Club Management System

UCS 503
Software Engineering

Submitted by

Rohan Goyal	101815118
Utkarsh Babbar	101995007
Jatin Aggarwal	101815123
Sudhanshu Mahajan	101815125
Tarun Walia	101815073

B.E Third Year, Electronics & Computer Engineering

Submitted to

Dr. Amit Kumar Trivedi

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

TABLE OF CONTENTS

Contents **Page No.**

1. PROJECT OVERVIEW

1.1 PROJECT SCOPE

This project is a major step in connecting societies with the management, requests made will directly go to the different departments within a click and will hence reduce the time taken to both make a request and approve it. This will provide more transparency than the existing in-person management technique as every request made will be stored in database and be backed up in cloud. The dashboard will be displaying the current status of application in real-time, and will be updated as the database fetch results from the feedback cum approval form that will be mailed to the management bodies, directly from the system itself. The backend is made in Django, a fast, secure, and scalable web framework.

1.2 PROJECT FEATURES

1. Funding request-: In this feature the request is being posted by the club member to the specific authorities in order to get the funding from the management for the event being hosted by the club. All the necessary information i.e.

Name of the club

Amount of fund requested

Event for which the funding is required

is being added by the end user. The above information is then being sent to the authorities and a popup will be reflect in their email regarding the request received and the decision of accepting or rejecting the proposal is being done by authorities.

2. Permission request -: In this feature the request is being posted by the club member to the specific authorities in order to get the permission for the room/auditorium for hosting their specific event .The details of the event is being attached and posted. Hence the concerned authorities will receive the request for the permission and hence after analyzing the event details takes the decision of accepting or declining the request.

3. Status -: This feature allows the end user to check the status of the request being posted i.e. whether the proposal posted is being accepted by the concerned authorities or not.

Also if the request is being accepted by the authorities then, the user can download the acceptance letter from there which can be further shown for the verification purposes and the user will receive mail of proposal acceptance on their registered mail id.

In case the request is declined by the authorities then no letter is being generated and the user will

receives a mail of proposal being rejected.

4. Feedback-: This feature allows the user to provide to us with any suggestions or feedback after interacting with the software .This makes the software more interactive, since it's not just sending the different proposals to the authorities, since users can also contribute.

2. PROJECT REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

- Updation of data should be simple.
- Must provide the following features.
  a. Request Fund
  b. Request Permission
  c. Cancel Request
- User has to Login first before doing any operation. Also provide option for Signup.
- Provide option to block users.
- System should provide optional feature to upload photos of bills for Fund Request.
- System should support additional Text with every permission.

2.2 NON-FUNCTIONAL REQUIREMENTS

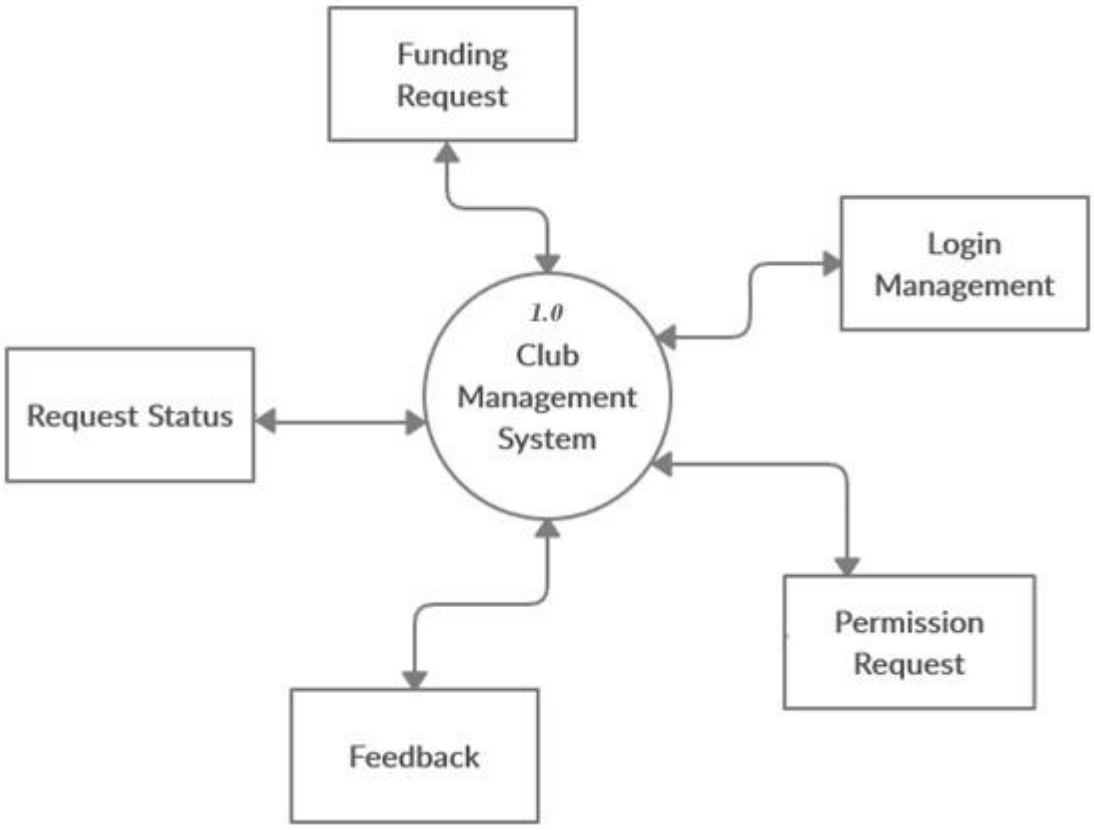
- Fast Response Time - There should not be much delay between the actions.
- Error Free - System should be bug free and must have any option to report bug, if found.
- Security- Data should be encapsulated from other users and system must be secure from the hackers.
- Concurrency - System should able to handle multiple users at same time.
- Simplicity - should be easy to use.
- Supportability - should be supportable to any type of devices (PCs, Mobiles, etc.)

3. STRUCTURED ANALYSIS

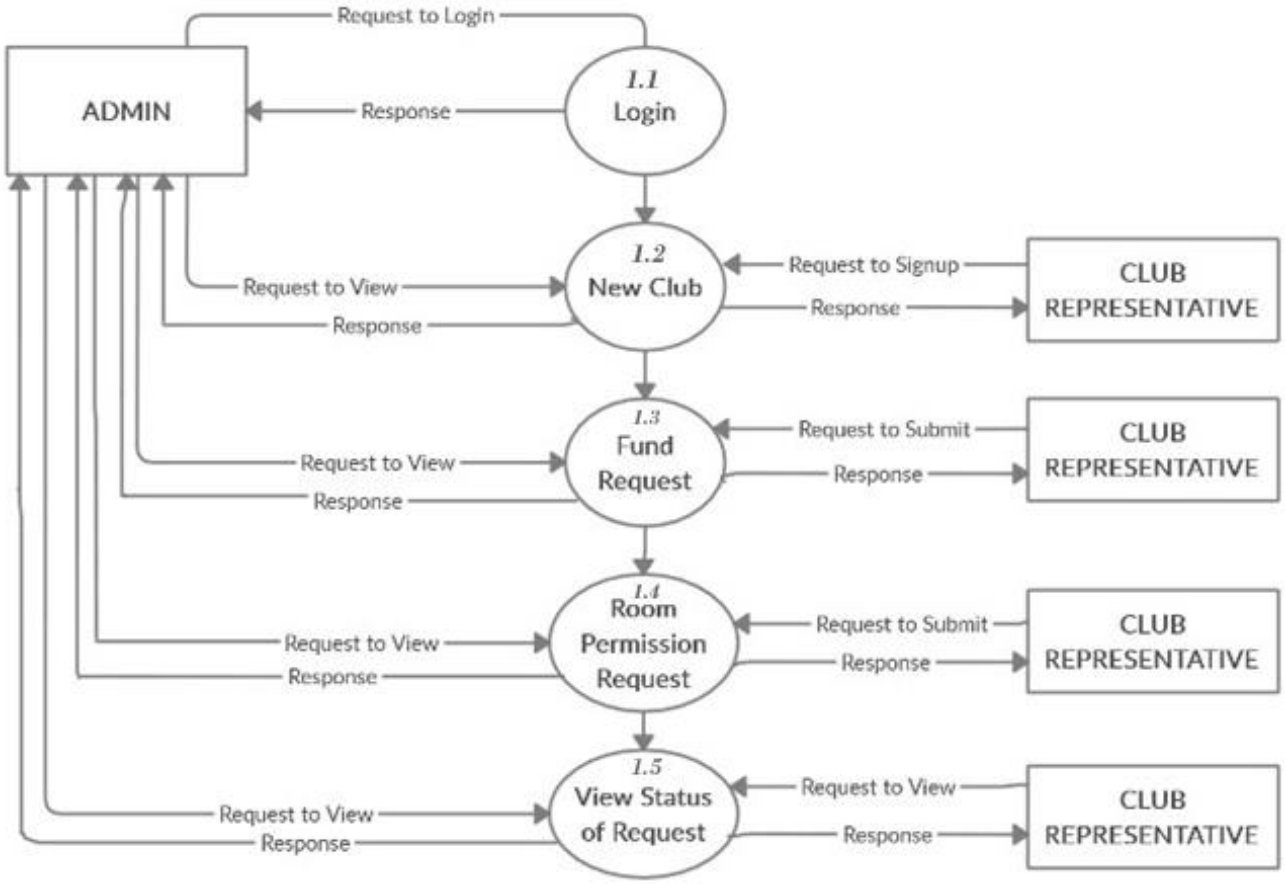
3.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. It depicts the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how the data enters and leaves the system, what changes the information, and where data is stored.
The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used to perform a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond.

3.1.1 DFD LEVEL 0



3.1.2 DFD LEVEL1



3.1.3 DATADICTIONARY

PROCESSES:

- Login: Using a unique Username and Password, users/admin can log into their accounts, provided they have created one before.
- Sign Up: Creating an account on the website by creating a unique Username and Password and filling in their basic information like Club Name, Email Address.
- Display: Front End Design of the website where all codes related to the user's interest will be displayed in one concise location.
- Search: Using the request ID, the user can look for the desired request initiated or cancelled by them.
- Request Forms: The club representatives can initiate/cancel permission or funding request through these forms on website.
- Check Status: The club representatives can check the current status of the request initiated by them.

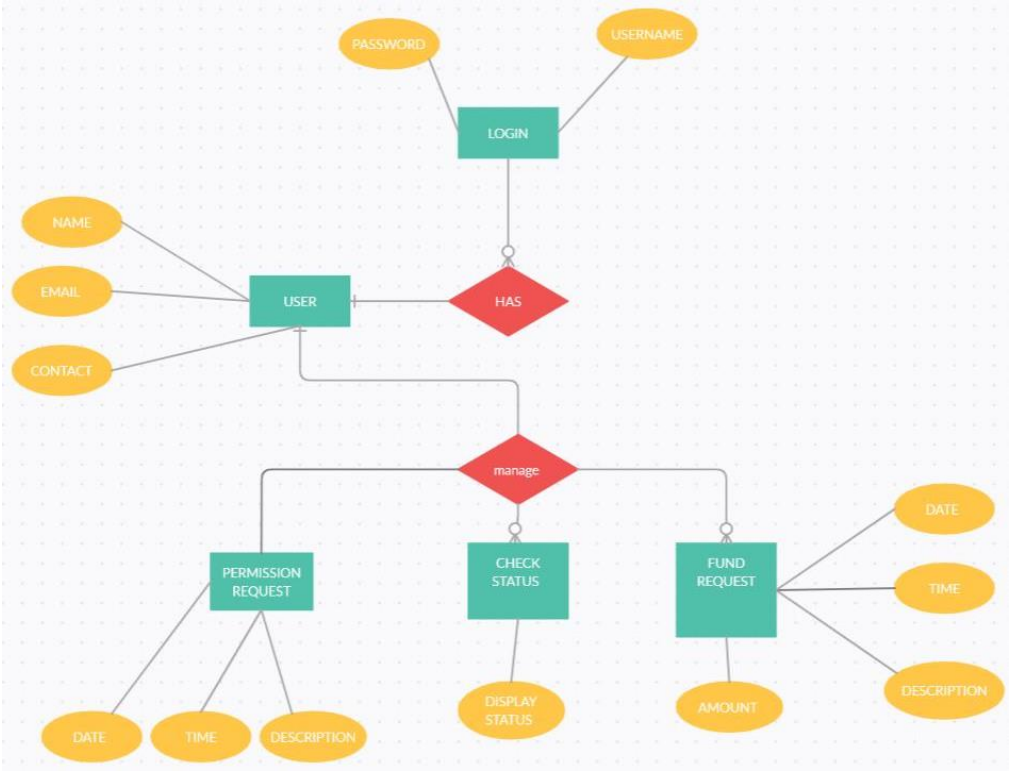
DATA STORES:

- User Database: Where information intended for the user access is stored like Club Name, Email Address, Contact Number, Username and Password.
- Admin Database: Where information intended for the admin access is stored like details of all clubs.

DATA FLOWS:

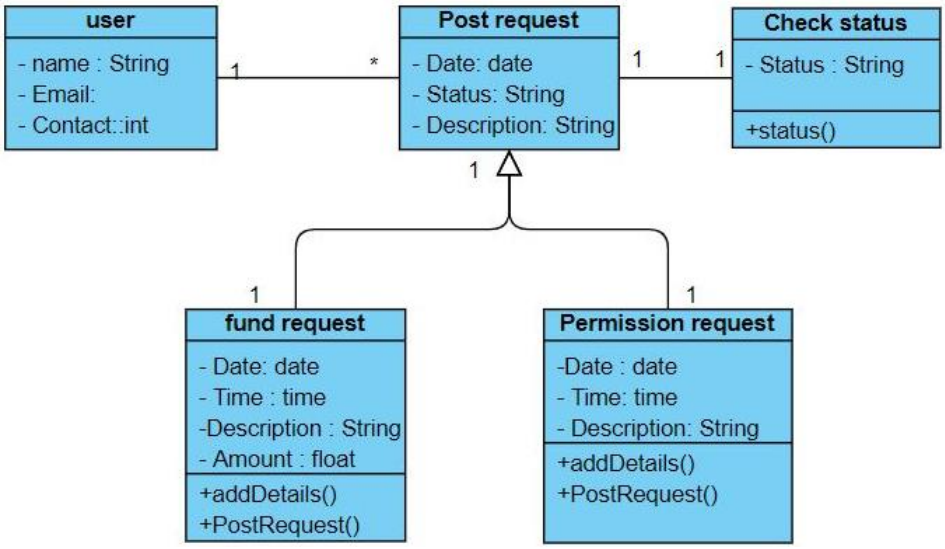
- Credentials: This includes Username and Password of the user for login.
- Verify Credentials: The credentials are sent to the database where they are matched with existing details, and if the details match access is granted to the user.
- Authorization: Granting of access to verified/authorized users.
- Club Profile: All information about the club that have been entered while signing up as club representative.

3.2 ER DIAGRAM



An Entity Relationship Diagram (ERD) shows the relationship of entity sets stored in a database. An entity in this context is an object, or a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

4. CLASS DIAGRAM

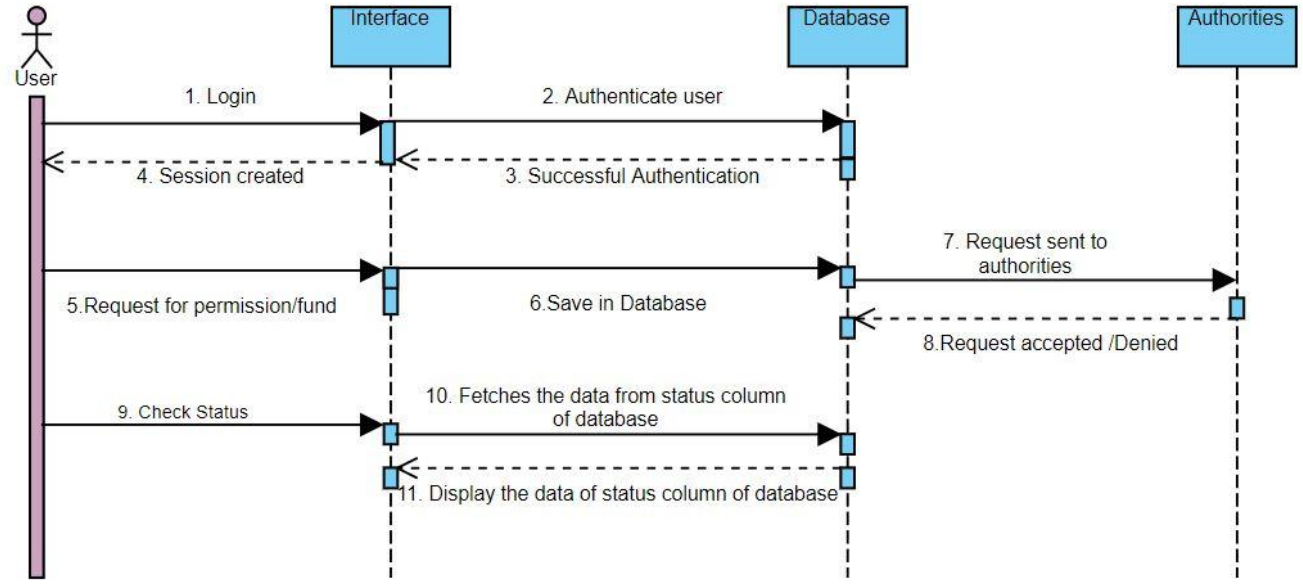


Class diagrams are the main building blocks of every object oriented methods. These can be used to show the classes, relationships, interface, association, and collaboration.
Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

The main purpose to use class diagrams are:

1. It is the only UML which can appropriately depict various aspects of OOPs concept.
2. Proper design and analysis of application can be faster and efficient.
3. It is base for deployment and component diagram.

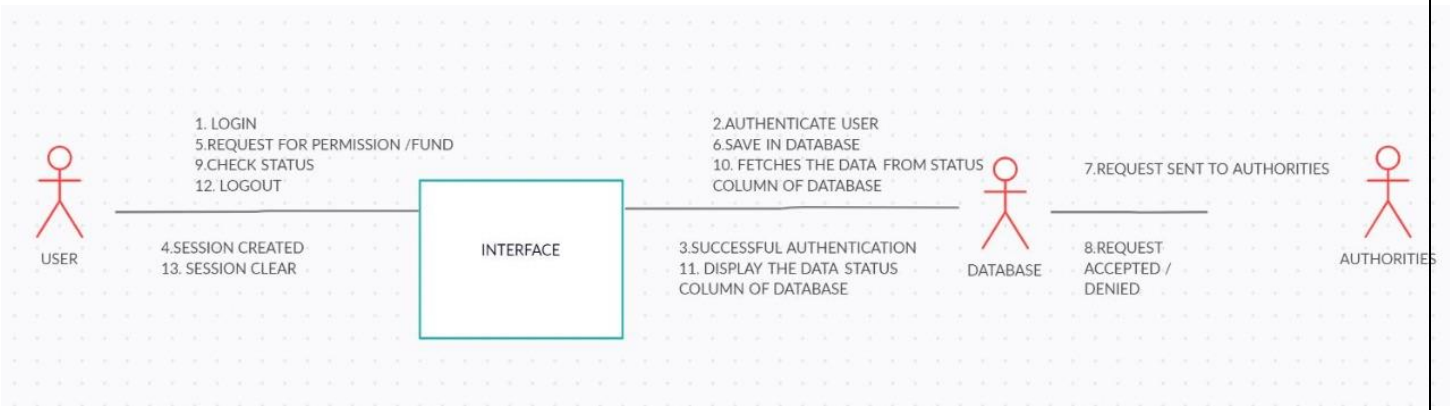
5. SEQUENCE DIAGRAM



A sequence diagram depicts the interaction between the objects in a sequential order i.e. the order in which these interactions take place.

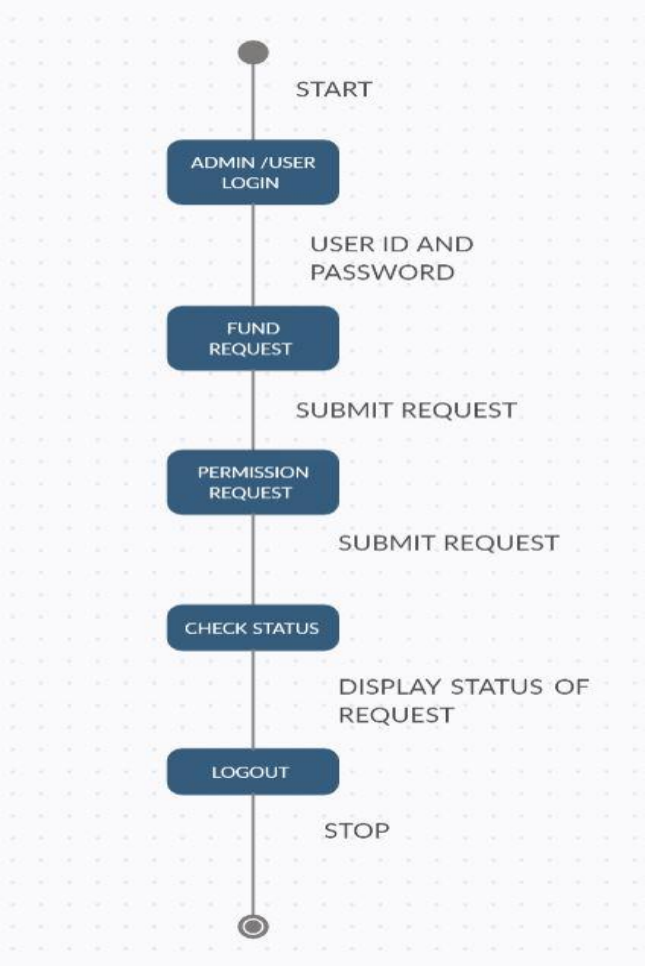
1. Firstly the user enters his /her login credentials which is then authenticated from the database, and upon successful authentication; a session of that user will be created.
2. Upon posting a request for Permission/Fund then all the details entered by the user in the request form shall be saved in the database and sent to authorities for approval. If the authorities either accept/deny, the request is then saved in database.
3. If the user clicks on the check status button then the data is being fetched from the database and displayed to the user.

6. COLLABORATION DIAGRAM



Collaboration Diagram represents the interaction of the objects to perform the behavior of a particular use case or a part of use case.
It is used to represent the structural organization of the system and the messages that are sent and received.

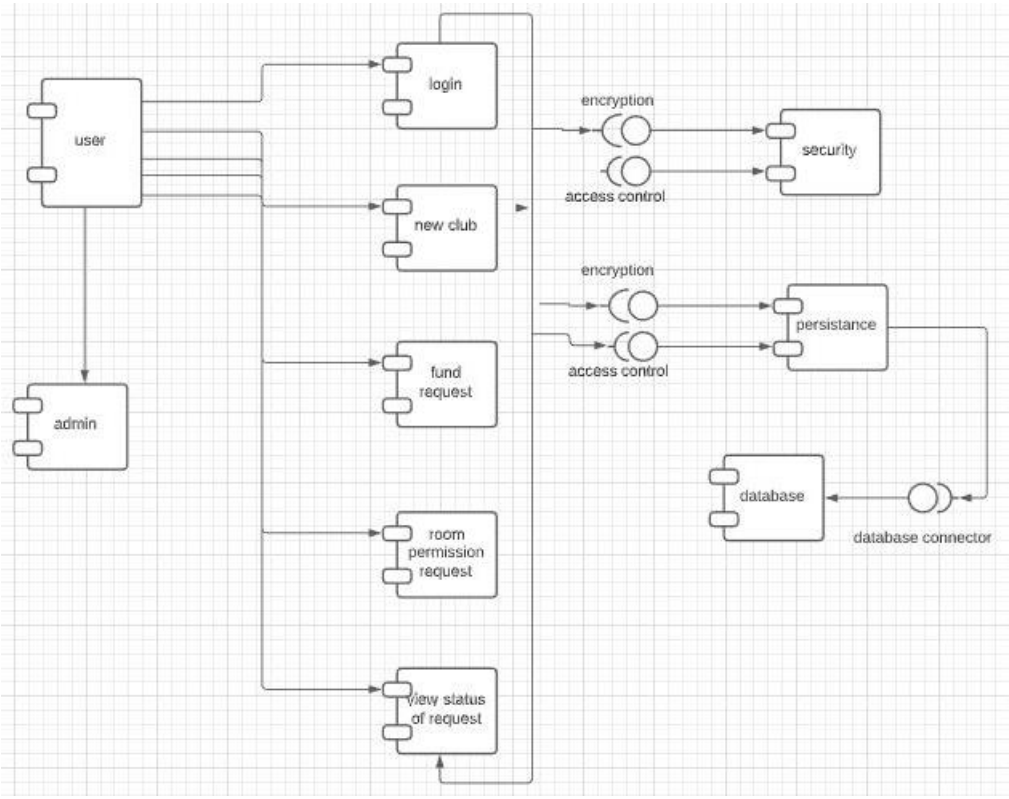
7. STATE CHART DIAGRAM



The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

The main purposes of using State Chart Diagrams are:
1. To model the dynamic aspect of a system.
2. To model the life time of a reactive system.
3. To describe different states of an object during its life time.
4. Define a state machine to model the states of an object.

8. COMPONENT DIAGRAM

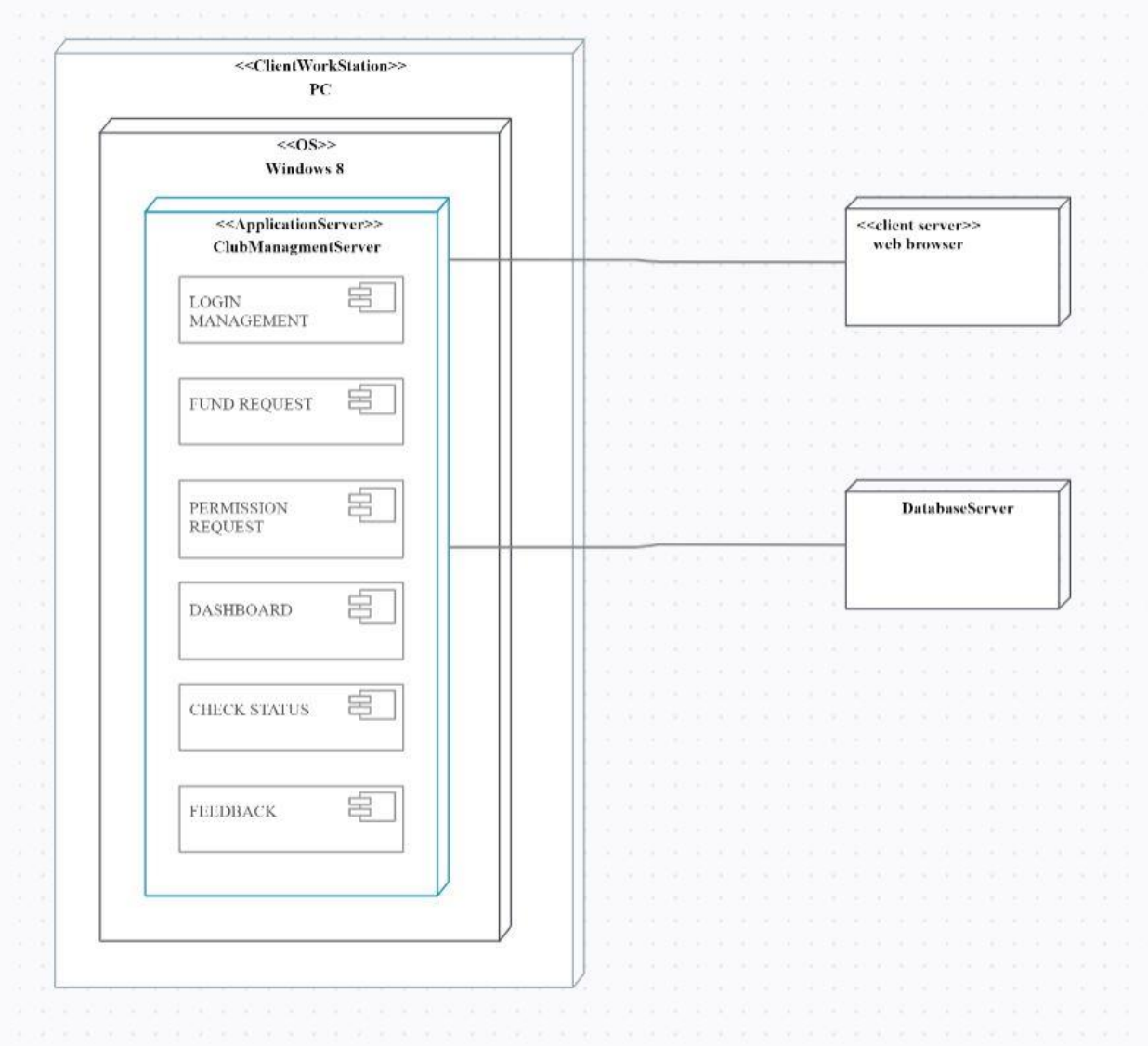


Component diagrams are different in terms of nature and behavior. They are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements

such as executables, libraries, files, documents, etc. which reside in a node.
Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far.
It does not describe the functionality of the system but it describes the components used to make those functionalities.

The purpose of the component diagram can be summarized as −
1. Visualize the components of a system.
2. Construct executables by using forward and reverse engineering.
3. Describe the organization and relationships of the components

9. DEPLOYMENT DIAGRAM

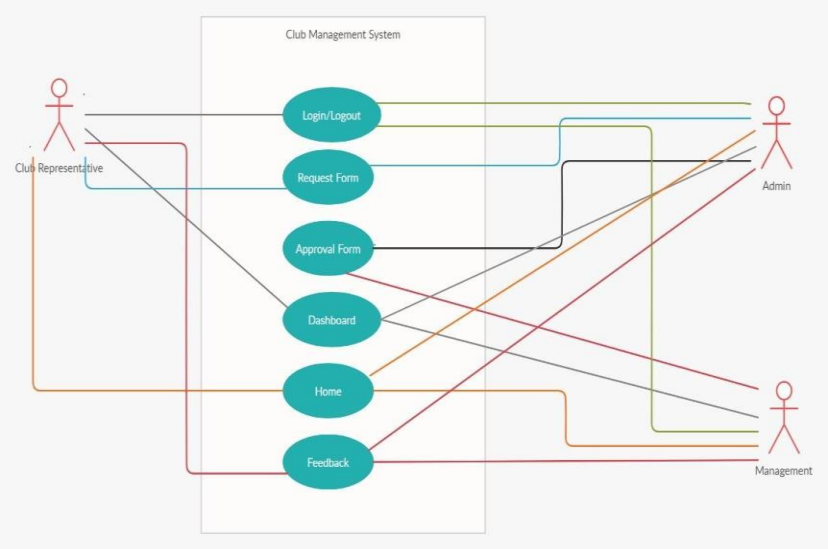


Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. The term Deployment itself describes the purpose of the diagram.

The purpose of deployment diagrams can be described as:
1. Visualize the hardware topology of a system.
2. Describe the hardware components used to deploy software components.
3. Describe the runtime processing nodes.

10. OBJECT ORIENTED ANALYSIS

10.1 Use Case Diagram



10.2 Use Case Template

ID:	1
Title:	Login/Logout
Description:	To login into the system and logout from the system
Preconditions:	None
Main Success Scenario:	User Logged In Successfully
Post Conditions:	User gets updated about all the previous requests
Frequency of Use:	Every time User Starts the website from an unauthenticated device.
Task Sequence:	Login Screen pops up if user isn't already authenticated Logout option is always present
Modification History:	Nov 1,2020

ID:	2
Title:	Request
Description:	To submit a request, into the system
Preconditions:	Logged in
Main Success Scenario:	Request Submitted Successfully
Post Conditions:	Request updated in database and a confirmation mail is sent
Frequency of Use:	Every time user click on request form button
Task Sequence:	1.Form pops up 2. User Fills the form and place request
Modification History:	Nov 1,2020

ID:	3
Title:	Approval
Description:	To give approval or rejection to already submitted requests
Preconditions:	Logged in as management
Main Success Scenario:	Approved or rejected displayed against the form
Post Conditions:	User gets updated about all the previous requests
Frequency of Use:	Every time management fill the approval form
Task Sequence:	1.Management fills the approval form 2. Hit submit 3. Database updates itself
Modification History:	Nov 1,2020

ID:	4
Title:	Dashboard
Description:	To display all the requests made by user and updates on those requests
Preconditions:	Logged in
Main Success Scenario:	All prior requests and their status is shown
Post Conditions:	User gets updated about all the previous requests
Frequency of Use:	Every time User Starts the website and login the page is displayed
Task Sequence:	1.Login Screen pops up if user isn't already authenticated 2.Dashboard button is clicked
Modification History:	Nov 1,2020

ID:	5
Title:	Home
Description:	Home page of CMS
Preconditions:	None
Main Success Scenario:	Opens without delay as soon as url is hit
Post Conditions:	All major links to CMS services are displayed here
Frequency of Use:	Every time User Starts the website
Task Sequence:	1.Login Screen pops up if user isn't already authenticated 2.Home page is displayed
Modification History:	Nov 1,2020

ID:	6
Title:	Feedback
Description:	To give feedback to developers about any possible bug or difficulty users may face
Preconditions:	Logged In
Main Success Scenario:	Feedback form submitted successfully
Post Conditions:	Mail sent to developers, according to responses received in feedback form
Frequency of Use:	Every time user is logged in and clicks on feedback button
Task Sequence:	1.Login Screen pops up if user isn't already authenticated 2.Feedback button is clicked 3.Form is filled 4.Mail is sent to developers
Modification History:	Nov 2,2020

11. TEST CASE REPORT

12.1 Login Page Test

Test Case #: 1	**Test Case Name:** Log In	**Page:** 1
System: Club Management System	**Subsystem:** Log In Page	
Designed By: 3ENC6_1_SOFTWARE	**Design Date:** 15/11/2020	
Executed By:	**Execution Date:**	
Short Description: Test the Login Page.		

Pre-Condition:

1. User must be connected to internet.
2. The Website must be opened.

Test ID	Action Value	Expected Result	Remarks (Pass/Fail)	Comments
1.	Enter Username = “ABC” and Password = “12346”.	Invalid Password!	Fail	Incorrect Username or Password entered.
2.	Enter Username = “ABC” and Password = “12345”.	Login Successful	Pass	Username and Password correctly entered.

Post Condition: User marked logged in in database.

12.2 Fund Request Test

Test Case #: 2	**Test Case Name:** Fund Request	**Page:** 2
System: Club Management System	**Subsystem:** Fund Request Page	
Designed By: 3ENC6_1_SOFTWARE	**Design Date:** 15/11/2020	
Executed By:	**Execution Date:**	
Short Description: Test the Fund Request Page.		

Pre-Condition:

1. User must be connected to internet.
2. The Website must be opened.

Test ID	Action Value	Expected Result	Remarks (Pass/Fail)	Comments
1.	Click the 'Fund Request' button.	The system displays a form asking the user to enter the necessary details like event details, estimated fund, etc. for which the fund is required.	Pass	Details submitted successfully.
2.	Click Submit Button.	The system displays a message that the request is successfully posted and directs to the Home page.	Pass	Fund Request posted successfully.

Post Condition: The request for fund along with the necessary details will be saved in the database.

12.3 Permission Request Test

Test Case #: 3	**Test Case Name:** Permission Request	**Page:** 3
System: Club Management System	**Subsystem:** Permission Request Page	
Designed By: 3ENC6_1_SOFTWARE	**Design Date:** 15/11/2020	
Executed By:	**Execution Date:**	
Short Description: Test the Permission Request Page.		

Pre-Condition:

1. User must be connected to internet.
2. The Website must be opened.

Test ID	Action Value	Expected Result	Remarks (Pass/Fail)	Comments
1.	Click the 'Permission Request' button.	The system displays a form asking the user to enter the necessary details like event details, room no., date and time, etc. for which the permission is required.	Pass	Details submitted successfully.
2.	Click Submit Button.	The system displays a message that the request is successfully posted and directs to the Home page.	Pass	Permission Request posted successfully.

Post Condition: The request for permission along with the necessary details will be saved in the database.

12.4 Check Status Test

Test Case #: 4	**Test Case Name:** Check Status	**Page:** 4
System: Club Management System	**Subsystem:** Check Status Page	
Designed By: 3ENC6_1_SOFTWARE	**Design Date:** 15/11/2020	
Executed By:	**Execution Date:**	
Short Description: Test the Check Status Page.		

Pre-Condition:

1. User must be connected to internet.
2. The Website must be opened.

Test ID	Action Value	Expected Result	Remarks (Pass/Fail)	Comments
1.	Click the 'Check Status' button.	The system displays the current status of the Fund/Permission Request being initiated by the user.	Pass	Status displayed successfully.

Post Condition: Current Status of the Fund/Permission Request is stored in the database.