EGERTON UNIVERSITY

PROJECT SOFTWARE REQUIREMENTS SPECIFICATION

FOR

TITLE: GAMES MANAGAMENT SYSTEM

PREPARED BY: STUDENTS NAME:	REG. NO.:
DAVID OPIKO	S13/20794/13
JACKSON KING'ORA	S13/21372/14
PAPA JEREMY	S13/21417/14
HUMPHREY ADALA	S13/21423/14
MIKE OMAMO	S13/21412/14

PROJECT CORDINATOR: MR. ODIYO

DATES: 01-08-2018
VERSION: 002

TABLE OF CONTENTS:

1. INTRODUCTION

1.1 Purpose

This document describes the requirements that will be used by the developer to design and implement the proposed GAMES MANAGEMENT SYSTEM. The proposed system is a web-based management system for the university to manage all its gaming activities.

1.2 Document Conventions

Main Section Titles:

- Font: Times new Roman (Body)
- Face: Bold
- Size: 12
- Underline: True

Sub Section Title

- Font: Times new Roman (Body)
- Face: Bold
- Size: 12
- Underline: True

Other Text Explanations

- Font: Times new Roman (Body)
- Face: Normal
- Size: 12

1.3 Intended Audience and Reading Suggestions

i. Clients: The users of the system will get a clear idea of the software and hardware requirements that are needed.
ii. Developers: The project developers will be able to quickly understand the methodology enabled and personalize the system.
iii. Students and games patrons: They will get to know the various requirements that are needed for the development of this system.
iv. Supervisors – Critic the system based on requirements.

The author suggests that clients go through the requirements section thoroughly before installing the software.

The developers can use this document as a resource in developing the system.

1.4 Project Scope

The system brings three modules together:

i. coordinator module - Registers every sport in the university, assigns every game with a patron, approves funds requested for by patrons, communicates directly with patrons, approves every sport events and activities and generate reports.
ii. Games patron module – Requests for funds, approves new students who wishes to join the sport, informs the coordinator about upcoming sporting events and activities, communicate directly with both the coordinator and students in his or her sport.
iii. Students module – View upcoming events and activities, check whether remunerations have been disbursed, communicate directly with the patron.
iv. Store clerk module – Manage inputs and outputs to the game department store including their source and current receptacle of a particular store entity.

2. Overall Description

2.1 Product Perspective

This is a new web based system that will replace the current manual system in the Games department.

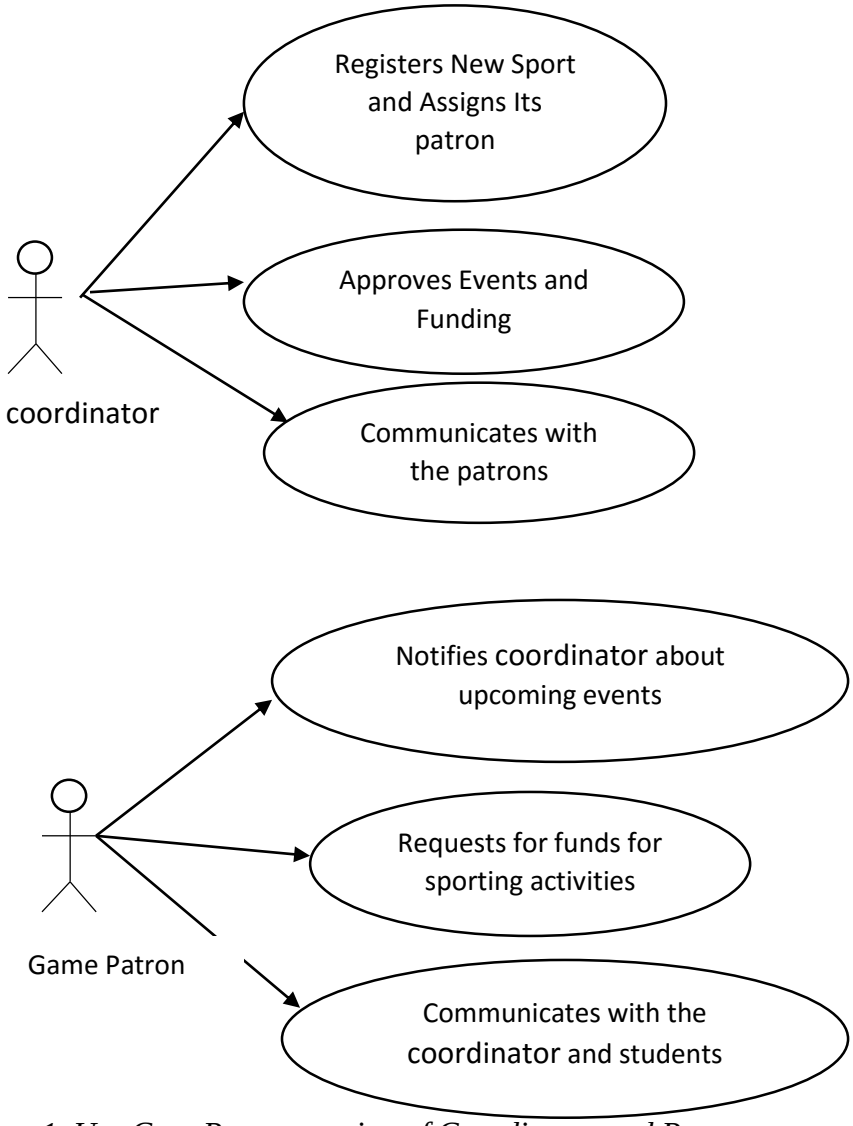


Figure 1. Use Case Representation of Coordinator and Patron as a product

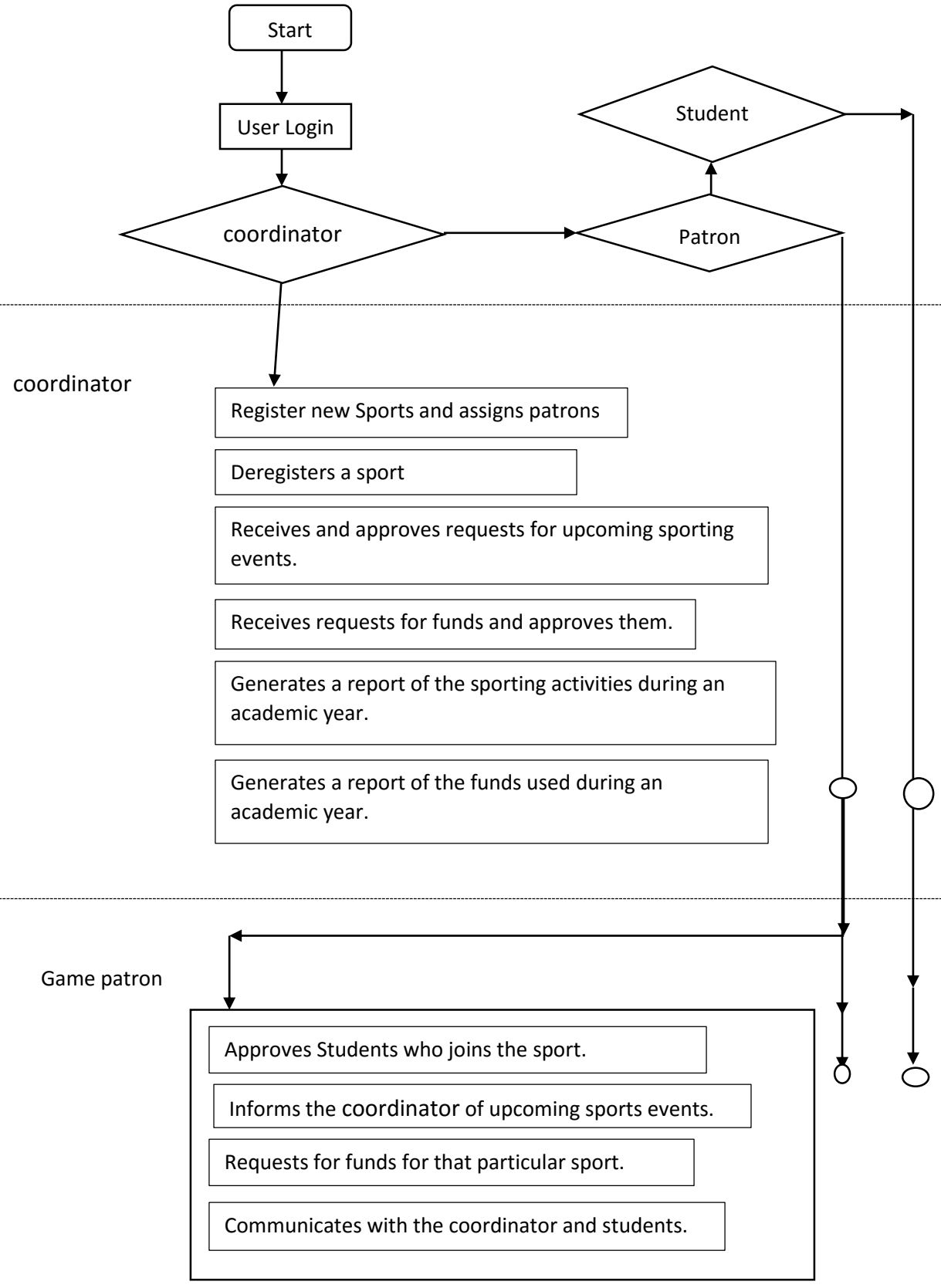
2.2 Product Features

i. coordinator module - Registers every sport in the university, assigns every game with a patron, approves funds requested for by patrons, communicates directly with patrons, approves every sport events and activities and generate reports.

ii. Games patron module – Requests for funds, approves new students who wishes to join the sport, informs the coordinator about upcoming sporting events and activities, communicate directly with both the coordinator and students in his or her sport.

iii. Students module – View upcoming events and activities, check whether remunerations have been disbursed, communicate directly with the patron.

Flow Diagram



2.3 User Problem Statement

The users of this system are expected to be familiar with computer hardware and software systems. This will be crucial in using this system.

2.4 User Objectives

i. The system is expected to be a computerized online system.
ii. It should provide a login so as to give access to authorized users.
iii. The coordinator should be able register new sports and assign patrons.
iv. The system should provide a means for the various user modules to communicate directly such as the coordinator and patrons.
v. The coordinator should be able to generate various reports such as financial reports and sporting events reports for a particular academic year.
vi. The coordinator should be able to approve sporting events and funds once he or she receives them.
vii. Patrons should be able to approve every student that requests to join the sport.
viii. Store keeper keeps records about the store safely and up to date.

2.5 Operating Environment

The expected system is an online web-based application system. Thus, it can run on any Operating System platform such as Linux, Windows or Mac.

Using a web browser such as Google Chrome or Mozilla Firefox, enables a user to access this system.

Internet connection should be available.

The system would require a web server to be hosted from it and a database such as MySQL to store information.

2.6 Design and Implementation Constraints

Each user must keep their passwords confidential and they must have a username which must be unique to everyone.

Only the coordinator can register and deregister a sport. Furthermore, he or she can approve funds and sporting events.

Only the patron should approve students who wish to join a particular sport.

2.7 User Documentation

Once the product is designed and implemented, user manuals (softcopy and hardcopy) will be provided along with the system documentation.

2.8 Assumptions and Dependencies

There is only one Coordinator who can assign another person to be the assistant Coordinator.

A store keeper operates independently hence has unique code to access his/her login page.

The computers are installed with modern browsers.

Each user has a unique username and a password.

Internet connection is available.

2.9 User Constraints

The system should be user-friendly.

The User Interface should be appealing to the eye.

The system should take at most 5 seconds to accomplish a particular task.

The system should be able to run on a computer with at least 512Mb of R.A.M.

The system should adjust accordingly if subjected to a new platform.

3. System Features

3.1 Authentication

3.1.1 Description and Priority

The system is accessed via Internet connection at the client level by validating the user with the unique username and password.

Priority: 9 of 9.

3.1.2 Stimulus/Response sequences

1. Access Games Portal
2. Users – Login
3. Coordinator – Registering and deregistering new sports, generating reports
4. Patron – Approving new students
5. Students – Checks on the registered sporting event
6. Store Keep – Record various resources inputs and outputs

3.1.3 Functional Requirements

The system identifies individual users before giving them access to the system.

3.2 Accountability

3.2.1 Description and Priority

Every user event that occurs in the system is logged making it easier to track every action that occurred in the system.

3.2.2 Stimulus/Response sequences

7. Coordinator – Registering and deregistering new sports, generating reports.
8. Patron – Approving new students.
9. Store Keeper – View store's reports

3.2.3 Functional Requirements

The system should identify users by their names once they log in. It should keep of the time events occurred and who triggered these events

4. External Interface Requirements

4.1 User Interface Requirements

- Login Screen: This is for all the system users to access the system. It requires a unique username and password.
- Account Details: Shows the account details of various users.
- New registration: For students to join a particular sport.
- New registration: For Store Clerks to manage the store system.
- Reports: This helps the coordinator to generate reports.
- User account: This enables the user to view the details of their own account.

4.2 Hardware interface

A computer connected with Internet to access the system.

4.3 Software Interface

The system is a multi-user system. It uses a programming code to access the Apache Web server and MySQL database. The system is installed on this server which is used to give access to the server.

Some of the data that transmitted through this platform are:

- Login details.
- Communications between the various user modules.

4.4 Communication Interface

The system uses HTTP for transmission of data from the user client side to the server side.

5. Other Non-Functional Requirements

5.1 Performance Requirements

Time constraint: Most tasks should be completed within a maximum of 5 seconds in order to improve performance. These tasks are: Reports generation and approval of various activities.

5.2 Safety Requirements

The data handled in this system is very vital. The server that hosts the system should always be confirmed to run properly and the data saved to the database at consecutive intervals. A backup server and database, apart from the primary ones, should be in place at a different location in order to prevent loss of data in the event of data loss.

The computers used in the Game department should be configured with Uninterrupted Power Supply to allow user to save their data in case of power outage.

5.3 Security Requirements

The system has a login to authenticate all the users who access the system.

During registration and deregistration of sports, and approval of funds and events, the Coordinator will be required to provide his or her login details.

5.4 Software quality attributes

Availability: The system should be available to the users 24 hours a day. This is achieved by making an online system.

Robustness: The system should be less susceptible to crashing even if the number of users at a particular time is very large.

Usability: The system should be easy to use without many significant challenges. The users of this system should also have great experience while using this system.

5.5 Use Cases

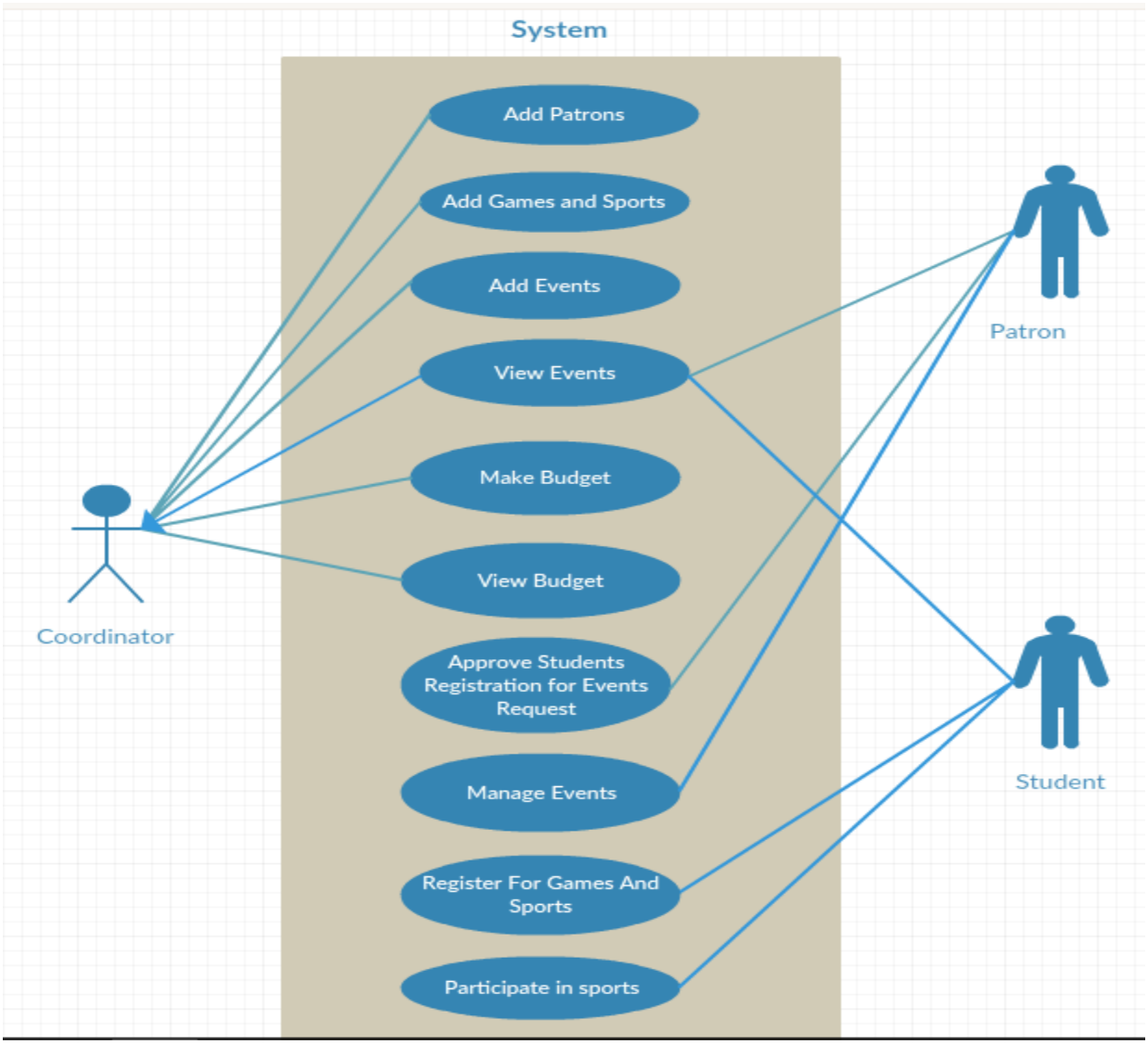


Figure 5: Use case for Coordinator, patron and students

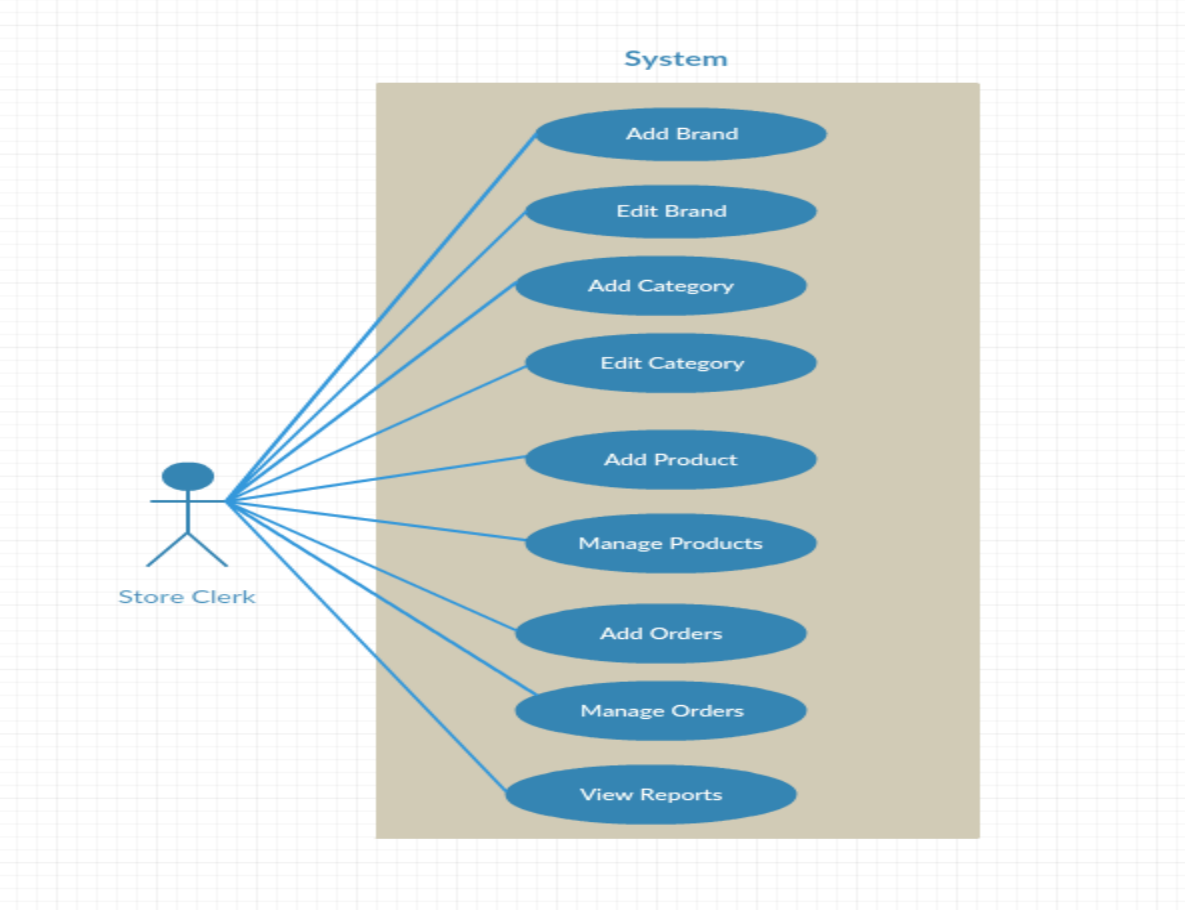
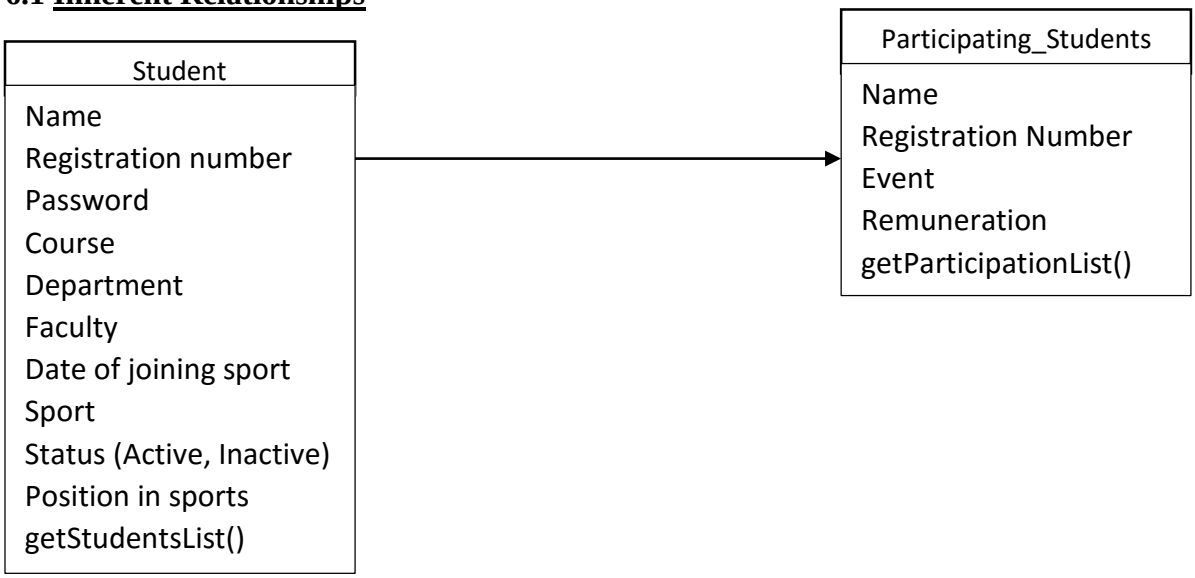


Figure 6: Use case for the store clerk

6 Preliminary Object-Oriented Domain Analysis

6.1 Inherent Relationships



6.2 User Classes and Characteristics

Student: This class provides the details of students who participate in the various sports.

Its attributes include:

- Name
- Registration number
- Password
- Course
- Department
- Faculty
- Date of joining sport
- Sport
- Status (Active, Inactive)
- Position in sports
- getStudentsList()

Parton

- Name
- Username
- Password
- Sport
- Date assigned
- Date unassigned
- Status (Active, Inactive)

Game

- Name
- Id
- Number of players
- Captain
- Patron name
- Patron id
- Day established

Coordinator

- Name
- Username
- Password
- Status (Active, Inactive)

Events

- Name
- Event type (Multiple games, single games)
- Date of event
- Date requested
- Person Requesting
- Expenditure
- Status (Approved, Disapproved, Pending)
- Venues (s)

Participating_Students

- Name
- Registration Number
- Event
- Remuneration
- getParticipationList()

Database_Connection

- ServerName
- User
- Password
- DatabaseName
- getConnection()

Communication

- Sender
- Recipient
- Message
- Subject
- Date_sent
- Status (read, unread)

StoreClerk

- Name
- Username
- Password
- Date of recordings
- Users

6.2.1 Abstract or Concrete

i. Student

ii. Events
iii. Coordinator
iv. Game
v. Patron
vi. Store Clerk
vii. Database_Connection

6.2.2 List of super classes

i. Student
ii. Events
iii. Coordinator
iv. Game
v. Patron
vi. StoreClerk
vii. Database_Connection

6.2.3 List of subclasses

- Participating_Students
- Communication

6.2.4 Purpose

Participating_Students class will help to get the list of students who will be playing during a particular event.

Coordinator, Patrons and Store_Keeper classes will help in setting user login sessions for the system to authenticate and give access to users.

6.2.5 Collaborations

Several classes will operate together to achieve some tasks.

Students, Coordinator, Patrons, Games and Participating_Students will collaborate with Database_Connection class in order to access the database.

Patron and Coordinator classes will collaborate with Participating_Students and Events classes in order to generate a list of students who will be taking part in an event.

The Coordinator class will collaborate with Events class in order to generate reports of finance and all the events that have taken place in a particular academic year.

Coordinator, Students and Patrons classes will work with Communications class to achieve communication between the various user modules.

6.2.6 **Attributes**

Student

- Name
    - Registration number
    - Password
    - Course
    - Department
    - Faculty
    - Date of joining sport
    - Sport
    - Status (Active, Inactive)
    - Position in sports
    - getStudentsList()

Parton

- Name
- Username
- Password
- Sport
- Date assigned
- Date unassigned
- Status (Active, Inactive)

Game

- Name
- Id
- Number of players
- Captain
- Patron name
- Patron id
- Day established

Coordinator

- Name
- Username
- Password

- Status (Active, Inactive)

Events

- Name
- Event type (Multiple games, single games)
- Date of event
- Date requested
- Person Requesting
- Expenditure
- Status (Approved, Disapproved, Pending)
- Venues (s)

Participating_Students

- Name
- Registration Number
- Event
- Remuneration
- getParticipationList()

Database_Connection

- ServerName
- User
- Password
- DatabaseName
- getConnection()

Communication

- Sender
- Recipient
- Message
- Subject
- Date_sent
- Status (read, unread)

StoreClerk

- Name
- Username
- Password
- Date of recordings
- Users

6.2.7 Operations

Database_Connection class: This will help all other classes access the database.

Participating_Students class: This will generate all the students who will be participating in a particular event together with their total numbers.

Events class: This will give the profile of every sport in the University. It will also be used to register and deregister a new sport.

Coordinator, Student, Patron and StoreClerk classes: This class will be used by the users for logging in to the system.

Communication class: This will be used by coordinator, Patron and Student classes to achieve communication among them.

6.2.8 Constraints

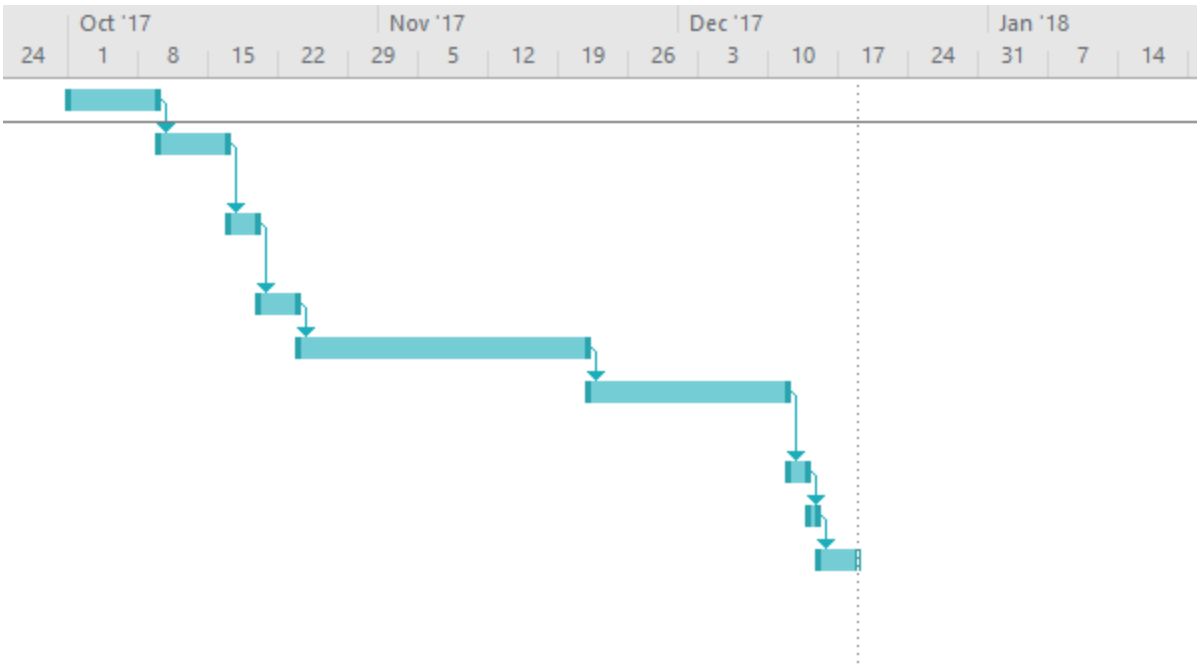
Participating_Students class will only produce a list of students who are participating in that event.

Student class will only generate a list of students who participate in that sport.

7. Preliminary Budget and Schedule

The schedule is as follows:

i. Project planning – 7 days.
ii. Requirements elicitation – 5 days.
iii. Requirements analysis – 3 days.
iv. Feasibility analysis – 2 days.
v. System coding – 21 days.
vi. Interface design – 14 days.
vii. System testing and debugging – 2 days.
viii. Configuration – 1 days
ix. Final Testing – 2 days.



	Planning	7 days	Sun 10/1/17	Mon 10/9/1	
	Requirements Illicitation	5 days	Tue 10/10/17	Mon 10/16/17	1
	Requirements Analysis	3 days	Tue 10/17/17	Thu 10/19/17	2
	Feasibility Analysis	2 days	Fri 10/20/17	Mon 10/23/	3
	Back and Coding	21 days	Tue 10/24/1	Tue 11/21/1	4
	User Interface Design	14 days	Wed 11/22/17	Mon 12/11/17	5
	Testing	2 days	Tue 12/12/1	Wed 12/13/	6
	Configuration	1 day	Thu 12/14/1	Thu 12/14/1	7
	Final Testing	2 days	Fri 12/15/17	Mon 12/18/	8

Expenditure Description	Budget Requested (Ksh)	Justification for Expenditures
Equipment		
Computer(Laptop)	35, 000	A computer is provided for in the lab (coding and testing)
Supplies and services		
Hosting	5,000	Hosting the system
Risks and uncertainties	4,000	Recovering from any risk that might happen
Project evaluation	2,000	For testing before implementation
Implementation and maintenance	4,000	Updating the system

8. References

Akhter, Z., 2016. *Project report.* [Online]
Available at: https://www.slideshare.net/zuhaib446/sports-event-management-systemreport

Bowker-Saur, 1992. *Information sources in sport and leisure.* 1st ed. London: Michele Shoebridge.

Kramer, S. D., February 17, 2009. *Fantasy Sports Ventures Takes Minority Stake In Sports Reference LLC.* 1st ed. Cambrigdge: Wikipedia.

Projectidea, 2013. *Sports Management System.* [Online]
Available at: http://www.studentprojectcode.com/vbnetproject/sports-management-system-project/

Van Vliet, H., 2000. *Software Engineering—Principles and Practice.* 2nd edn. ed. New York: John Wiley & Sons.