

# SPORTS MANAGEMENT SYSTEM

*Software Engineering*

*Project Report*

Submitted by:

Arun Chandra     3472 / 21025570019

Juman Kaushik     3478 / 21025570040

Nitin Dhiman     3466 / 21025570058

Aashish Kumar     3461 / 21025570088

Supervisor:



2023

Department of Computer Science  
Hansraj College  
University of Delhi

## **ACKNOWLEDGEMENT**

We wish to express our sincere gratitude to our Professor Mr. Suyash Kumar for providing his invaluable guidance, comments and suggestions throughout the course of the project. We also wish to express our gratitude to the officials and other staff members who rendered their help in requirement gathering in the Project. We also thank the Principal Professor (Dr.) Rama and Course Coordinator of B.Sc. (H) Computer Science Ms. Sunita Chand for providing us the opportunity to embark on this Project.

Problem Statement .....	1
Process Model .....	2
1. Software Requirement Specification .....	4
1.1 Overall Description .....	4
1.1.1 Product Functions.....	4
1.1.2 User Characteristics.....	5
1.1.3 General Constraints .....	6
1.1.4 Assumptions and Dependencies.....	6
1.2 External Interface Requirements .....	7
1.2.1 User Interfaces.....	7
1.2.2 Hardware Interfaces .....	8
1.2.3 Software Interfaces.....	9
1.3 Functional Requirements.....	10
1.3.1 User Login.....	10
1.3.2 Facility to issue sports item online.....	10
1.3.3 Item Return.....	11
1.3.4 Inventory Management (Add Item/Delete Item).....	12
1.3.5 Calculating Dues .....	13
1.4 Performance Requirement.....	13
1.4.1 Response Time .....	13
1.4.2 Concurrent Users.....	13
1.4.3 Scalability.....	14
1.4.4 Availability.....	14
1.4.5 Security.....	14
1.5 Design Constraints .....	14

1.5.1	Performance .....	14
1.5.2	Scalability.....	14
1.5.3	Usability .....	15
1.5.4	Security.....	15
1.5.5	Technology Constraints.....	15
1.6	Data Flow Diagram .....	15
1.6.1	Level 0.....	15
1.6.2	Level 1 .....	17
1.6.3	Level 2.....	18
	Login .....	18
	Issue Item .....	19
	Return Item .....	20
	Updating Inventory .....	21
	Calculating Dues .....	22
	Report Generation.....	23
1.7	Use Cases .....	24
1.8	Data Dictionary .....	32
2.	Estimations.....	34
2.1	Function Points.....	34
2.2	Efforts.....	36
3.	Scheduling.....	38
4.	Risk Management .....	39
5.	Design .....	42
5.1	System Design.....	42

5.2	Data Design .....	45
6.	Coding.....	47
7.	Testing.....	51
8.	References.....	78
9.	Appendix.....	79

## Problem Statement

Hansraj College wishes to develop a software for Sports Management that manages and maintains all the records of sports items issued by sports authority of college to students. Students face multiple issues while issuing any sports items manually because of its availability and usability. Also, it becomes an overhead for the sports faculty to manage all the details of issuing items and inventory on paper on a regular basis. So, this project will complete all the procedures digitally which will be convenient for all the students and authority.

- The system should include an efficient way of managing the inventory by keeping records of all the sports items.
- Information of all the exchanges done by the authority to any students.
- Admin should have all the information of fine generated by the system for not returning any items.

This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes.

Thus, this system reduces manual work to a great extent allows smooth flow of sports management by removing chances of errors in the details.

## Process Model

For this Sports Management System project, we have decided to follow a linear and sequential process model, also known as the Waterfall Model. This approach consists of a series of sequential phases, where each phase must be completed before moving on to the next phase.

The Waterfall Model comprises the following sequential phases:

1. Requirements gathering: In this phase, the requirements for the project are identified and documented. This includes identifying the functional and non-functional requirements, as well as any constraints and assumptions that may affect the project.

2. Design: In this phase, the system architecture is designed, including the hardware and software components, data flow diagrams, and user interfaces.

3. Implementation: In this phase, the software is developed based on the design specifications. This includes writing the code, integrating the different components, and performing unit testing.

4. Testing: In this phase, the software is tested to ensure that it meets the functional and non-functional requirements, and that there are no defects or errors.

5. Deployment: In this phase, the software is deployed to the production environment and made available for use by end-users.

6. Maintenance: In this phase, any issues or bugs that are reported by end-users are fixed,

and any updates or enhancements to the software are made.

The linear and sequential approach of the Waterfall Model ensures that each phase is completed before moving on to the next phase, which helps to minimize errors and ensure that the final product meets all the specified requirements. However, it does not allow for much flexibility or adaptability during the development process, which can be a limitation in some cases.



# 1. Software Requirement Specification

## 1.1 Overall Description

The Sports Management System is a computerized system that aims to streamline the sports management process in Hansraj College. This system allows for efficient management of the sports inventory and records of sports items issued by the sports authority of the college to students. It enables sports faculty to issue and return sports items digitally for students and removes the need for manual paperwork.

The system offers various features such as a user login system for students and staff, an inventory of sports items, and automatic fine calculation for students who fail to return items on time. It also provides an online interface for the admin to view details of sports items issued, issue and return time within a day, and a digital record of all the past issuance and returns.

The Sports Management System is designed to make the sports management process more efficient, convenient and error-free. By providing an automated solution, it minimizes the chances of manual errors and makes the process of sports item issuance and management more transparent. The system will be implemented using a linear and sequential approach, with each phase building on the previous one.

In the following sections, we will discuss the product functions, user characteristics, general constraints, assumptions and dependencies, external interface requirements, functional requirements, performance requirements, design constraints, data flow diagram, and data dictionary of the Sports Management System.

### 1.1.1 Product Functions

The product functions of the Sports Management System are as follows:

1. Inventory Management: The system allows the sports faculty to manage the inventory of sports items by keeping records of all the sports items available and issued to the students. Also, sports faculty can add or delete items from the online inventory when any quantity of that item changes in the physical sports room. The faculty can view the current stock of items and issue them to the students.

2. Issue and Return of Sports Items: The system provides a facility for sports faculty to issue and return sports items online on Student ID when student want to issue or return. The students can request the sports faculty for issuance and return.

3. Calculating Due: The system has a mechanism to calculate the fine amount for the students who do not return the sports items on time. The sports faculty can generate the fine report after a certain period of time.

4. User Authentication: The system provides secure user authentication for the sports faculty ensuring that only authorized users can access the system.

5. Report Generation: The system generates reports for the administration, which includes the list of sports items issued and returned, fine generated, and other relevant details. These reports help the administration in decision making and monitoring purposes.

### **1.1.2 User Characteristics**

The users of the Sports Management system can be classified into one category:

1. Sports Faculty/Manager: The primary user of the system would be the sports faculty/manager. They are responsible for maintaining the inventory of sports items and issuing them to the students. They should have knowledge about the sports items, their quantity, and usage.

They will be authorized to access the system to manage the inventory and issue sports items to students.

### **1.1.3 General Constraints**

The following are the general constraints for the sports management system:

1. The system should be developed in a way that it can be easily used by the Sports Faculty/Manager without requiring any specialized technical knowledge.
2. The system should be designed in a way that it can handle a large number of students, sports items and their details efficiently
3. The system should be secure and only authorized users should be able to access it.
4. The system should be scalable and able to accommodate future changes and updates easily.
5. The system should be compatible with various devices and browsers, so that users can access it from different platforms.
6. The system should be designed in a way that it is user-friendly and has an intuitive interface, so that it is easy for users to navigate and perform tasks.
7. The system should be robust and able to handle unexpected errors and crashes without losing data.
8. The system should be developed in a way that it complies with relevant laws and regulations related to data protection and privacy.

### **1.1.4 Assumptions and Dependencies**

Assumptions:

1. The hardware and software requirements for the proposed sports management system

are readily available and meet the system requirements.

2. The users of the system have a basic understanding of how to use a computer and are familiar with basic software applications.

3. The system will be developed using a linear and sequential process model.

4. The database used in the system is reliable and secure.

5. The system will be maintained and updated regularly to meet the changing needs of the sports department.

Dependencies:

1. The proposed sports management system is dependent on the availability of a reliable internet connection.

2. The system is dependent on the timely submission of information by the sports faculty/manager.

3. The system is dependent on the availability of hardware and software resources required for its smooth functioning.

4. The system is dependent on the availability of accurate and up-to-date information related to sports items inventory and student records.

## ***1.2 External Interface Requirements***

### **1.2.1 User Interfaces**

The user interface of the sports management system will consist of a web-based interface, accessible by the sports faculty/manager. The interface will be simple and user-friendly with easy navigation, and all the necessary functions will be available on a single dashboard.

The following are the main user interfaces of the system:

1. Login Interface: This interface will be used by the sports faculty/manager to log in to the system using their unique username, email and password.

2. Dashboard Interface: This interface will be the main page of the system and will display all the necessary information for the sports faculty/manager. It will have the following functionalities:

- Issuing Sports Item on Student ID
- Returning Sports Item on Student ID
- Inventory Management (Add Item/Delete Item)
- View pass generated reports
- View fine generated for late returns

3. Issue/Return Interface: This interface will allow the sports faculty/manager to issue sports items to the students, record the return date, and generate fines for late returns. The interface will require the following information to issue/return sports items:

- Student ID
- Sports Item ID
- Sports Category under which the item lies

The user interface will be designed using modern web development technologies and will be responsive to ensure it can be accessed from any device, including desktops, laptops, tablets, and smartphones.

### **1.2.2 Hardware Interfaces**

As the proposed sports management system is a software solution, there are no specific hardware requirements for the end-users. However, the system will require a standard desktop or

laptop computer to run the software.

### 1.2.3 Software Interfaces

The sports management system has several software interfaces that are necessary for its operation. These include:

1. Database Management System: The system requires a database management system to store all the information related to sports items, students, and faculty. The system will be designed to work with any standard database management system such as MySQL, Oracle, or SQL Server.

2. Web Server: The system requires a web server to host the web application. The web server should support HTML, CSS, and backend language i.e., Python Flask and MySQL to host the sports management system.

3. Browser: The system requires a browser to access the web application. The web application will be designed to work with any modern browser such as Google Chrome, Mozilla Firefox, or Safari.

4. Operating System: The sports management system will be designed to work on any standard operating system such as Windows, Linux, or MacOS.

5. Programming Language: The system will be developed using Python Flask programming language with HTML, CSS for the user interface.

These software interfaces are essential for the proper functioning of the sports management system.

## **1.3 Functional Requirements**

### **1.3.1 User Login**

Functional Requirement 1: User Login

- Description: The system shall provide a user login functionality to allow only authorized sports faculty/managers to access the system's features.

- Inputs: Username, email and password

- Processing: The system shall verify the user's credentials with the stored database of authorized users and grant access if the credentials match.

- Outputs: A successful login message and access to the system's features for authorized users.

- Dependencies: The user must have a valid user ID and password to log in to the system. The system must have access to the stored database of authorized users to verify the user's credentials.

### **1.3.2 Facility to issue sports item online**

Functional Requirement 2: Facility to issue sports item online

Description: The system should allow the sports faculty/manager to issue sports items on the student id online. The faculty/manager will log in to the system, select the student who wants to borrow the sports item, and select the item from the inventory. The system should then update the inventory and keep a record of the issuance, including the date and time of issuance, the name of the student, and the item borrowed.

Inputs:

- Login credentials of the sports faculty/manager
- Student details (name, ID, etc.)
- Sports item details (ID, Sports Category, etc.)

Processing:

- Verify the login credentials of the sports faculty/manager
- Select the student from the list of registered students
- Select the item from the inventory
- Update the inventory to reflect the issuance
- Record the issuance details (date, time, student name, item borrowed) in the system

Outputs:

- Confirmation of issuance to the sports faculty/manager
- Updated inventory details.

### **1.3.3 Item Return**

Functional Requirement: Item Return

Description: The system shall allow the Sports Faculty/Manager to return the sports items to the inventory after they have been used.

Inputs: Student ID, Item ID, Sports Category

Processing: The system shall verify the item ID and check if the item is available for return. If the item is eligible for return, the system shall update the inventory with the returned item, and



mark the item as available for issue. The system shall also update the item condition based on the input received.

Outputs: Confirmation message of the return status, Updated inventory status.

#### **1.3.4 Inventory Management (Add Item/Delete Item)**

Functional Requirement: Inventory management (Add Item/Delete Item)

Description: The system shall provide functionality for the sports faculty/manager to manage the inventory by adding new items or deleting existing items.

Inputs:

- Add Item: Sports Item ID, Sports Item Category, Item Quantity
- Delete Item: Sports Item ID, Sports Item Category, Item Quantity

Processing:

- Add Item: The system shall validate the input data, check for duplicate items, and add the new item to the inventory.
- Delete Item: The system shall validate the input data, check for the existence of the item in the inventory, and remove the item from the inventory.

Outputs:

- Add Item: A confirmation message indicating successful addition of the new item to the inventory.
- Delete Item: A confirmation message indicating successful removal of the item from the inventory.

### **1.3.5 Calculating Dues**

Functional Requirement: Calculating Dues

Description: The system shall provide functionality to calculate the dues for any sports item that is not returned within the specified period. The due amount shall be calculated based on the number of days elapsed since the due date and the present fine rate per day. It will work with the return item Module. When a sports item is being returned on a student id, if there exist some dues on the student, it will run and return the calculating due on the user's interface and also provide a facility to clear the due itself.

Inputs: Student ID, Sports Item ID, Sports Item Category

Processing: Calculate the number of days elapsed since the due date, Multiply the fine rate per day by the number of days elapsed, Generate a due amount

Outputs: Calculated Due amount till that time and date

## ***1.4 Performance Requirement***

### **1.4.1 Response Time**

The system shall provide a response time of less than 5 seconds for all user interactions, including adding or removing items, checking inventory status, and calculating dues.

### **1.4.2 Concurrent Users**

The system shall be able to handle at least 50 concurrent users without a significant

decrease in performance.

### **1.4.3 Scalability**

The system shall be designed to handle a minimum of 1000 items in inventory and shall be scalable to handle up to 5000 items without a significant decrease in performance.

### **1.4.4 Availability**

The system shall be available 24/7 with a maximum downtime of 2 hours per week for maintenance.

### **1.4.5 Security**

The system shall provide secure user authentication and access control mechanisms to prevent unauthorized access to the inventory data. The system shall also encrypt sensitive data such as user login credentials and financial transactions. The system shall comply with all applicable data privacy and security regulations.

## ***1.5 Design Constraints***

### **1.5.1 Performance**

The system shall be designed to respond to user requests within a maximum response time of 3 seconds. The database queries shall be optimized to minimize the response time.

### **1.5.2 Scalability**

The system shall be designed to handle a large number of concurrent users. It shall be scalable enough to accommodate increasing numbers of users and transactions without compromising the system's performance.

### **1.5.3 Usability**

The user interface shall be designed in a way that is easy to use and understand for the sports faculty/manager. The system shall have clear and concise instructions for using its features.

### **1.5.4 Security**

The system shall be designed with appropriate security measures to ensure the confidentiality and integrity of the data. User authentication and authorization shall be implemented to prevent unauthorized access to the system. Data encryption shall be used to protect sensitive information during transmission.

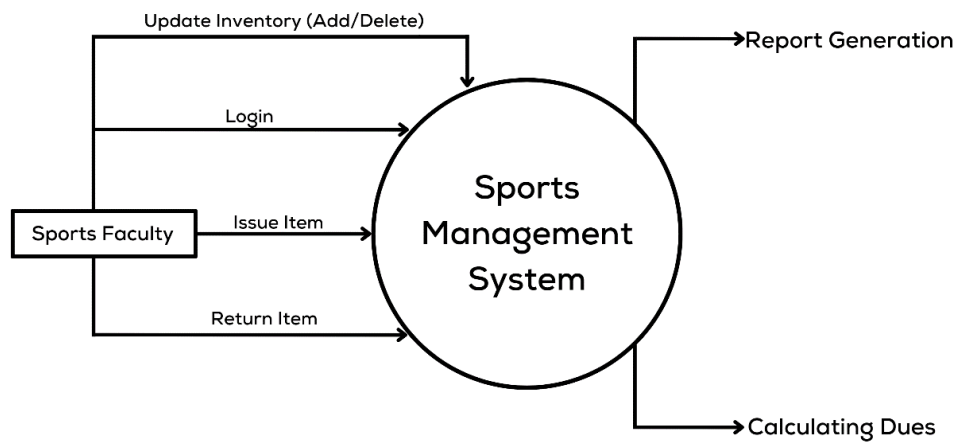
### **1.5.5 Technology Constraints**

The system shall be designed using appropriate technology stacks, such as a suitable programming language, web server, and database management system. The hardware and software used shall comply with the organization's standards and requirements.

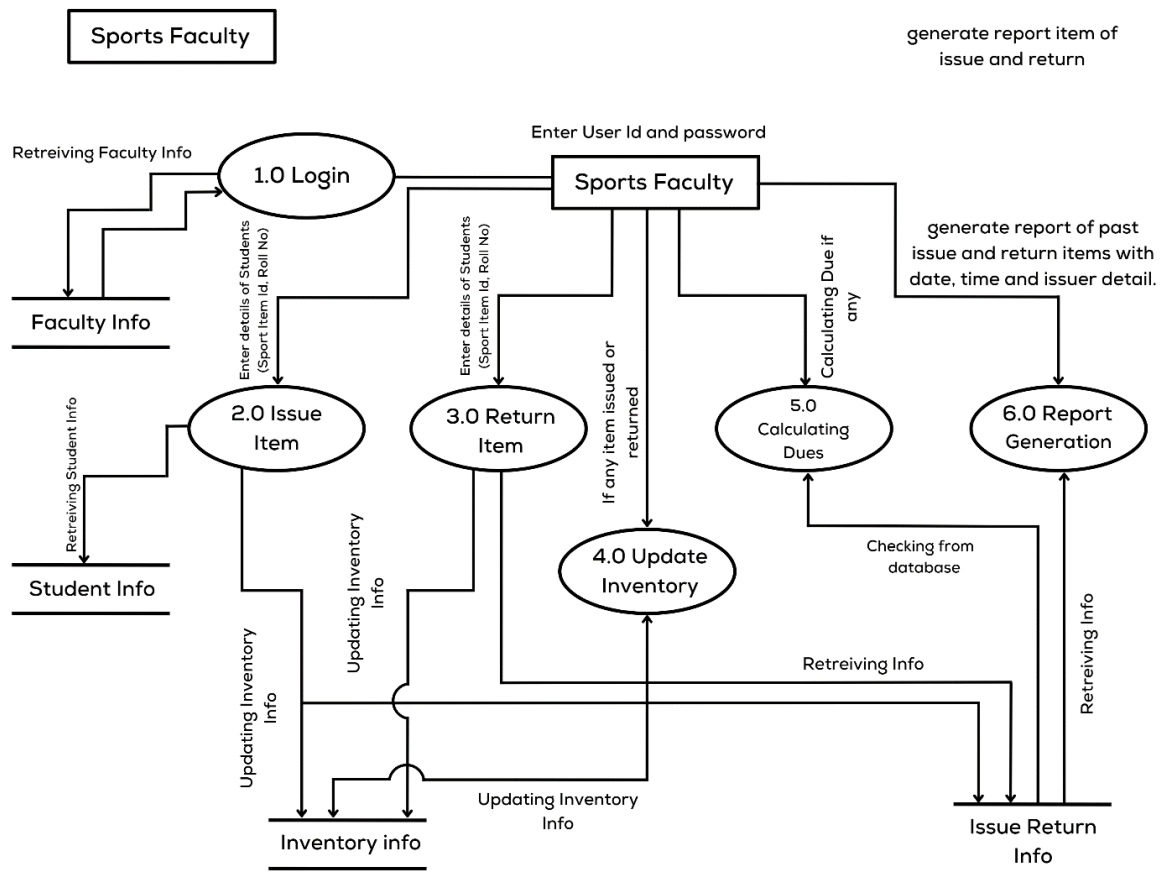
software used shall comply with the organization's standards and requirements.

## ***1.6 Data Flow Diagram***

### **1.6.1 Level 0**

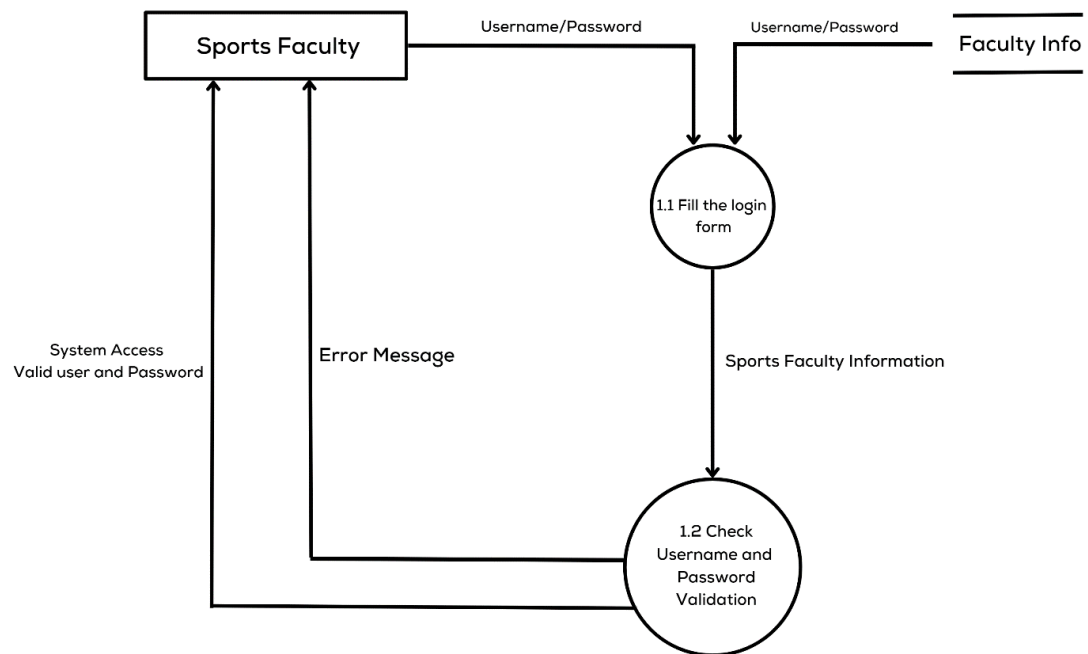


## 1.6.2 Level 1

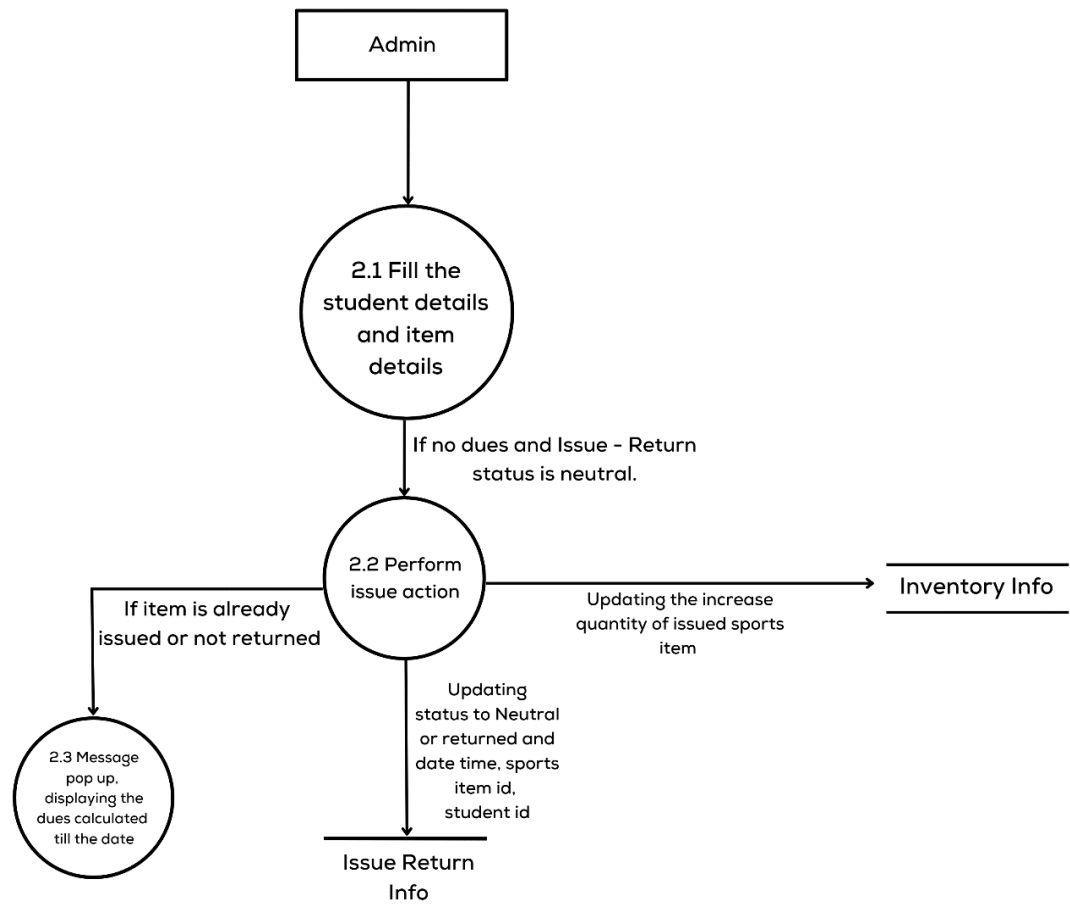


### 1.6.3 Level 2

#### Login

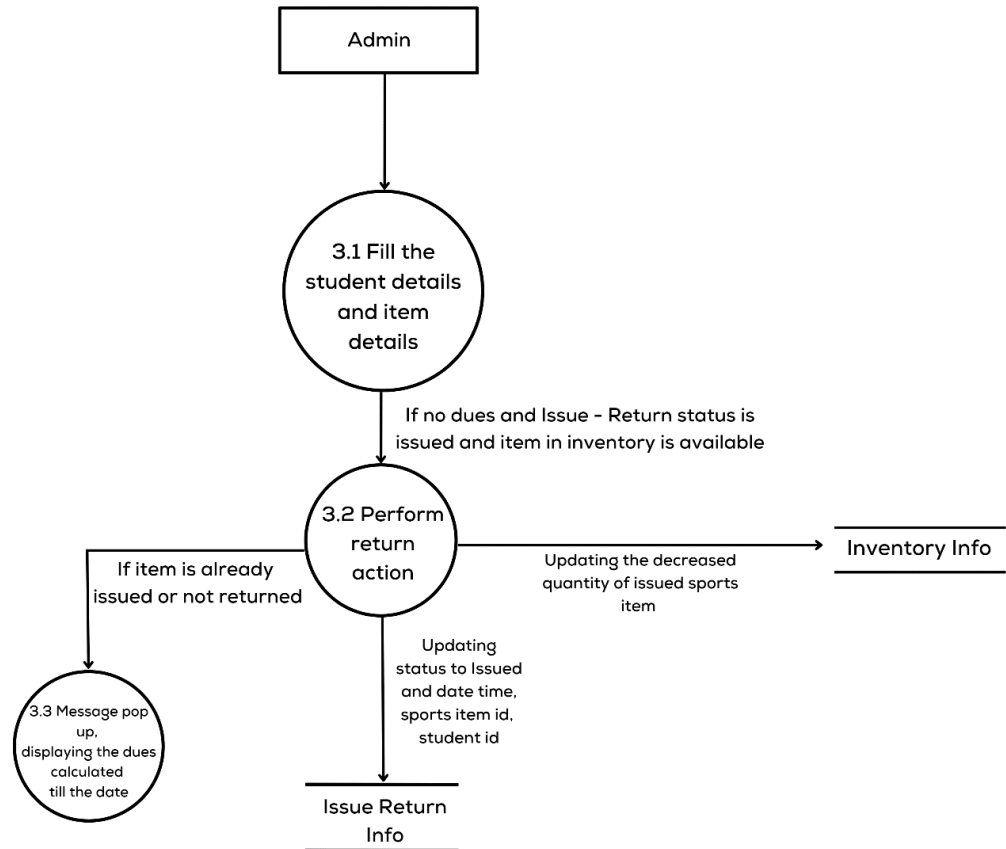


## Issue Item

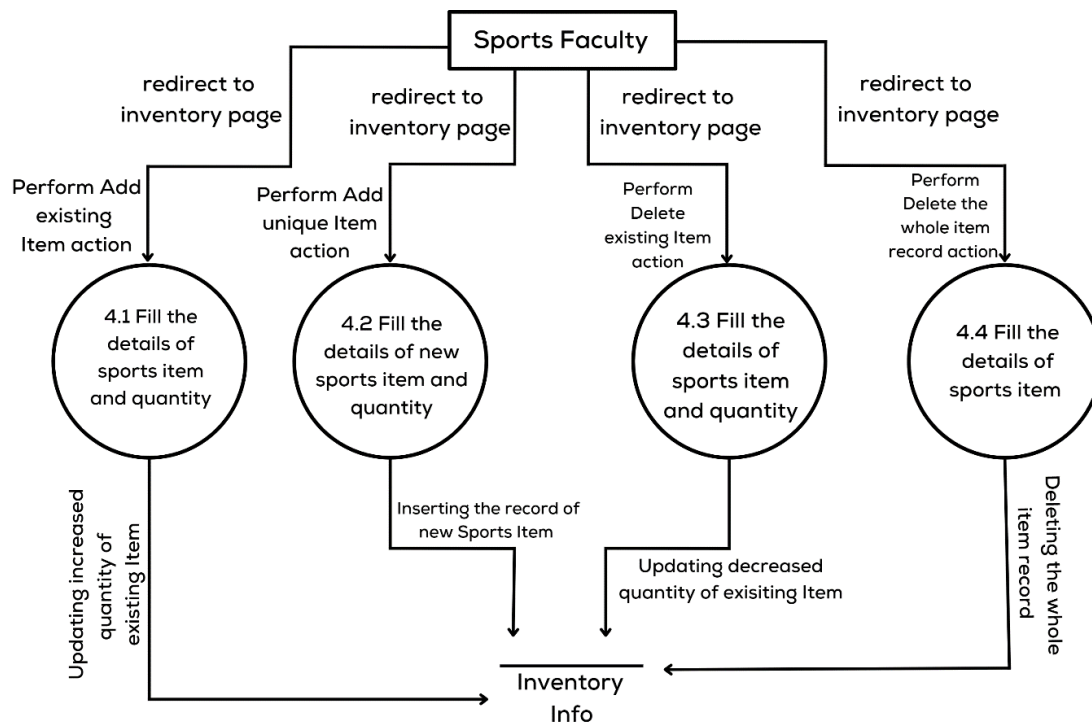




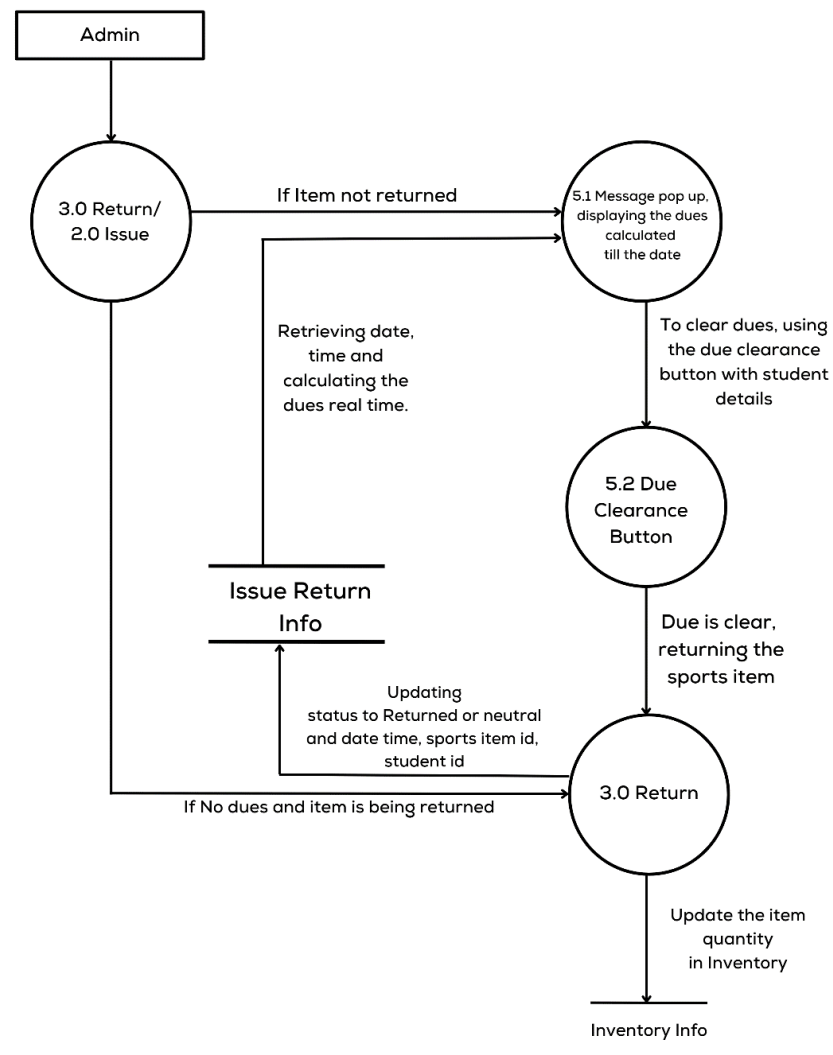
## Return Item



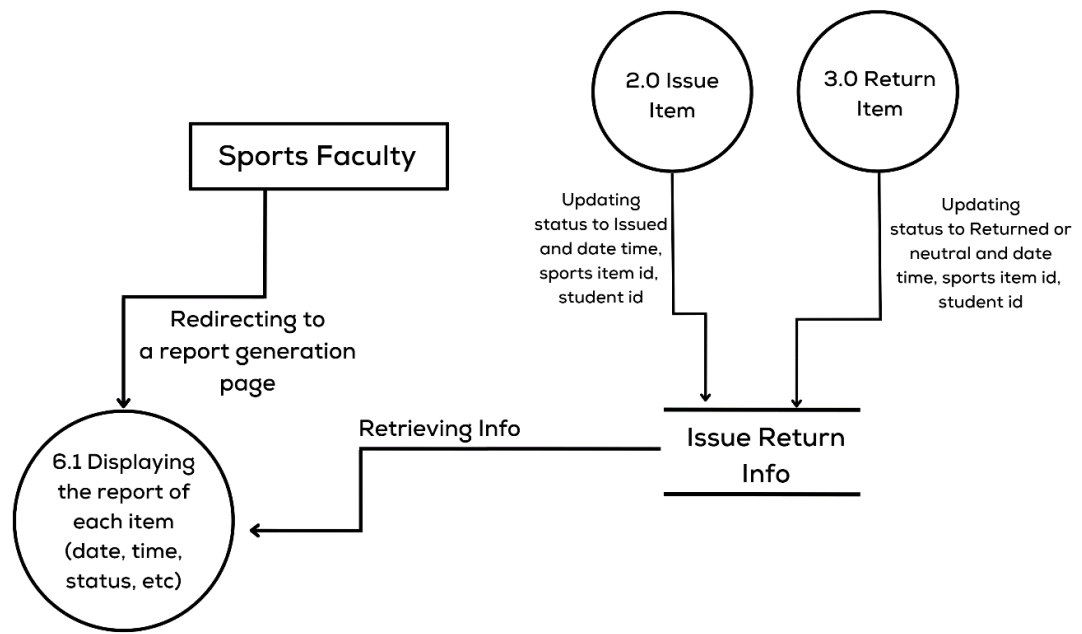
## Updating Inventory



## Calculating Dues



## Report Generation



## **1.7 Use Cases**

### **1) Login**

#### **1.1 Brief description**

This use case allows the actor to log in into the system. use case starts when the actors will be log into the admin/user panel.

#### **1.2 Actors**

Sports Faculty

#### **1.3 Flow of Events**

##### **1.3.1 Basic Flow**

This use case starts when the actor wishes to login into the system.

- 1) The system will request actor to enter the user id and password given from the officials.
- 2) The actor will enter his/her user id and password.
- 3) The system validates the entered information.

##### **1.3.2 Alternative Flows**

###### **1.3.2.1 Invalid Name/ Password /Role**

If the actor enters an invalid information system displays an error

#### **1.4 Special Requirements**

None

#### **1.5 Pre-Conditions**

Must have official user-id and password given from the college.

#### **1.6 Posts-Conditions**

Successfully logged in.

#### **1.7 Extension Points**

None

### **2) Add Items in Inventory**

#### **2.1 Brief description**

This use case allows the actor to add one or more than one sports item in the inventory.

## **2.2 Actors**

Sports Faculty

## **2.3 Flow of Events**

### **2.3.1 Basic Flow**

This use case starts when the actor wishes to add sports in the inventory.

- 1) Actor will have an option available to add items in the inventory.
- 2) He/she can add new item with the specified add option.
- 3) The system will add the new item in the inventory.

### **2.3.2 Alternative Flows**

**2.3.2.1** If the item is already in the inventory, Actor want to increase the number of respective items.

Actor will have an another add option with each individual distinct item to increase the number of it. He/she can add items by using that option.

## **2.4 Special Requirements**

None

## **2.5 Pre-Conditions**

Actor must be logged in successfully.

## **2.6 Posts-Conditions**

Record of sports items are added in the inventory.

## **2.7 Extension Points**

None

# **3) Delete Items in Inventory**

## **3.1 Brief description**

This use case allows the actor to delete sports item in number in the inventory.

## **3.2 Actors**

Sports Faculty

## **3.3 Flow of Events**

### **3.3.1 Basic Flow**

If the item is already in the inventory, Actor want to delete the sports item. In other words, he/she want to decrease the number of respective items.

Actor will have a delete option with each individual distinct item to decrease the number of it. He/she can delete items by using that option.

### **3.3.2 Alternative Flows**

None

### **3.4 Special Requirements**

None

### **3.5 Pre-Conditions**

Actor must be logged in successfully.

### **3.6 Posts-Conditions**

Record of sports items are deleted in the inventory.

### **3.7 Extension Points**

None

## **4) Calculating Dues**

### **4.1 Brief description**

This use case allows the actor to get the calculated dues on not returned item.

### **4.2 Actors**

Sports Faculty

### **4.3 Flow of Events**

#### **4.3.1 Basic Flow**

Sports Authority want to see the calculated dues' record of not returned sports items.

- If the student doesn't return the item on time, a penalty will be given and dues will be counted based on that.
- The due will be calculated based on how much time he/she has exceeded.

#### **4.3.2 Alternative Flows**

The dues will be withdrawn after paying the due.

- No penalty if returned on time.

### **4.4 Special Requirements**

None

#### **4.5 Pre-Conditions**

Actor must be logged in successfully & students didn't return the items on time.

#### **4.6 Posts-Conditions**

Actor can see the report of dues.

#### **4.7 Extension Points**

None

### **5) Issue Item**

#### **5.1 Brief description**

Student wants to issue any sport item linked to his/her id.

#### **5.2 Actors**

Sports Faculty

#### **5.3 Flow of Events**

##### **5.3.1 Basic Flow**

Availability check: First the availability of the sports item will be checked in the inventory.

Issue status: The authority confirms the issue of a sport item and the system updates the status to "issued".

##### **5.3.2 Alternative Flows**

If the item is not available then he/she request the item and not found message will be displayed.

#### **5.4 Special Requirements**

Conditions on students should be by sports authority like item should not be damaged and defecting.

#### **5.5 Pre-Conditions**

Student's id exist in the student database and the item should be available in inventory.

#### **5.6 Posts-Conditions**

Sports Item issued until he/she returned.

#### **5.7 Extension Points**

None



## **6) Return Item**

### **6.1 Brief description**

Student want to return the issued item linked to his/her id.

### **6.2 Actors**

Sports Faculty

### **6.3 Flow of Events**

#### **6.3.1 Basic Flow**

The sport authority selects the option of returning item

The system confirms the item has been returned and update status of item to “available”.

#### **6.3.2 Alternative Flows**

The authority inspects the item defected or not after returning.

If he/she didn't return, dues for the same will be increasing.

The system will update information of student.

### **6.4 Special Requirements**

None

### **6.5 Pre-Conditions**

Sport item has been issued on student' id.

### **6.6 Posts-Conditions**

Item has been returned and status has been updated in the system.

### **6.7 Extension Points**

None

## **7) Generating Report**

### **7.1 Brief description**

This use case allows the actor to get the generated reports of all past issue and return items.

### **7.2 Actors**

Sports Faculty

### **7.3 Flow of Events**

### **7.3.1 Basic Flow**

The actor will have an option to get reports of past issue and return items in one go.

He/she can get the reports by using that.

### **7.3.2 Alternative Flow**

None

## **7.4 Special Requirements**

None

## **7.5 Pre-Conditions**

Actor must be logged in successfully.

## **7.6 Posts-Conditions**

Report generated successfully and actor can see it.

## **7.7 Extension Points**

None

# **8) Logout**

## **8.1 Brief description**

This use case allows the actor to logout from the system.

## **8.2 Actors**

Sports Faculty

## **8.3 Flow of Events**

### **8.3.1 Basic Flow**

This use case starts when the actor wishes to logout from the system.

The system will have an option to perform the logout action. The actor can perform the same by using that.

### **8.3.2 Alternative Flows**

None

## **8.4 Special Requirements**

None

## **8.5 Pre-Conditions**

Actor must be logged in successfully.

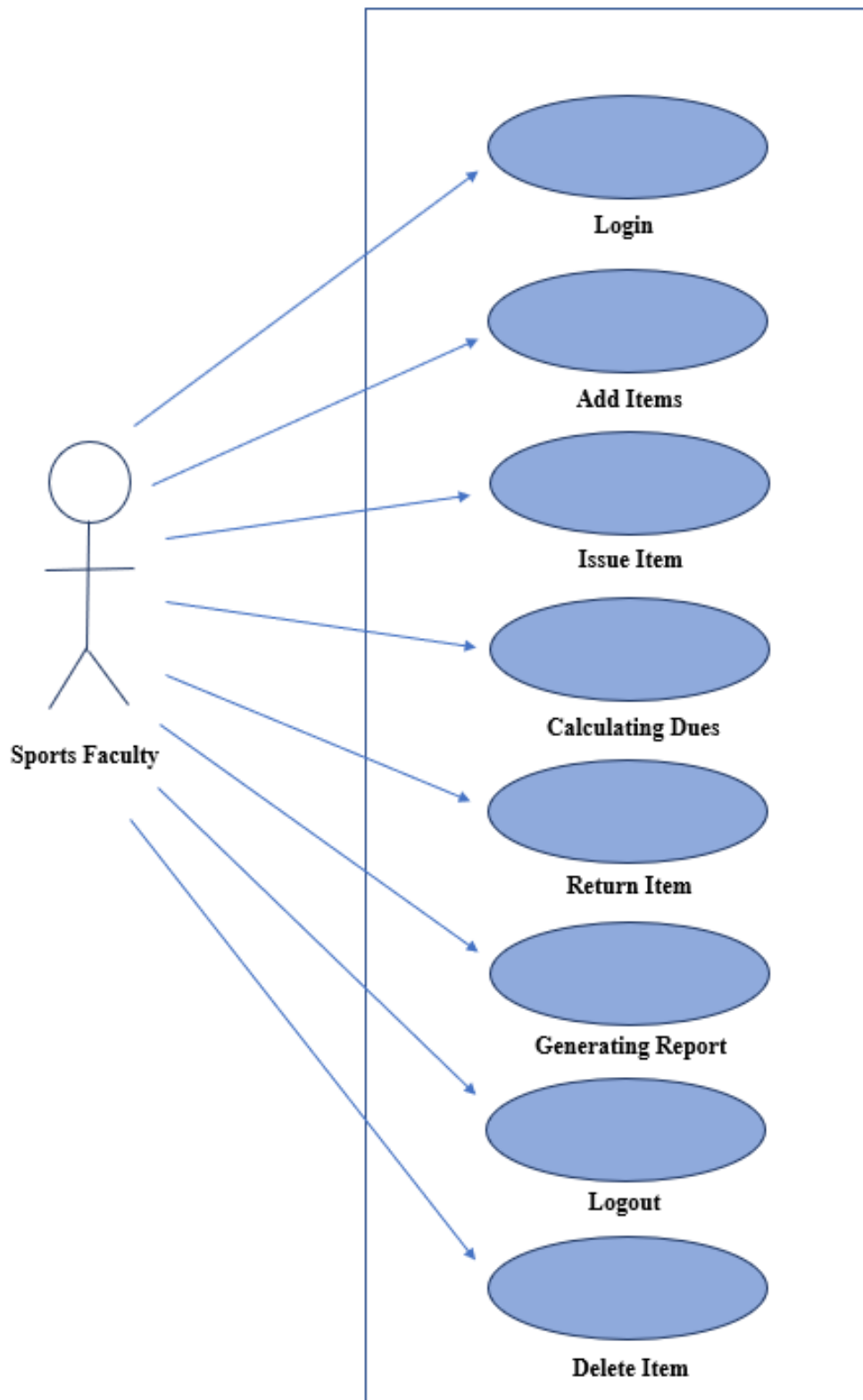
**8.6 Posts-Conditions**

Actor has been logged out from the system.

**8.7 Extension Points**

None

Use Case Diagram:



## 1.8 Data Dictionary

### Student info

Attribute Name	Data Type	Size	Constraint	Description	Example
<b>SID</b>	int	10	Primary Key, Not Null	Unique identifier for student (Roll No)	3472
<b>Name</b>	varchar	25	Not Null	Name of the Student	Arun Chandra
<b>Course</b>	varchar	50	Not Null	Course of a Student	B.Sc. Hons CS
<b>Semester</b>	int	5	Not Null	Semester of student	IV
<b>Due</b>	Int	5	Not Null	Due	0

### Faculty Info

Attribute Name	Data Type	Size	Constraint	Description	Example
<b>Username</b>	varchar	20	Primary Key, Not Null	Unique username for faculty	Hari20012
<b>Email</b>	varchar	20	Not Null	Email address of faculty	hari20012@gmail.com
<b>Password</b>	varchar	20	Not Null	Password	johndoe@123

### Issue Info

Attribute Name	Data Type	Size	Constraint	Description	Example
<b>SID</b>	int	10	Foreign Key, Not Null	Unique Identifier for Student. Foreign Key is linked with Table "Student Info"	3472
<b>Item ID</b>	varchar	20	Primary Key, Not Null	Unique Identifier for Item	CBAT
<b>Sports Name</b>	varchar	50	Not Null	Sports Name under which the Item lies	CRICKET
<b>Status</b>	varchar	10	Not Null	Status of Item - Issued	IV
<b>Date</b>	Date	-	Not Null	Date on which the item is issued	2023-03-11
<b>Time</b>	Time	-	Not Null	Time on which the item is issued	08:26:48

### Return Info

Attribute Name	Data Type	Size	Constraint	Description	Example
<b>SID</b>	int	10	Foreign Key, Not Null	Unique Identifier for Student. Foreign Key	3472

				is linked with Table "Student Info"	
<b>Item ID</b>	varchar	20	Primary Key, Not Null	Unique Identifier for Item	CBAT
<b>Sports Name</b>	varchar	50	Not Null	Sports Name under which the Item lies	CRICKET
<b>Status</b>	varchar	10	Not Null	Status of Item - Returned	IV
<b>Date</b>	Date	-	Not Null	Date on which the item is returned	2023-03-11
<b>Time</b>	Time	-	Not Null	Time on which the item is returned	10:16:32

## Inventory Info

Attribute Name	Data Type	Size	Constraint	Description	Example
<b>Sports Name</b>	varchar	50	Not Null	Sports Name under which the Item lies	CRICKET
<b>Item ID</b>	varchar	20	Primary Key, Not Null	Unique Identifier for Item	CBAT
<b>Quantity</b>	Int	100	Not Null	Available Quantity of Sports Item for Issuance	17
<b>Total Quantity</b>	Int	100	Not Null	Total Quantity of Sports Item	20

## 2. Estimations

### 2.1 Function Points

FACTOR	VALUE
BACKUP AND RECOVERY	0
DATA COMMUNICATION	3
DISTRIBUTED PROCESSING FUNCTIONS	0
CRITICAL PERFORMANCE	0
PERFORMANCE IN EXISTING AND HEAVILY UTILISED ENVIRONMENT	1
ONLINE DATA ENTRY	3
INPUT TRANSACTION BUILT OVER MULTIPLE SCREENS	2
ILFS UPDATED ONLINE	1
INPUTS, OUTPUTS, FILES COMPLEX	0
INTERNAL PROCESSING COMPLEX	1
REUSABLE CODE	1
CONVERSIONS AND INSTALLATIONS INCLUDED	0
MULTIPLE INSTALLATIONS	0
CHANGE AND EASE OF USE	3

$$\Sigma(F_i) = 15$$

PARAMETERS	COUNT	SIMPLE	AVERAGE	DIFFICULT	TOTAL
<b>EXTERNAL INPUT</b>	6	6X3	0X4	0X6	18
<b>EXTERNAL OUTPUT</b>	5	5X4	0X5	0X7	20
<b>EXTERNAL INQUIRIES</b>	1	1X3	0X4	0X6	3
<b>EXTERNAL INTERFACE FILES</b>	0	0X5	0X7	0X10	0
<b>INTERNAL LOGICAL FILES</b>	4	0X7	4X10	0X15	40

$$\text{UFP} = \text{EI} + \text{EO} + \text{EQ} + \text{ILF} + \text{EIF} = 81$$

$$\text{FP} = \text{UFP} * [0.65 + 0.01 * (\Sigma(\text{Fi}))]$$

$$= 81 * [0.65 + 0.01 * 15]$$

$$= 64.8$$



## 2.2 Efforts

To calculate the effort for the Sports Equipment Management System using the COCOMO II model, we need to consider the following parameters:

### 1. Scale Factors:

The scale factors are used to adjust the estimated effort based on the project's characteristics. The scale factors and their values for the Sports Equipment Management System are as follows:

- Precedent Ness: Low
- Development Flexibility: High
- Architecture/Risk Resolution: Very High
- Team Cohesion: Very High
- Process Maturity: High

### 2. Cost Drivers:

The cost drivers are used to adjust the estimated effort based on the project's attributes. The cost drivers and their values for the Sports Equipment Management System are as follows:

- Required Software Reliability: Very High
- Data Base Size: Low
- Product Complexity: High
- Required Reusability: Very High
- Documentation Match to Life-Cycle Needs: Very High

- Execution Time Constraint: Very High
- Storage Constraint: Low
- Platform Volatility: Low
- Analyst Capability: High
- Programmer Capability: High
- Personnel Continuity: Very High
- Applications Experience: High
- Platform Experience: Low
- Language and Tool Experience: High

### 3. Size:

The estimated size of the Sports Equipment Management System is 900 function points.

Using the above parameters and the COCOMO II model, we can estimate the effort as follows:

$$\text{Effort} = A * \text{Size}^E * M$$

where  $A = 2.94$  (for the Basic COCOMO model with a project size of 900 function points)

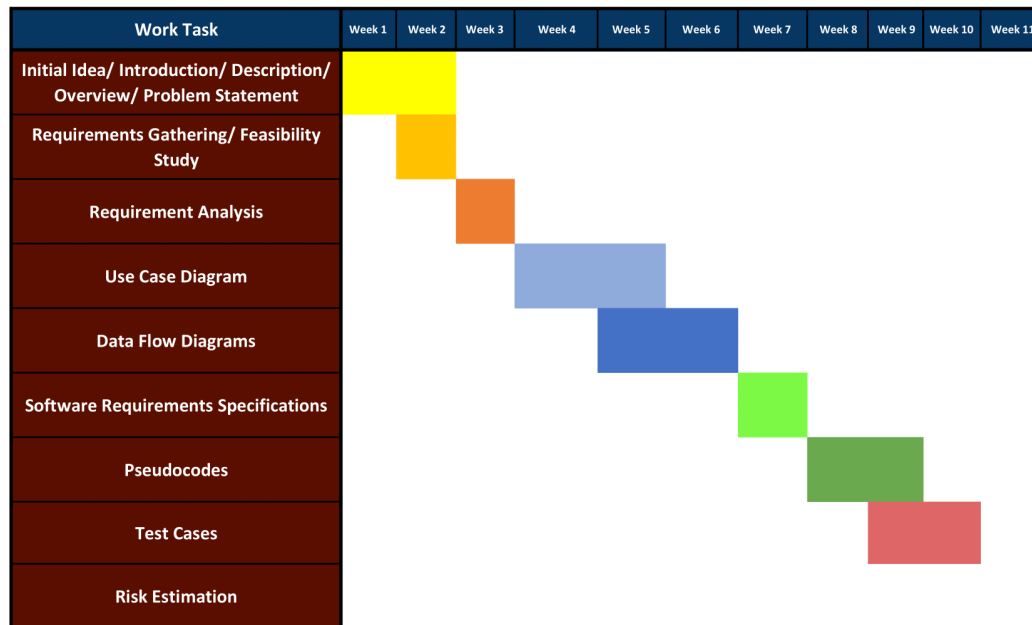
$E = 1.10$  (for the Basic COCOMO model)

$M = \text{Product of all the cost drivers, which is } 1.790$

Therefore,  $\text{Effort} = 2.94 * 900^{1.10} * 1.790 = 7.76$  person-months (approximately)

Hence, the estimated effort required to develop the Sports Equipment Management System using the COCOMO II model is 7.76 person-months.

### 3. Scheduling



## 4. Risk Management

### Risk Management:

The success of the Sports Management System project depends on the effective management of potential risks that may arise during the project's lifecycle. The following risk management plan outlines the steps to identify, assess, and mitigate these risks.

### Risk Identification:

The project team will identify potential risks in the following areas:

- Technical risks: including hardware and software failures, integration issues, and performance problems.
- Schedule risks: including delays caused by unforeseen circumstances, resource constraints, and dependencies on third-party suppliers.
- Financial risks: including budget overruns, unexpected costs, and unforeseen expenses.
- Human risks: including staff turnover, skill gaps, and training issues.
- Security risks: including unauthorized access, data breaches, and cybersecurity threats.

### Risk Assessment:

For each identified risk, the project team will assess its probability of occurring and its potential impact on the project's objectives. The following risk assessment matrix will be used to prioritize risks:

### Risk Mitigation:

The project team will develop and implement risk mitigation strategies to reduce the impact and probability of risks. The following strategies will be used:

- Technical risks: implementing quality assurance processes, conducting thorough testing, and ensuring compatibility between hardware and software components.
- Schedule risks: creating a realistic project schedule, identifying critical paths, and using project management tools to monitor progress.
- Financial risks: developing a detailed budget, conducting regular cost reviews, and forecasting potential budget overruns.
- Human risks: conducting regular staff reviews, providing training and development opportunities, and creating a positive team culture.
- Security risks: implementing access control measures, using encryption, and conducting regular security audits.

#### Risk Monitoring and Control:

The project team will monitor and control risks throughout the project's lifecycle by regularly reviewing risk assessments, tracking risk mitigation strategies, and adjusting the risk management plan as needed. The team will also communicate risk status to stakeholders and take appropriate actions to address any emerging risks.

#### Conclusion:

Effective risk management is essential to the success of the Sports Management System project. By identifying, assessing, and mitigating potential risks, the project team can ensure that the project meets its objectives and delivers high-quality results.

Risk	Category	Probability	Impact
Data loss due to server failure	Technical	Low	High
Cyber-attack resulting in data breach	Security	Medium	High

System downtime due to power outage	Technical	Low	Medium
Incomplete or inaccurate data entered by user	Human Error	High	Medium
Inability to integrate with existing system	Technical	Medium	Medium
Lack of user adoption due to poor user interface	Usability	High	Medium
Inadequate training provided to users	Human Error	Medium	Low
Incomplete or incorrect requirement gathering	Planning	Medium	High
Inability to scale to meet growing user demand	Technical	Low	High

## 5. Design

### 5.1 System Design

Based on the functional and non-functional requirements, the following is a high-level system design for the sports equipment management system:

#### 1. Architecture:

The system will follow a three-tier architecture: Presentation layer, Application layer, and Database layer.

#### 2. Presentation Layer:

The user interface will be a web-based application, designed to be user-friendly and intuitive. The interface will have the following components:

- Issuing Sports Item on Student ID
- Returning Sports Item on Student ID
- Inventory Management (Add Item/Delete Item)
- View pass generated reports
- View fine generated for late returns

#### 3. Application Layer:

The application layer will handle the business logic and processing of the system. It will consist of the following components:

- Authentication module for handling user authentication and authorization

- Inventory management module for managing sports items, adding and deleting them.
- Item transaction module for handling item issue and return transactions
- Report generation module for generating various reports
- Calculate due module for calculating dues for overdue sports items

#### **4. Database Layer:**

The database layer will store all the data related to the system, including user information, item information, transaction information, and system settings. It will consist of the following components:

- User database for storing user authentication and authorization information
- Inventory database for storing sports item information
- Issue Return database for storing item issue and return transactions
- Student Database for storing the students' details

#### **5. Security:**

The system will have several security measures in place to protect against unauthorized access and data breaches. This will include:

- Secure login using encryption and hashing techniques
- Role-based access control to restrict access to sensitive information
- Data encryption for sensitive data
- Regular data backups to prevent data loss in case of system failure or breach
- Regular security audits and vulnerability testing to identify and address potential threats



**6. Scalability:**

The system will be designed to handle a large number of concurrent users and high volumes of data. This will include:

- Horizontal scaling using load balancers to distribute traffic
- Vertical scaling using cloud-based infrastructure to increase computing power and storage capacity
- Database sharing to distribute data across multiple database instances

**7. Performance:**

The system will be designed to meet the performance requirements specified in the non-functional requirements. This will include:

- Efficient algorithms and data structures to optimize system performance
- Caching mechanisms to reduce database queries
- Asynchronous processing to handle long-running tasks without blocking the user interface

**8. Usability:**

The system will be designed with usability in mind, to ensure that users can easily navigate and use the system. This will include:

- User-friendly interface design with clear and concise labels and instructions
- Consistent design elements and navigation patterns
- Help documentation and tooltips to assist users with system usage

Overall, the system design will be scalable, secure, and user-friendly, meeting the functional and non-functional requirements of the sports equipment management system.

## **5.2 Data Design**

Data design for the Sports Inventory Management System will involve the following entities and their attributes:

### **1. Sports Items:**

This entity will store information about the sports items such as Item ID, Sports Category, Item Quantity and Current Status.

### **2. Users:**

This entity will store information about the users of the system such as Username, Email and Password.

### **3. Transaction of Issue and Return:**

This entity will store information about the issue/return made by the users such as Student ID, Item ID, Sports Name, Status, Date and Time.

### **6. Student:**

This entity will store information about the student such as Student ID, Name, Course, Year and Due.

The relationships between these entities are as follows:

- Each item can belong to one category and each category can have multiple items.
- Each issue and return transaction can be associated with one item and one user.
- Each user can have multiple transactions.

The data design will also include primary keys, foreign keys, and indexes to ensure data integrity and efficient data retrieval.

## 6. Coding

### Login Page

1. GET Name  
    GET Email  
    GET Password
2. if credentials valid
3. Login into Homepage of SMS
4. else:
5. "Invalid Credentials Error"
6. Go to Step 2

### Issue Item

1. Get Student ID  
    Get Item ID  
    Get Sports name
2. if student is present in the database
3.     if item in inventory is available
4.         if dues on Student == 0
5.             Issued successfully
6.     else Clear the Due and Return the Item
7.     else Item is not available

8. else Student Data doesn't exist

## Return Item

1. Get Student ID  
    Get Item ID  
    Get Sports Name
2. if student is present in the database
3.     if item in inventory is available
4.         if dues on Student == 0
5.             Return successfully
6.     else Run Calculating Dues Module
7.         if Due Clearance button  
            Go to Step 3
8.     else Not Returned
9.     else Item is not available
10. else Student Data doesn't exist.

## Update Inventory (Add Item)

1. Get Item ID  
    Get Sports Name  
    Get Quantity
2. If Item ID exists
3.  $\text{ItemID.quantity} = \text{ItemID.quantity} + \text{Quantity}$

4. else New Record with Item.ItemID = ItemID and ItemID.Sportsname = Sportsname and ItemID.quantity = Quantity will be added into the database

## Update Inventory (Delete Item)

1. Get Item ID  
  
    Get Sports Name  
  
    Get Quantity
2. If Item ID exists
3. If Quantity < ItemID.quantity
4.     ItemID.quantity = ItemID.quantity – Quantity
5. If Quantity < ItemID.quantity
6.     Delete the whole record of Item in database table
7. If Quantity < ItemID.quantity
8.     Invalid Quantity
9. else Item doesn't exist

## Report Generation

1. Fetching the data from Student Info and Issue Return Info database

Accessing the view of SID, Name, Course, Year, ItemID, Sports name, Status, Date and Time.

## Calculating Due

1. Open the calculated Due with 2 buttons: Clear Due & Return and Back
2. If Clear Due & Return get clicked
3. Item Returned Successfully
4. If back get clicked
5. Item Not returned and Dues remains.

STOP

## 7. Testing

### A. Login Page

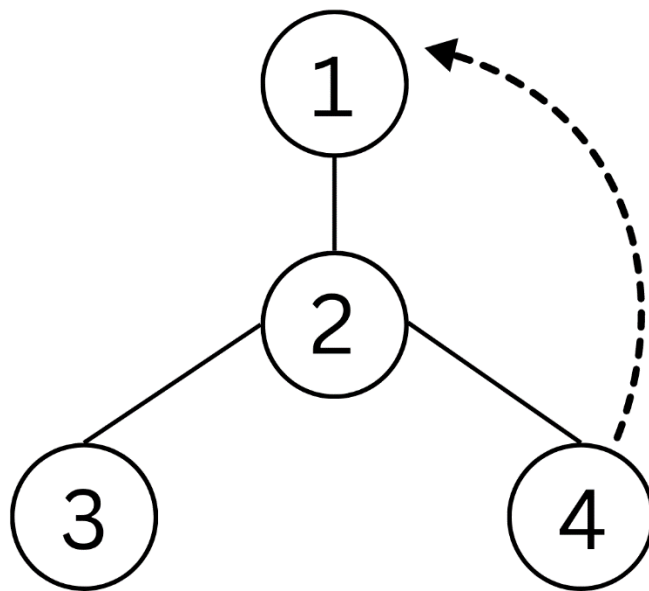
SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of login page with valid login details	Execution History:
Pre-Conditions: User's data should be present in database	Result: Successfully Logged in Homepage
Input: Login Details	If fails any possible Reason: None
Expected Output: Successfully Logged in Homepage	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Arun Chandra
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of login page with invalid login details	Execution History:
Pre-Conditions: User's data should be present in database	Result: "Invalid Credentials" Messaged Displayed
Input: Login Details	If fails any possible Reason: None



Expected Output: “Invalid Credentials” Messaged Displayed	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Arun Chandra
Date: 01-04-2023	Date: 01-04-2023

Flow Graph:



$$V(G) = E - N + 2P$$

Where: - E = edges; N = nodes; P = connected components

$$\text{Cyclomatic complexity} = 4 - 4 + 2(1)$$

$$V(G) = 2$$

Paths: P1 => 1→2→3

P2 => 1→2→4→1

## B. Issue Item

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of issue item page with valid field details with no dues on student	Execution History:
Pre-Conditions: Student Data should exist in database and no dues on student and Item should be available in Inventory and Student doesn't have currently issued any item.	Result: "CBAT Item Issued Successfully" Message displayed.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "<ItemName> Item Issued Successfully" Message displayed.	Any other observation: None
Post Conditions: Details of Issued Items and Student ID with data, time etc will be inserted into database.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of issue item page with invalid Student details with no dues on student	Execution History:
Pre-Conditions: Student Data shouldn't exist in database and no dues on student and Item should be available in Inventory and Student doesn't have currently issued any item.	Result: "Student Info doesn't exist" message displayed.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Student Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of issue item page with invalid Item details with no dues on student	Execution History:
Pre-Conditions: Student Data should exist in database and no dues on student	Result: "Item Info doesn't exist" message displayed.

and Item info doesn't exist in database and Student doesn't have currently issued any item.	
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Item Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of issue item page with valid field details with some dues on student	Execution History:
Pre-Conditions: Student Data should exist in database and there are dues present on student and Item info exist in database.	Result: As expected.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Clear Due and Return the Item <itemid>" message will be displayed.	Any other observation: None

Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

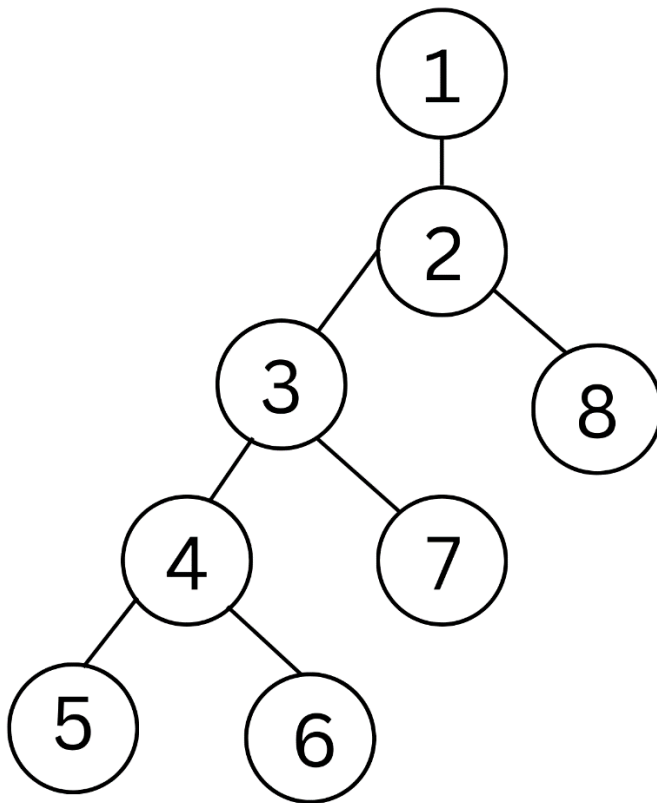
SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of issue item page with invalid field details with some dues on student.	Execution History:
Pre-Conditions: Either Student Data shouldn't exist in database or Item info doesn't exist in database or both, and there are dues present on student.	Result: As expected.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Item Info / Student Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of issue item page with valid field details with no dues when item is already issued.	Execution History:
Pre-Conditions: Student have no dues and Item should be already issued linked to Student ID Student Data should exist in database and Item info exist in database.	Result: As expected.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: “<itemname> Item Already Issued on this ID” message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of issue item page with invalid field details with no dues when item is already issued.	Execution History:

Pre-Conditions: Student have no dues and Item should be already issued linked to Student ID and Either Student Data shouldn't exist in database or Item info doesn't exist in database or both.	Result: As expected.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Item Info / Student Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

Flow Graph:



$$V(G) = E - N + 2P$$

Where: - E = edges; N = nodes; P= connected components

$$\text{Cyclomatic complexity} = 7 - 8 + 2(1)$$

$$V(G) = 1$$

Paths: P1  $\Rightarrow 1 \rightarrow 2 \rightarrow 8$

P2  $\Rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 7$

P3  $\Rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

P4  $\Rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$



## C. Return Item

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of return item page with valid field details with no dues on student	Execution History:
Pre-Conditions: Student Data should exist in database and no dues on student and Item should be available in Inventory and Student have currently issued an item.	Result: “CBAT Item Returned Successfully” Message displayed.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: “<ItemName> Item Returned Successfully” Message displayed.	Any other observation: None
Post Conditions: Details of Returned Items and Student ID with data, time etc will be inserted into database.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1	SEC 2
-------	-------

BEFORE EXECUTION	AFTER EXECUTION
Purpose: To test the functionality of Return item page with invalid Student details with no dues on student.	Execution History:
Pre-Conditions: Student Data shouldn't exist in database and no dues on student and Item should/should not be available in Inventory and Student have currently issued an item.	Result: "Student Info doesn't exist" message displayed.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Student Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of return item page with invalid Item details with no dues on student	Execution History:
Pre-Conditions: Student Data should/ should not exist in database and no	Result: "Item Info doesn't exist" message displayed.

dues on student and Item info doesn't exist in database and Student have currently issued an item.	
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Item Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of return item page with valid field details with some dues on student	Execution History:
Pre-Conditions: Student Data should exist in database and there are dues present on student and Item info exist in database.	Result: As expected.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: Calculating Due Dialog Box will be popped up.	Any other observation: None
Post Conditions: None	Any Suggestion: None

Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

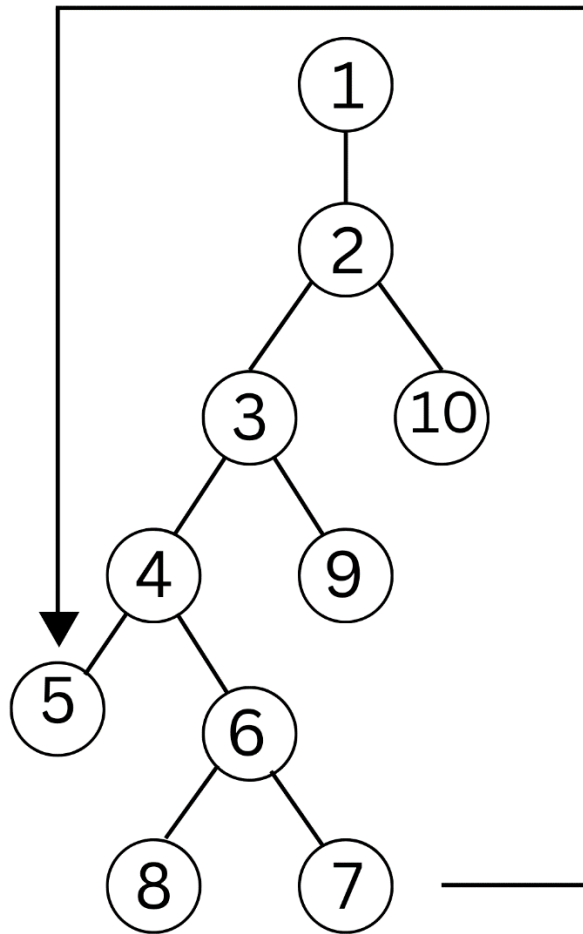
SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of return item page with invalid field details with some dues on student.	Execution History:
Pre-Conditions: Either Student Data shouldn't exist in database or Item info doesn't exist in database or both, and there are dues present on student.	Result: As expected.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Item Info / Student Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of return item page with valid field details with no dues when item is not issued.	Execution History:
Pre-Conditions: Student have no dues and No Item should be already issued linked to Student ID, Student Data should exist in database and Item info exist in database.	Result: As expected.
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: “No Item Issued on this ID” message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of return item page with invalid field details with no dues when no item issued on the id.	Execution History:
Pre-Conditions: Student have no dues and No Item should be already issued linked to Student ID and Either Student	Result: As expected.

Data shouldn't exist in database or Item info doesn't exist in database or both.	
Input: Student ID, Item ID, Sports Name	If fails any possible Reason: None
Expected Output: "Item Info / Student Info doesn't exist" message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

Flow Graph:



$$V(G) = E - N + 2P$$

Where: - E = edges; N = nodes; P= connected components

$$\text{Cyclomatic complexity} = 10 - 10 + 2(1)$$

$$V(G) = 2$$

Paths: P1 => 1→2→10

P2 => 1→2→3→9

P3=> 1→2→3→4→6→7→5

P4=> 1→2→3→4→6→8

P5=> 1→2→3→4→5

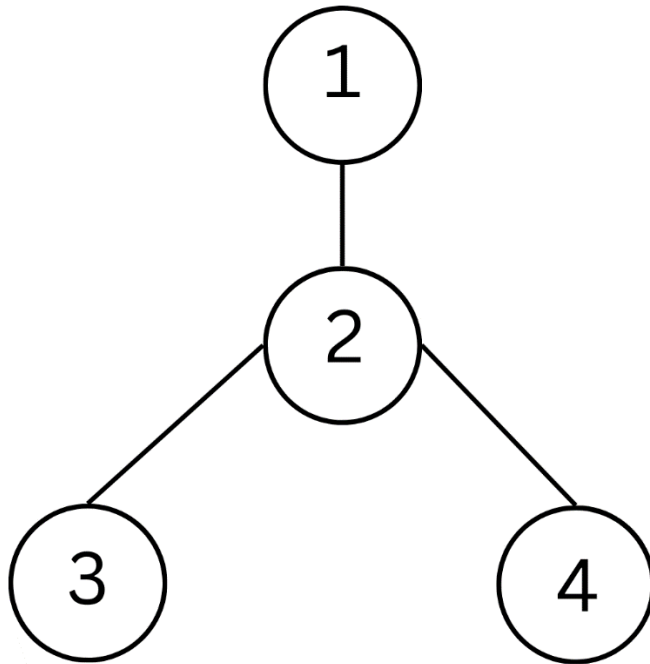
#### D. Update Inventory (Add Item)

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of Add Item page when adding new item.	Execution History:
Pre-Conditions: Item details shouldn't exist already in the Inventory Database Table.	Result: As expected.
Input: Item ID, Sports Name, Quantity	If fails any possible Reason: None
Expected Output: "New Item <Item ID> with quantity <Quantity> added successfully under <Sports Name>" message will be displayed.	Any other observation: None
Post Conditions: The Item ID, Sports name and the Quantity record will be inserted to the Inventory Database Table.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023



SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of Add Item page when adding quantity of existing item.	Execution History:
Pre-Conditions: Item details should exist already in the Inventory Database Table.	Result: As expected.
Input: Item ID, Sports Name, Quantity	If fails any possible Reason: None
Expected Output: “<Quantity> <Item ID> Item added successfully under <Sports Name>” message will be displayed.	Any other observation: None
Post Conditions: The Quantity of the existing it will be increased in the respective database table by the given input in Quantity Field.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

Flow Graph:



$$V(G) = E - N + 2P$$

Where: - E = edges; N = nodes; P= connected components

$$\text{Cyclomatic complexity} = 3 - 4 + 2(1)$$

$$V(G) = 1$$

Paths: P1 => 1→2→3

P2 => 1→2→4

### E. Update Inventory (Delete Item)

SEC 1	SEC 2
-------	-------

BEFORE EXECUTION	AFTER EXECUTION
Purpose: To test the functionality of Delete Item page when deleting quantity of existing item and the valid details are given.	Execution History:
Pre-Conditions: Item details should exist already in the Inventory Database Table and the valid details should be entered.	Result: As expected.
Input: Item ID, Sports Name, Quantity	If fails any possible Reason: None
Expected Output: “<Quantity> <Item ID> Item deleted successfully under <Sports Name>” message will be displayed.	Any other observation: None
Post Conditions: The Quantity of the existing it will be decreased in the respective database table by the given input in Quantity Field.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

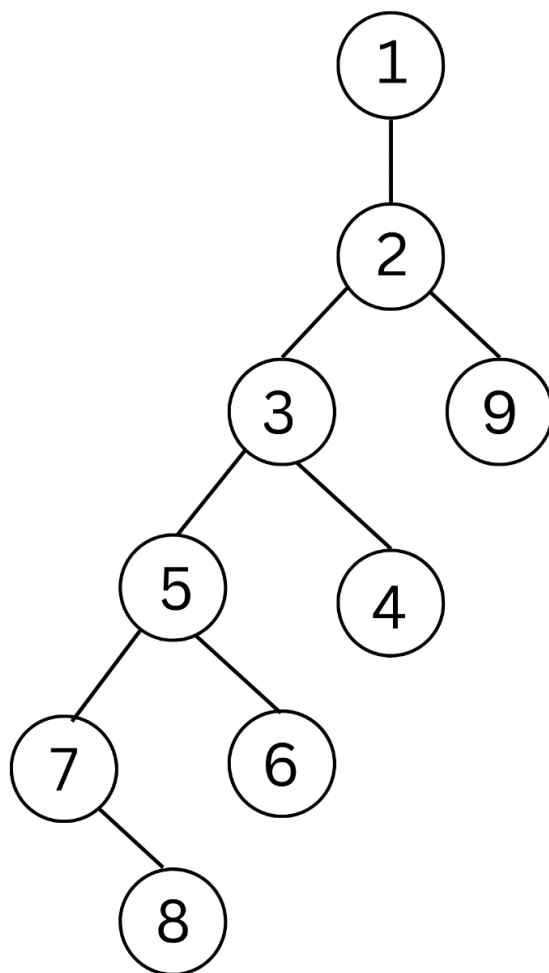
SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of Delete Item page when deleting quantity of existing item and the invalid details are given.	Execution History:

Pre-Conditions: Item details should exist already in the Inventory Database Table and the invalid details should be entered.	Result: As expected.
Input: Item ID, Sports Name, Quantity	If fails any possible Reason: None
Expected Output: "No item exists to delete." message will be displayed.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of Delete Item page when deleting the whole item.	Execution History:
Pre-Conditions: Item details should exist already in the Inventory Database Table and the quantity given for the item should be equal to the available quantity of the Item.	Result: As expected.
Input: Item ID, Sports Name, Quantity	If fails any possible Reason: None

Expected Output: “Item Deleted Successfully” message will be displayed.	Any other observation: None
Post Conditions: The whole record of the item will be deleted from the database table.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

Flow Graph:



$$V(G) = E - N + 2P$$

Where: - E = edges; N = nodes; P= connected components

$$\text{Cyclomatic complexity} = 8 - 9 + 2(1)$$

$$V(G) = 1$$

Paths: P1 => 1→2→9

P2 => 1→2→3→4

P3=> 1→2→3→5→7→8

P4=> 1→2→3→5→6

## F. Report Generation

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of report generation item page.	Execution History:
Pre-Conditions: None	Result: As expected.
Input: None	If fails any possible Reason: None
Expected Output: Fields from Student Database Table as well Issue Return Database Table like SID, Name, Year, Item ID, Sports name, Status, Date, Time will be displayed in tabular manner.	Any other observation: None
Post Conditions: None	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

Flow Graph:



$$V(G) = E - N + 2P$$

Where: - E = edges; N = nodes; P= connected components

$$\text{Cyclomatic complexity} = 0 - 1 + 2(1)$$

$$V(G) = 1$$

Paths: P1 => 1

## G. Calculating Due

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of calculating due dialog box page if a person clicks on "Clear Due & Return" Button.	Execution History:
Pre-Conditions: Item should already be issued and there are some dues present on the Student ID.	Result: As expected.
Input: <Clear Due & Return> Button or <Back>	If fails any possible Reason: None

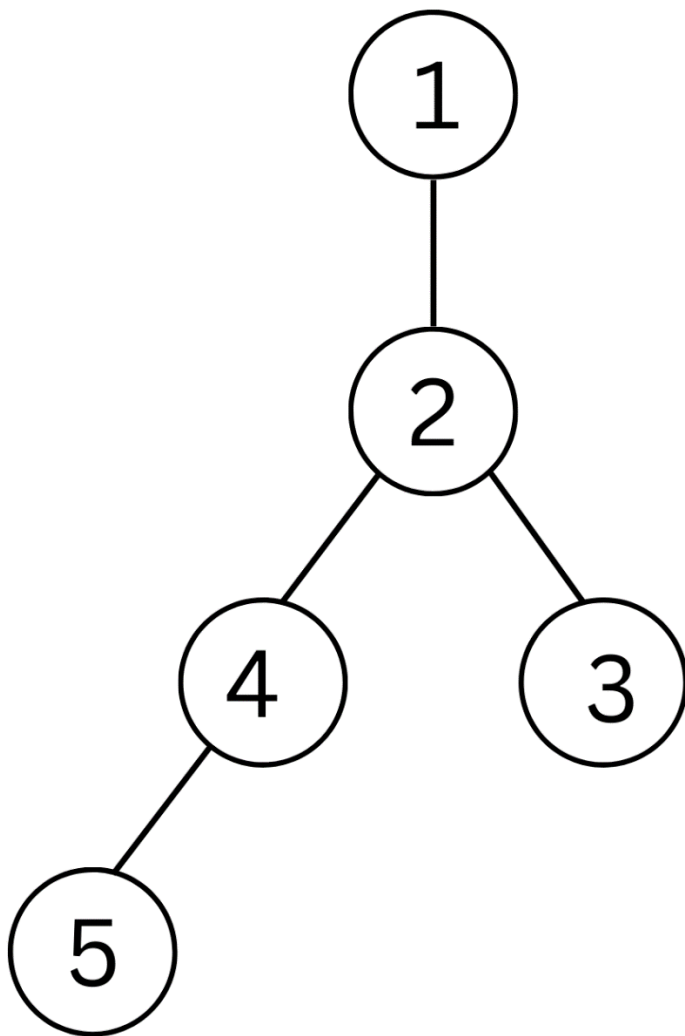
Expected Output: “<ItemName> Item Returned Successfully” Message displayed.	Any other observation: None
Post Conditions: Details of Returned Items and Student ID with data, time etc will be inserted into database.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman
Date: 01-04-2023	Date: 01-04-2023

SEC 1 BEFORE EXECUTION	SEC 2 AFTER EXECUTION
Purpose: To test the functionality of calculating due dialog box page if a person clicks on back button.	Execution History:
Pre-Conditions: Item should already be issued and there are some dues present on the Student ID.	Result: As expected.
Input: <Clear Due & Return> Button or <Back>	If fails any possible Reason: None
Expected Output: Back to the Return Page.	Any other observation: None
Post Conditions: The Item will not be returned and the dues still remains in the Student ID.	Any Suggestion: None
Written By: Arun Chandra	Run By: Nitin Dhiman



Date: 01-04-2023	Date: 01-04-2023
------------------	------------------

Flow Graph:



$$V(G) = E - N + 2P$$

Where: - E = edges; N = nodes; P= connected components

$$\text{Cyclomatic complexity} = 4 - 5 + 2(1)$$

$$V(G) = 1$$

Paths: P1  $\Rightarrow 1 \rightarrow 2 \rightarrow 3$

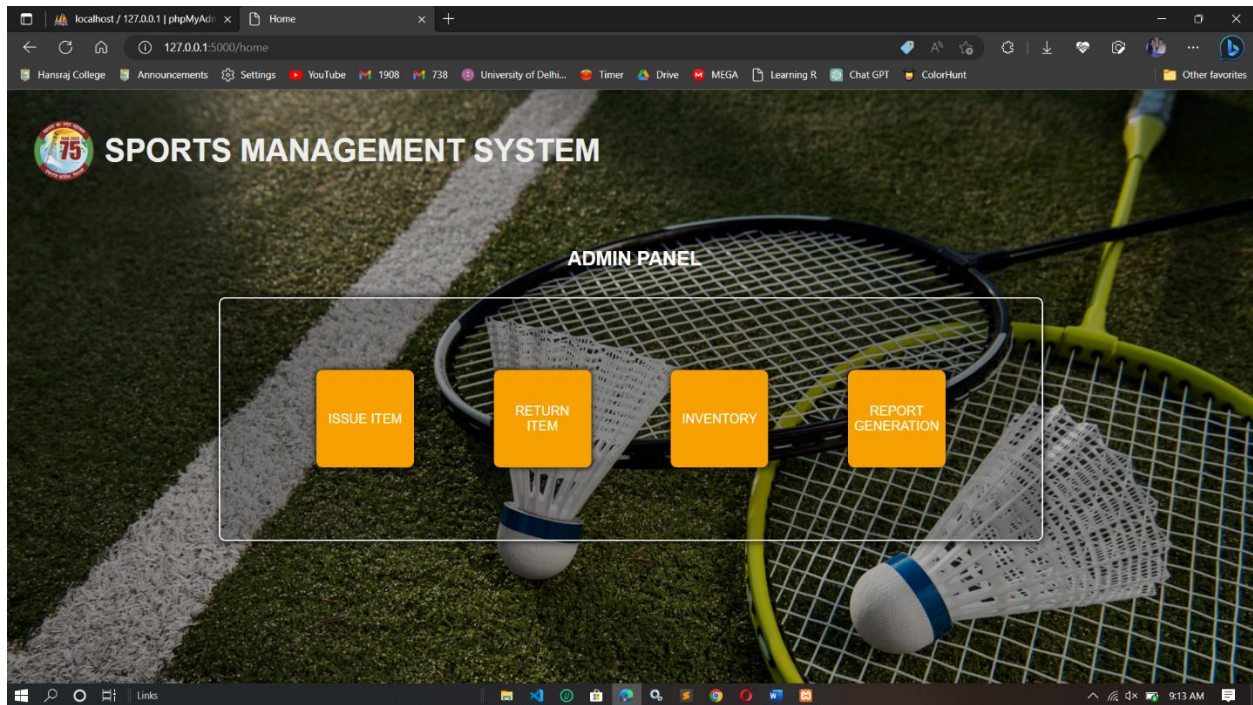
P2  $\Rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5$

## 8. References

- [1] Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach. 8th edition. McGraw-Hill.
- [2] Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition. New Age International Publishers.

## 9. Appendix

a.



b.

SID	Name	Course	Year	Item ID	Sports Name	Status	Date	Time
3472	ARUN CHANDRA	B.Sc. Hons CS	2	TBAT	TENNIS	RETURNED	2023-04-29	21:51:39
3472	ARUN CHANDRA	B.Sc. Hons CS	2	TBAT	TENNIS	ISSUED	2023-04-29	21:49:51
3466	NITIN DHIMAN	B.Sc. Hons CS	2	BBALL	BASEBALL	RETURNED	2023-04-10	13:26:31
3466	NITIN DHIMAN	B.Sc. Hons CS	2	BBALL	BASEBALL	ISSUED	2023-04-10	13:26:17

c.

**SPORTS MANAGEMENT SYSTEM**

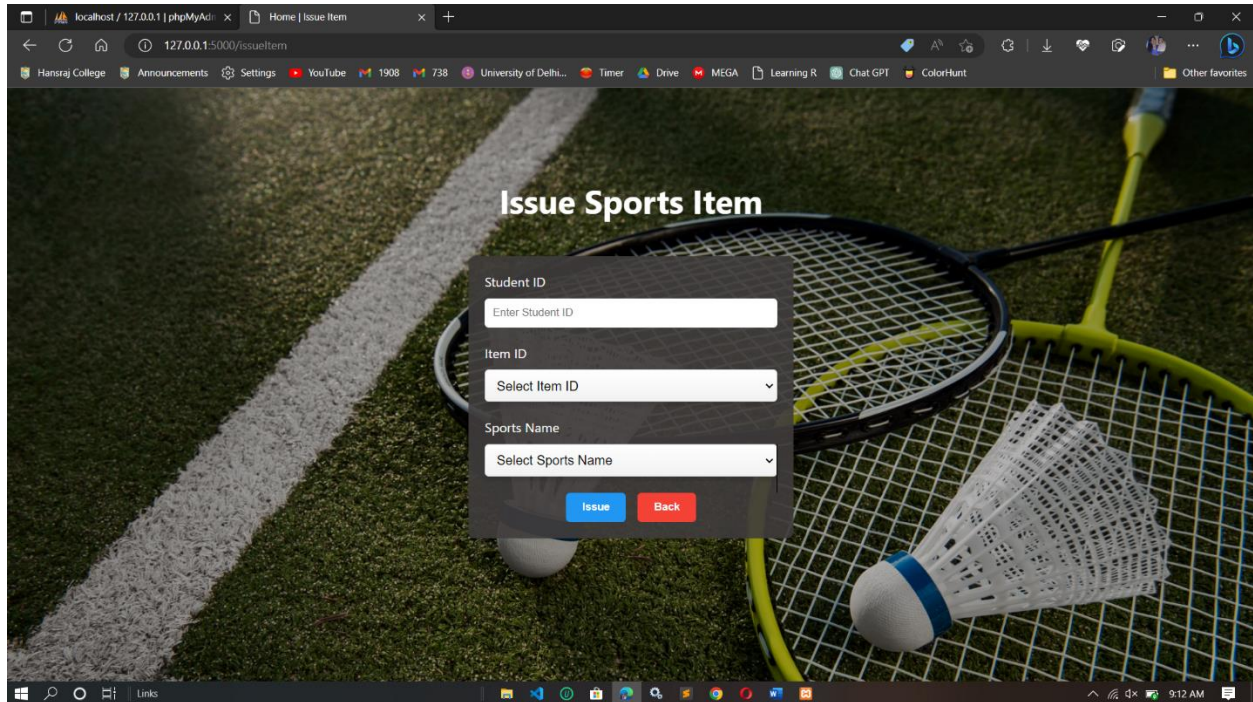
Name:

Email:

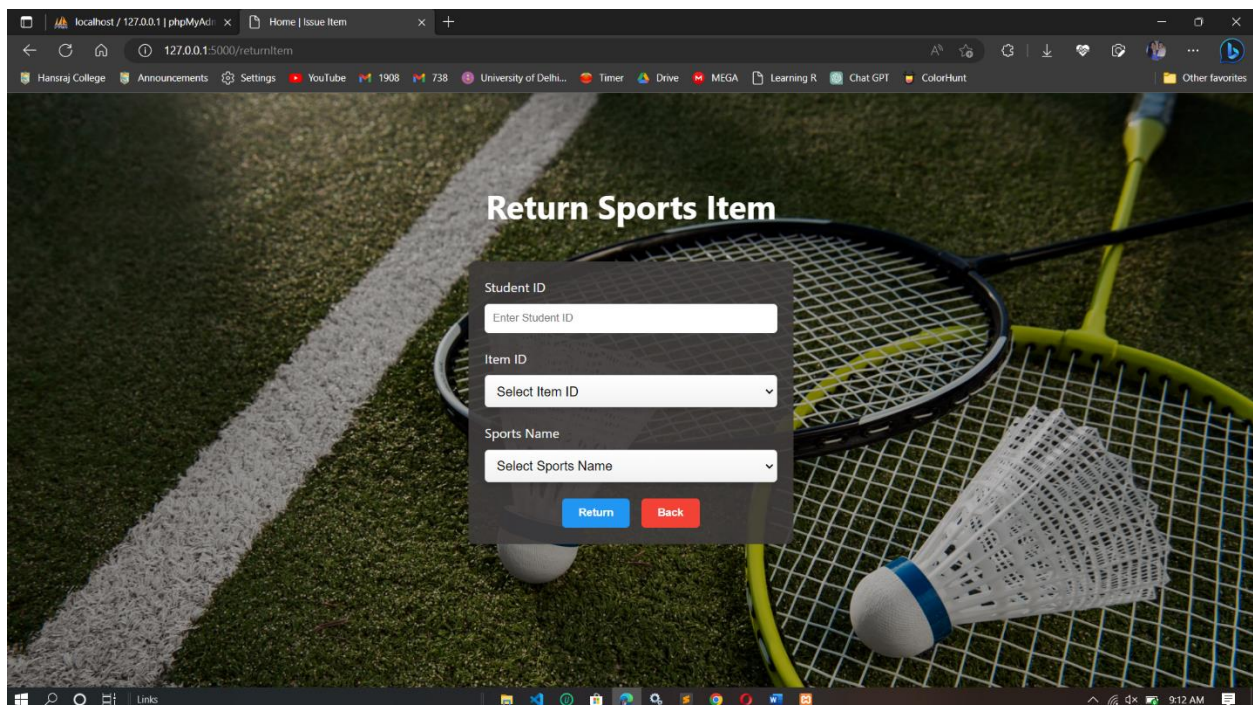
Password:



d.



f.



g.

