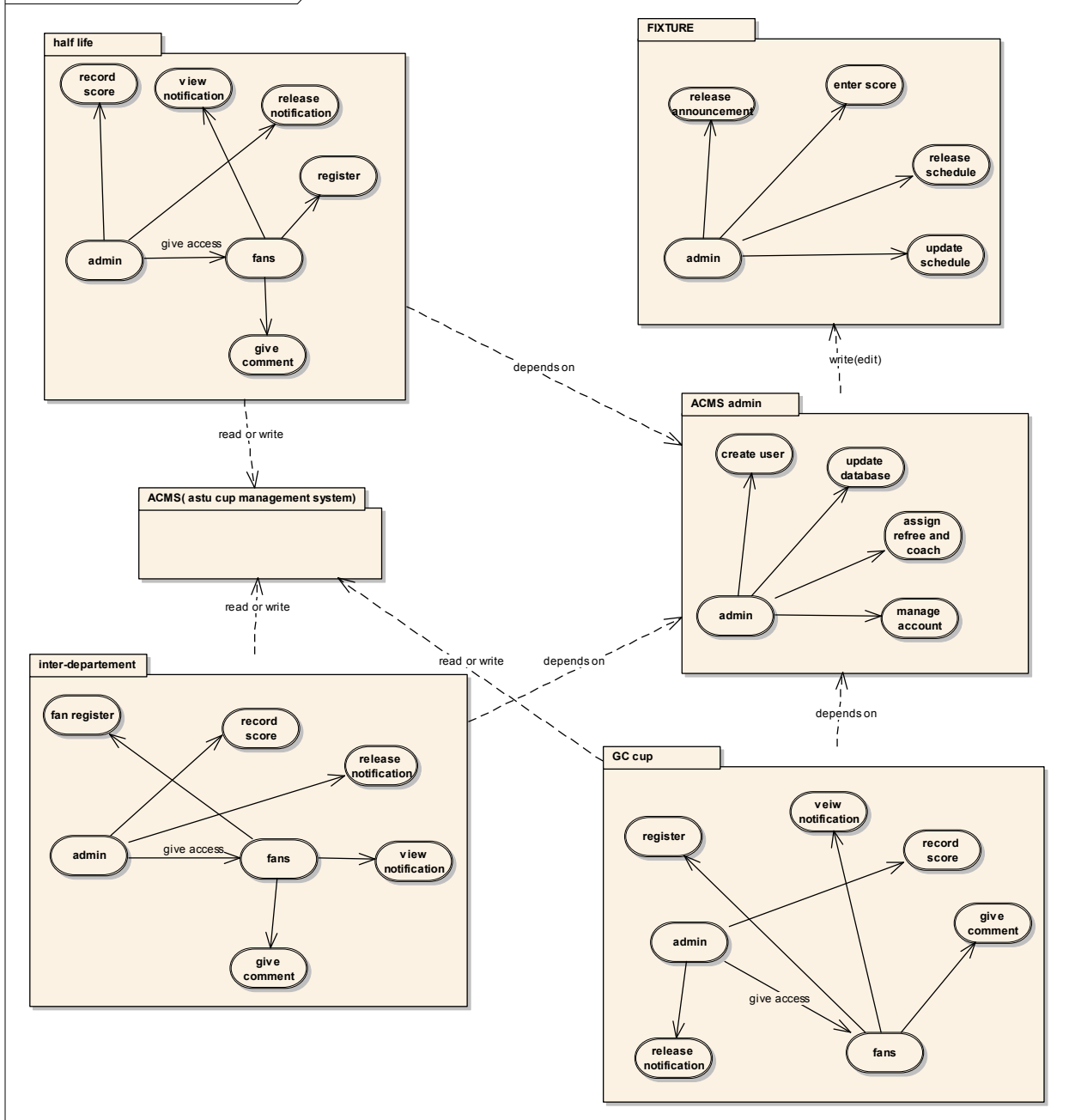*Figure* **4.3.2 second level subsystem diagram**

4.3.3 Subsystem description

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

Table 1. Subsystem description

4.3.3 Hardware software mapping

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

Where us the web server process is responsible to host the web pages of the system and process clients' request. In case of the client side, only a browser is required to access the objects.
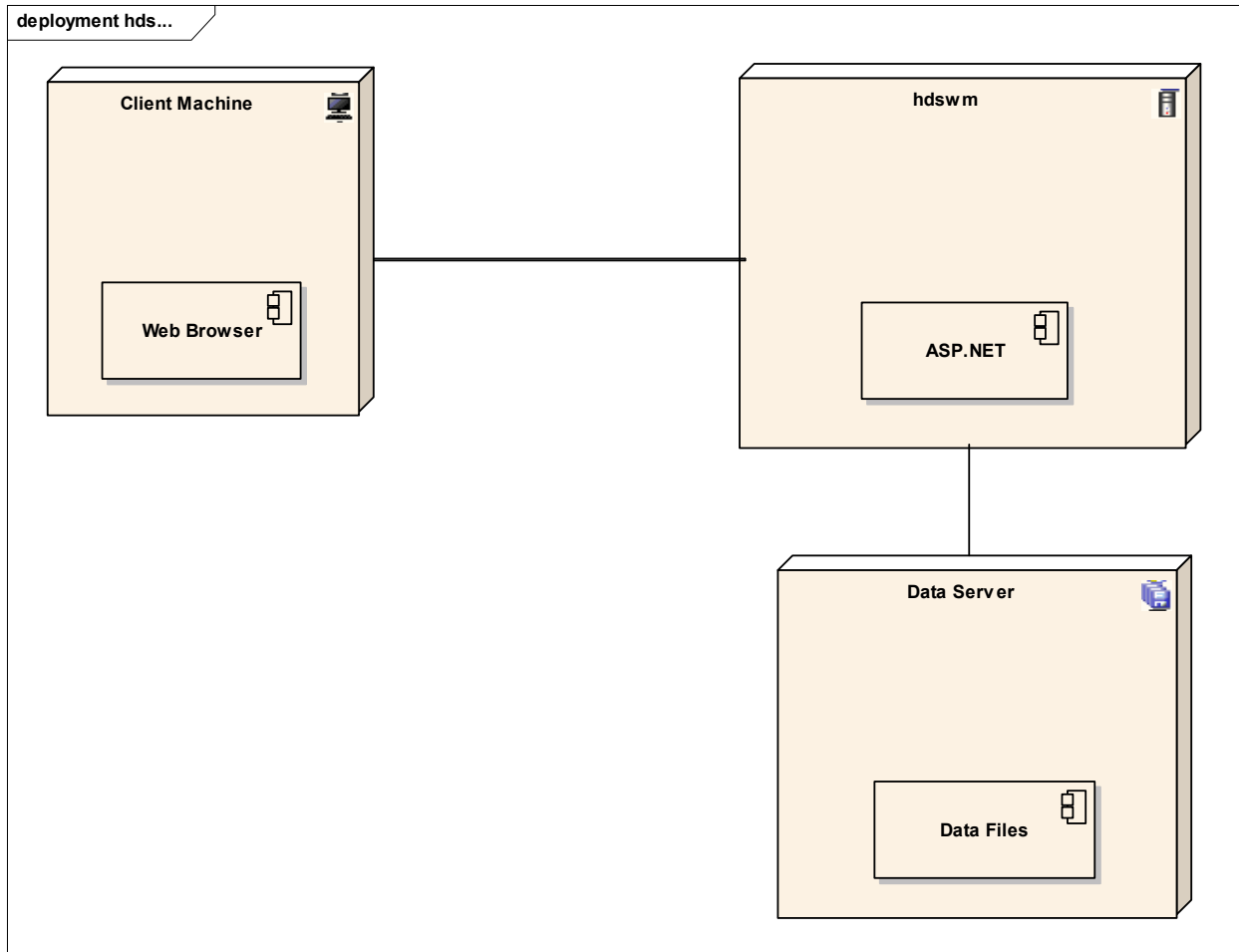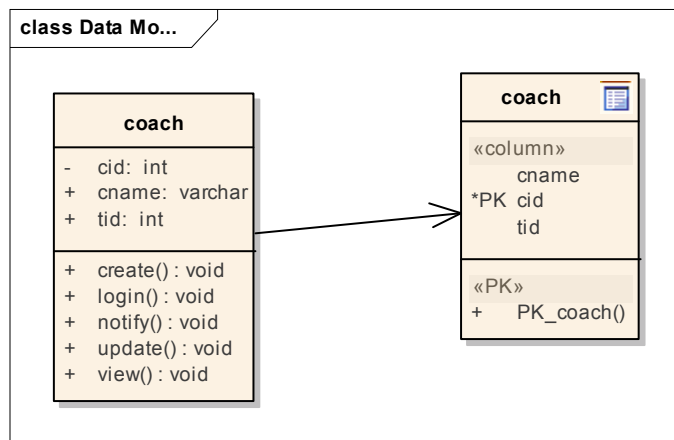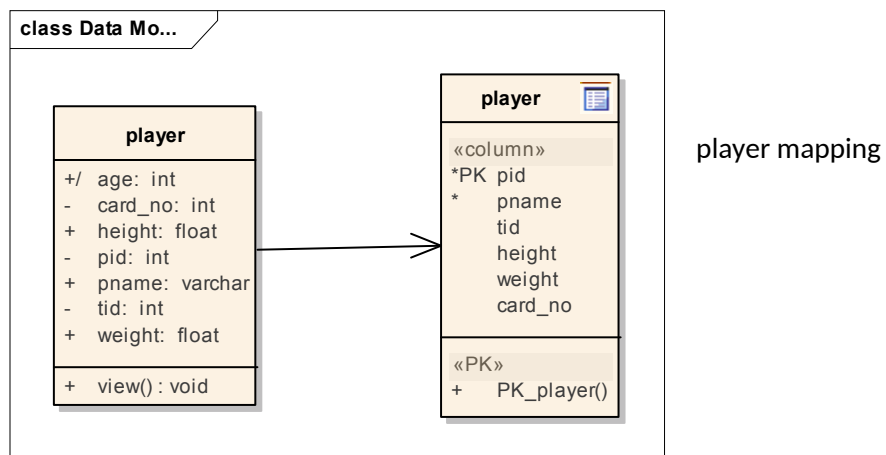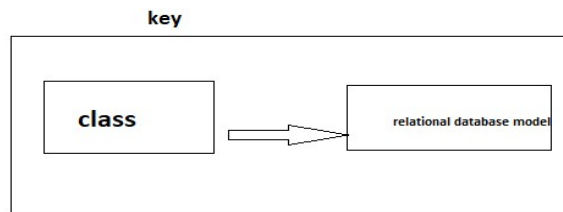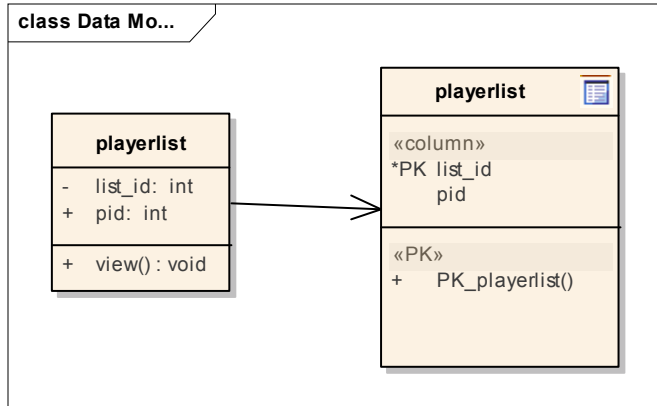
Fig hardware s
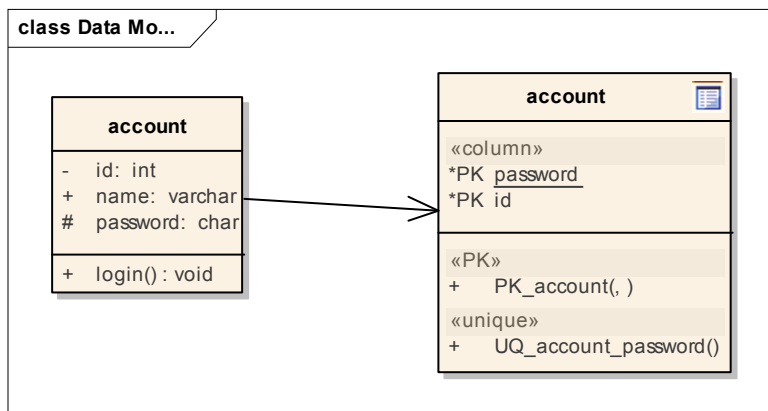
4.3.4 Persistence data management

The purpose of this section is to show the mapping of the objects/classes of the system, identified during the analysis stage, in to the corresponding relational database.
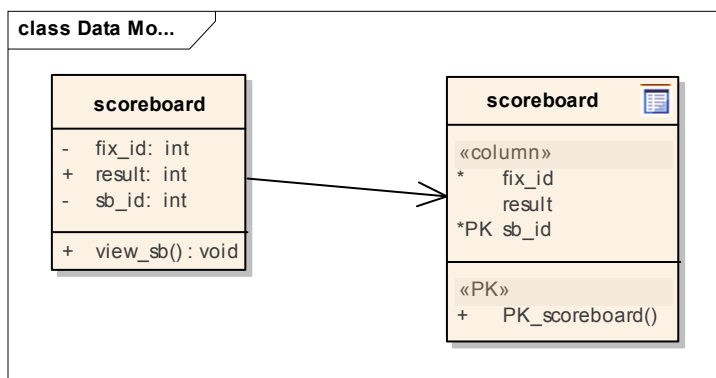
key

class	→	relational database model

class Data Mo...

player

+/ age: int
- card_no: int
+ height: float
- pid: int
+ pname: varchar
- tid: int
+ weight: float

+ view() : void

player

«column»
*PK pid
* pname
tid
height
weight
card_no

«PK»
+ PK_player()

player mapping

class Data Mo...

coach

- cid: int
+ cname: varchar
+ tid: int

+ create() : void
+ login() : void
+ notify() : void
+ update() : void
+ view() : void

coach

«column»
cname
*PK cid
tid

«PK»
+ PK_coach()

coach mapping

class Data Mo...

playerlist

- list_id: int
+ pid: int
+ view() : void

playerlist

«column»
*PK list_id
    pid

«PK»
+   PK_playerlist()

playerlist mapping

class Data Mo...

account

- id: int
+ name: varchar
# password: char
+ login() : void

account

«column»
*PK password
*PK id

«PK»
+   PK_account(,)
«unique»
+   UQ_account_password()

account mapping

class Data Mo...

scoreboard

- fix_id: int
+ result: int
- sb_id: int
+ view_sb() : void

scoreboard

«column»
*   fix_id
     result
*PK sb_id

«PK»
+   PK_scoreboard()
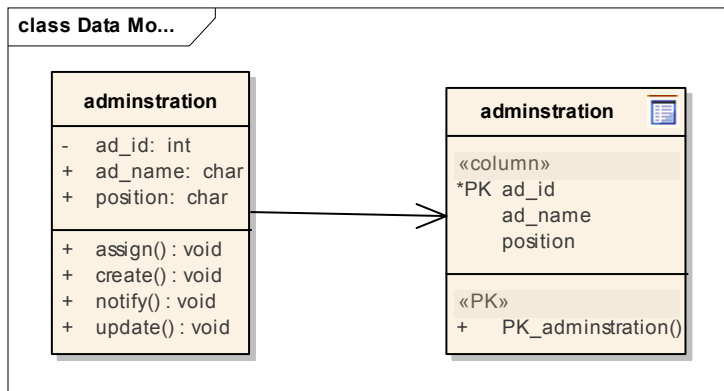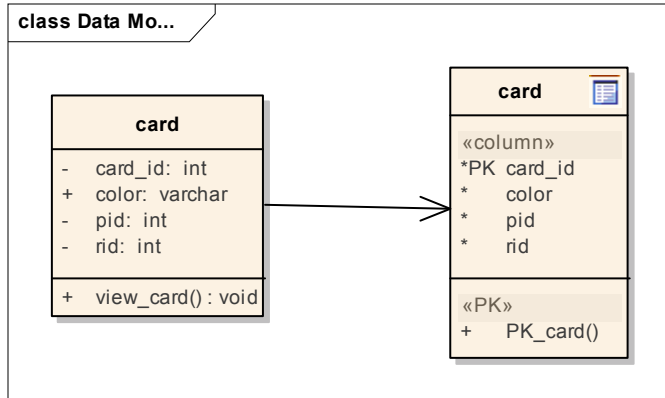
scoreboard mapping

fixtureboard mapping

adminstration mapping

class Data Mo...

card

-	card_id: int
+	color: varchar
-	pid: int
-	rid: int
+	view_card() : void

card

«column»
*PK card_id
*   color
*   pid
*   rid

«PK»
+   PK_card()

card mapping

class Data Mo...

refree

-	fix_id: int
-	rid: int
+	rname: varchar
+	create_card() : void
+	update_card() : void
+	view_card() : void
+	view_fix() : void

referee

«column»
*   fix_id
*PK rid
*   r_name

«PK»
+   PK_referee()

referee mapping

class Data Mo...

team

-	list_id: int
-	tid: int
+	tname: varchar
+	register() : void

team

«column»
*PK tid
       tname
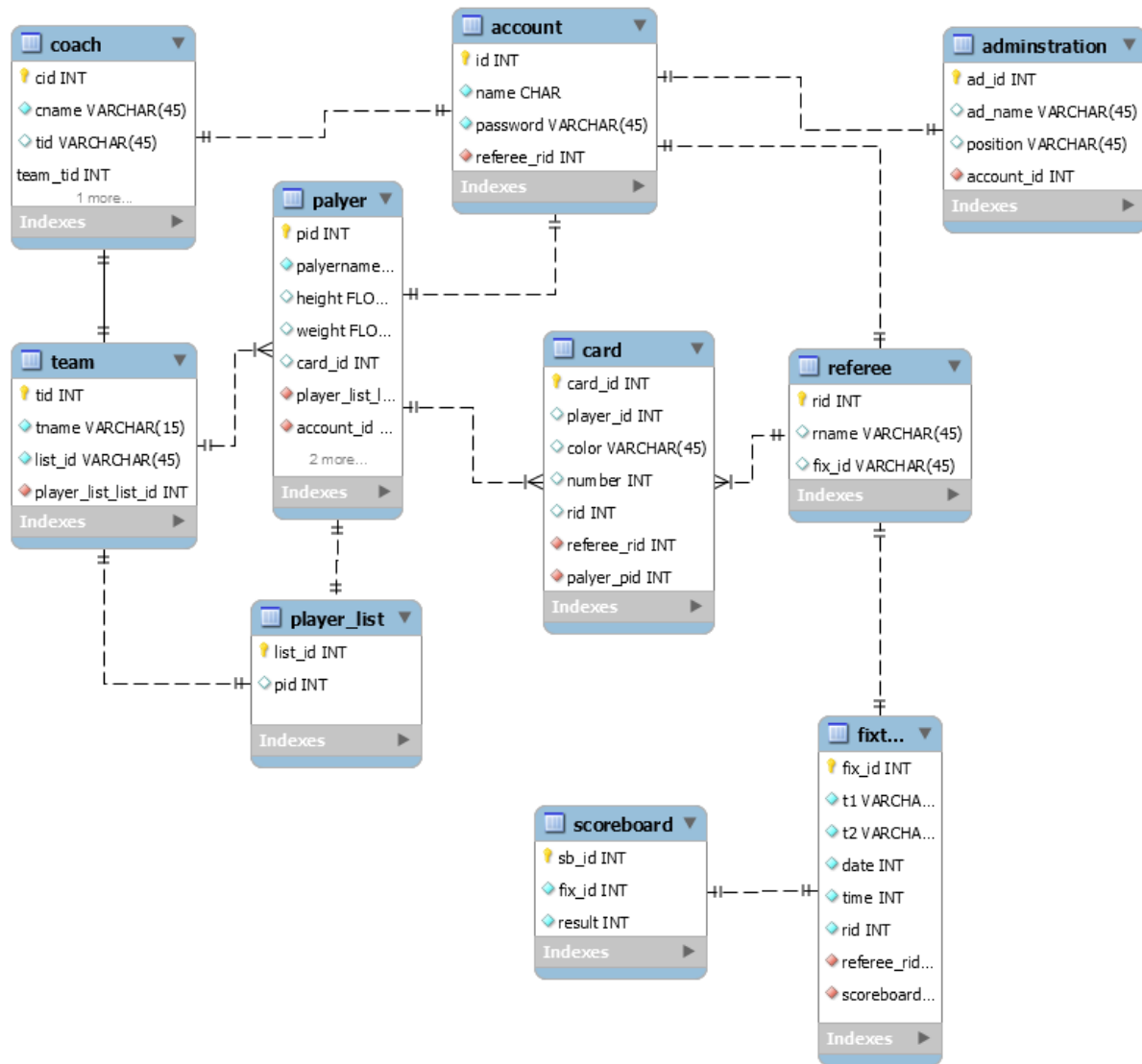   list_id

«PK»
+   PK_team()
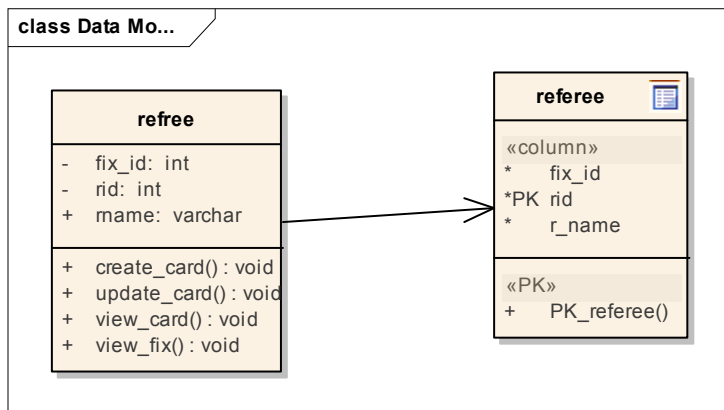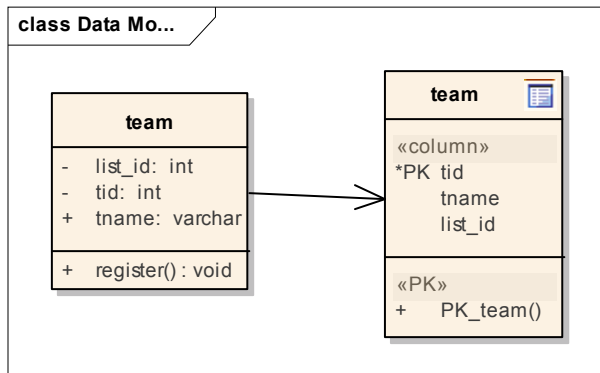
4.3.5 Database design

oftware mapping of the system

4.3.6 Component diagram

The purpose of a component diagram is to show the relationship between different components in a system. The term "component" refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system. There exists a whole development approach that revolves around components: component-

based development (CBD). In this approach, component diagrams allow the planner to identify the different components so the whole system does what it's supposed to do.
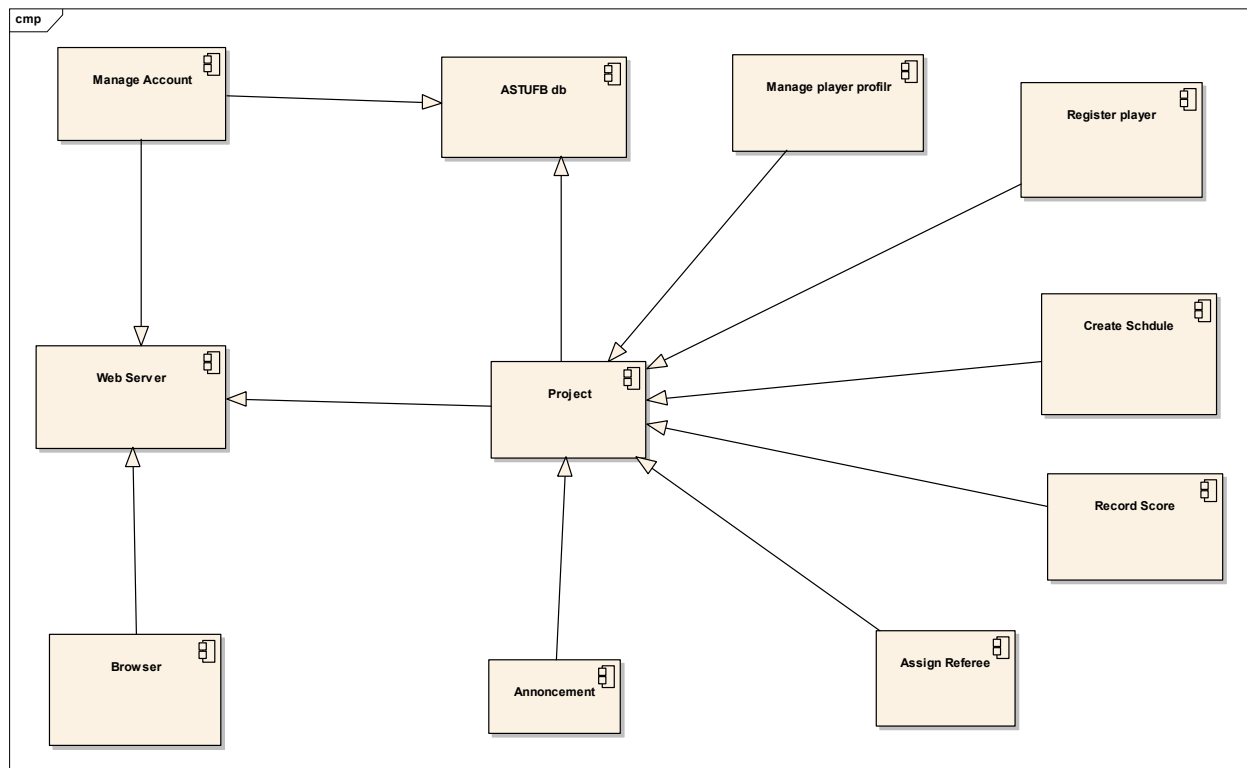

Fig component diagram

4.3.7 Component Description

Authentication controller

Description: This is a class responsible to authenticate a user

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

Page controller

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

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

PID generator

Description: the base class which is used during player registration

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

Data organizer

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

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

Component controller

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

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

Form controller

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

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

Profile model

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

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

Component model

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

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

Score model

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

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

Message model

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

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

4.3.8 Deployment diagram

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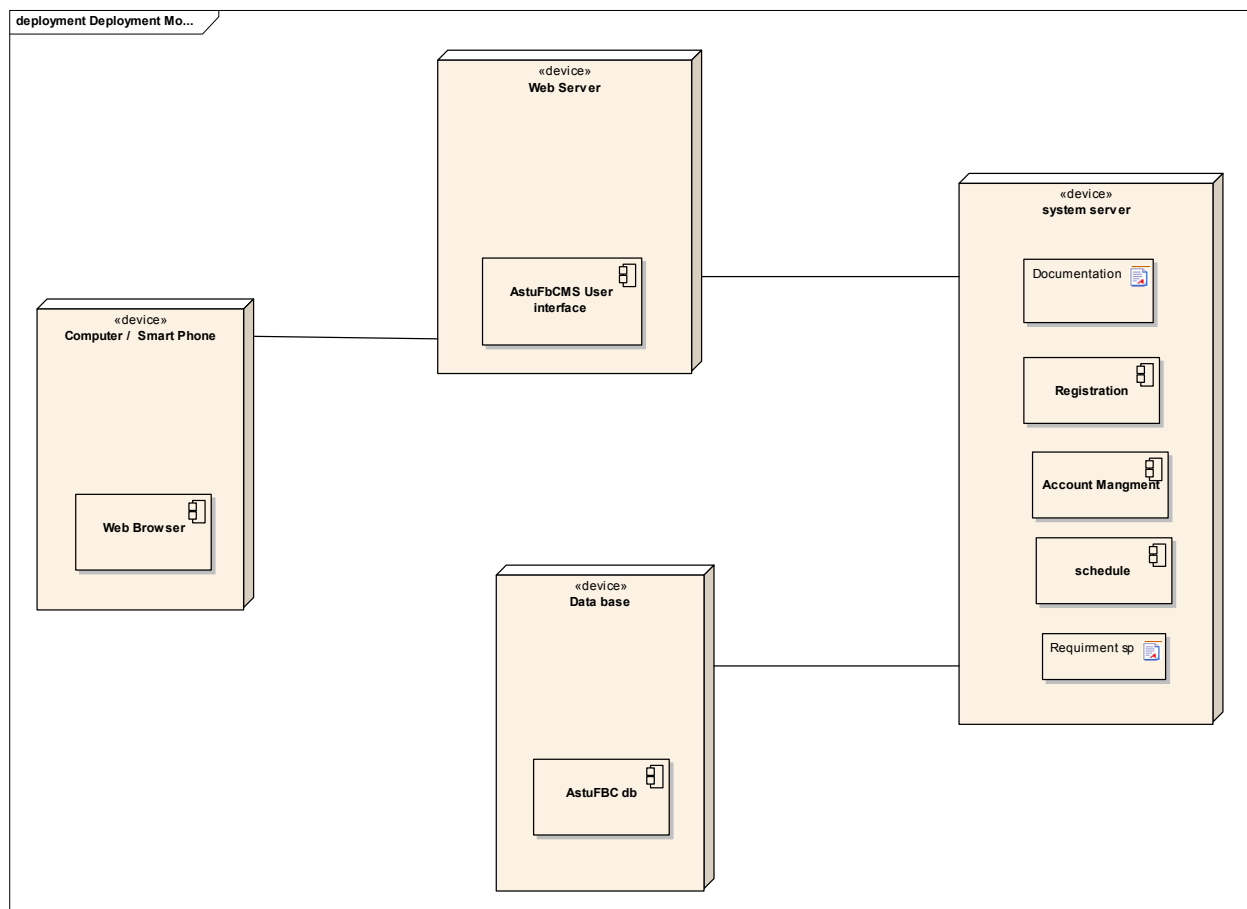

Fig deployment diagram

4.3.9 Boundary condition

System designs must be able to direct initiation and finalizations of a system. This can be done by administration use cases.

Initialization

In this process a system is brought from a non initialized state to a state also refered to as startup use cases. In order to achieve this, a web browser, internet connection and platform are concerned.

Dynamic Model of the System Startup

Dynamic model of system identification system is a system that is mandatory and always have to be turned on and can be used to obtain information about a football games of ASTU. While all these is carried on a steady internet connection is also necessary.
A web server that runs web App subsystem and webApp must be started so that a user is able to enter to the system from web.

Termination

This identifies the resources which are to be cleaned and the systems notified upon the process.
The proper start up of a system is necessary or problems may occur. For example if web server does not start up web interface may not be visible for the user. Once they are done interacting with the system , users can log out and leave web

Failures

failure maybe caused by- Bad internet connection, bugs, errors and power supply and others

boundary condition failure

Error	Cause
Fatal failure	eBay spyware
Server Not found	Internet connection fail

communication link fails	IP address or hostname in JDBC URL is wrong. Hostname in JDBC URL is not recognized by local DNS server. Port number is missing or wrong in JDBC URL. DB server is down. DB server doesn't accept TCP/IP connections. Something in between ASP.Net blocking connections, e.g. a firewall or proxy
Web server fail	Computational/logic Errors Power outages Overheating
	High humidity Natural disasters, e.g., hurricanes, Floods
Web Application fail	web application attacks (treat) cross-site scripting(XSS) and SQL injection Device driver failures I/O errors, e.g., hard disk failures(see database media failures) Memory parity errors Network hardware failures

4.3.10 Access control

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

Actors Objects	Fixture	Score Board	Player	Referee	Coach	Registration Form
Admin	View() Create()	View() Create()	View() Update()	View() Update()	View() Update()	View() Send()
Coach	View()	View()				
Referee	View()	View()				
Player	View()	View()	View() Update()			
Visitor	View()	View()				

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

References:

Ian Somerville, 9th edition.

Several YouTube lecture videos.