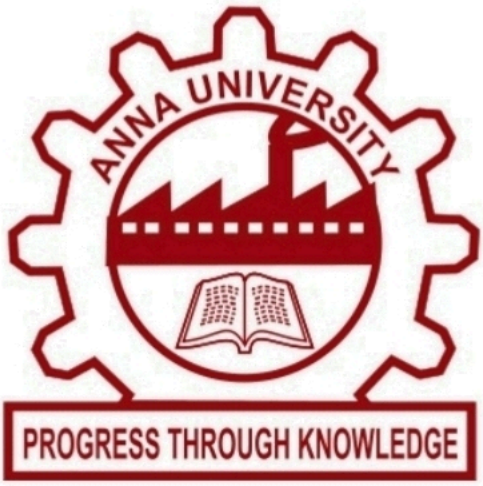
UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE)

KONAM, NAGERCOIL – 629 004



RECORD NOTE BOOK

CCS356-OBJECT ORIENTED SOFTWARE ENGINEERING

REGISTER NO :

NAME :

YEAR/SEMESTER :

DEPARTMENT :

UNIVERSITY COLLEGE OF ENGINEERINGNAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE)

KONAM, NAGERCOIL – 629 004

Register No:

Certified that, this is the bonafide record of work done by ***Mr./Ms.*** *…… of VI Semester in Computer Science and Engineering of this college, in the* ***CCS356 – OBJECT ORIENTED SOFTWARE ENGINEERING*** *during academic year 2023-2024 in partial fulfillment of the requirements of the B.E Degree course of the Anna University Chennai.*

Staff-in-charge **Head of the Department**

This record is submitted for the University Practical Examination held on .....

Internal Examiner **External Examiner**

INDEX

Exp No	Date	Title	Page	Sign
1.		Identify a software system that needs to be developed		
2.		Document the Software Requirements Specification (SRS) for the Conference Management System		
3.		Identify the use cases and the develop the Use Case model		
4.		Identify the conceptual classes and develop a Domain Model and also derive a Class diagram for the Conference Management System		
5.		Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration diagram		
6.		Draw relevant State Chart and Activity Diagram for the Conference Management System		
7.		Implement the system as per the detailed design		
8.		Test the software system for all the scenarios identified as per the usecase diagram		
9.		Improve the reusability and maintainability of the software system by applying appropriate design patterns		

EX NO :	IDENTIFY A SOFTWARE SYSTEM THAT NEEDS TO BE DEVELOPED
DATE :	

AIM :

To identify a software system that needs to be developed -Conference Management System.

INTRODUCTION :

A Conference Management System (CMS) is a software platform designed to streamline the organization, management, and execution of academic or professional conferences, symposiums, workshops, or similar events. It encompasses a range of functionalities to facilitate various aspects of conference planning, registration, communication, scheduling, and coordination.

FEATURES OF CONFERENCE MANAGEMENT SYSTEM :

1.Abstract and Paper submission :

- Allows authors to submit their abstracts, papers, or proposals online.
- Manages the submission process, including deadlines, formatting requirements, and submission categories.
- Provides tools for authors to track the status of their submissions and make revisions if necessary.

2.Review and Evaluation :

- Facilitates the peer-review process by assigning submissions to reviewers.
- Collects feedback, ratings, and evaluations from reviewers.
- Manages decision-making processes such as acceptance, rejection, or revision based on review outcomes.

3.Registration Management :

- Enables participants to register for the conference, select sessions, workshops, or events they wish to attend.
- Manages registration fees, discounts, and payment processing.
- Generates invoices, receipts, and badges for registered attendees.

4.Program and Schedule planning :

- Helps organizers create and manage the conference program, including keynote speeches, presentations, panels, and social events.
- Provides tools for scheduling sessions, assigning speakers, and updating program details.
- Allows attendees to view the conference schedule and create personalized agendas.

5.Participant communication :

- Facilitates communication between organizers, presenters, reviewers, and attendees.
- Sends announcements, notifications, reminders, and updates via email, SMS, or in-app messaging.
- Provides discussion forums, chat rooms, or social media integration for networking and collaboration.

6.Venue and Logistics Management :

- Assists in coordinating venue logistics, such as room assignments, equipment setup, catering services, and accommodation arrangements.
- Manages facility bookings, seating arrangements, transportation options, and onsite support services.

7.Financial Management :

- Tracks financial transactions related to conference registration, sponsorship, exhibitor fees, and other revenue streams.

- Generates financial reports, monitors budget allocations, and handles invoicing and payment processing.
- Integrates with accounting systems or payment gateways for secure transactions.

8.Data Analytics and Reporting :

- Provides analytical insights into conference metrics, attendee demographics, submission statistics, reviewer performance, and other relevant data.
- Generates customizable reports, charts, and graphs to assess the success of the conference and identify areas for improvement.

9.Accessibility and Security :

- Ensures accessibility compliance and data security measures to protect sensitive information, including personal data, financial transactions, and intellectual property rights.
- Implements user authentication, authorization, encryption, and data backup mechanisms to safeguard information.

10.Integration and Customization :

- Supports integration with external systems, such as academic databases, content management systems, or online collaboration platforms.
- Offers customization options to adapt to the specific needs and branding requirements of different conferences, including branding, themes, and configurations.

ARCHITECTURE OF CONFERENCE MANAGEMENT SYSTEM :

User Interface (UI):

- This is the front-end component that allows users to interact with the system.
- It includes web pages or mobile applications where users can register for conferences, submit papers, review papers, interact with other participants, etc.

Authentication and Authorization:

- This component manages user authentication and authorization.
- It ensures that only authorized users can access certain features or data within the system.

Database:

- The database component stores all the data related to conferences, users, submissions, reviews, schedules, etc.
- It should be capable of handling large volumes of data efficiently and securely.
- Commonly used databases include MySQL, PostgreSQL, MongoDB, etc.

RESULT :

Thus the software system that is need to be developed for Conference Management System was executed successfully.

EX NO :	DOCUMENT THE SOFTWARE REQUIREMENT SPECIFICATION FOR THE CONFERENCE MANAGEMENT SYSTEM
DATE :	

AIM :

To implement a software for Conference management system.

PROBLEM STATEMENT :

The Conference management system is basically designed to manage the details of the process that will be taken place in the conference in a place. It works along with the organizer , who arranges all these program and central management system, which consists of the all the details of the member who participates in the presentation. It is a system is web based software that supports organization of conference. It helps the paper presenters, conference organizers, authors &reviewers in their respective activities. The system maintains a website to which various members interested in attending conference register themselves as members by providing details, people who are willing to submit their paper to the system with the topic and domain. The conference management includes paper submission, reviewer assignments & register handling of conferences the system should be applicable to any technical conference which should be in format & standardized in html & can contain references.

SOFTWARE REQUIREMENT SPECIFICATION

Introduction:

The Conference Management System (COMS) provides administrative support to conference administrators by structuring and partially automating many of the workflows and management tasks that arise during the preparation

of a conference. Furthermore, functions to export statistics and details about delegates and their presentations support the creation of derived documents such as conference proceedings. Delegates are offered a powerful, yet easy to use platform for registering for the event, maintaining their personal data, submitting abstracts and full papers, performing online payments and obtaining current information. Event organizers are provided with comprehensive tools for information acquisition, including freely configurable registration and abstract submission forms, auxiliary questionnaires and document upload functions. There is also functionality to send email messages to individuals, such as event triggered confirmation emails, as well as to groups of delegates (personalized bulk email). COMS runs on the computers of IT Services. It does therefore not need any IT infrastructure or IT support at the site of the conference organiser, other than the availability of an Internet connection and a current web browser.

Purpose:

The purpose of the conference management system is that the system can easily review the process. The main process in this document is the submission of paper by the candidate, reviewing process by the reviewer and sending of acknowledgement to the candidates whose paper is selected.

Scope:

The scope of this conference management is to select the best candidate from the list of candidates based on their performance in the process.

INTENDED AUDIENCE :

1.Candidate:

The candidate can login and submit the Paper to the reviewer. After getting Acknowledgement the candidate will Submit the revised and camera ready paper Then registration process Will be carried out.

2.Reviewer :

Reviewer will reviews the paper and Sending acknowledgement to the candidate.

3.Database :

Database is used to verify login and store. The details of selected candidates.

4.Software Requirement Specification :

This software specification documents full Set of features and function for conference management System.

DEFINITIONS , ACRONYMS , AND THE ABBREVIATIONS:

Conference Management System :

A software application designed to streamline the organization and management of conferences , including tasks such as registration , paper submission , scheduling and attendee management.

User:

An individual who interacts with the conference management system , including organizers , speakers, reviewers and attendees.

Reviewer:

A user assigned to evaluate and provide feedback on submitted papers , ensuring quality and relevance for conference acceptance .

HTTP:

HyperText Transfer Protocol.

CMS:

Conference Management System.

UX :

User Experience.

DBMS :

Database Management System.

ASSUMPTION AND DEPENDENCIES:

Assumptions:

The assumption is made that conference attendees will have access to stable internet connections , enabling seamless participation in virtual events.

Dependencies:

The successful implementation of the conference management system depends on the availability and compatibility of third-party video conferencing APIs for integrated virtual sessions.

INDENDED USE:

The user interface to make the process should be effective that is the system Will help the candidates to register easily. The system should be user friendly.

SYSTEM FEATURE AND REQUIREMENTS :

1.Functional Requirements:

Functional requirements are those that refer to the functionality of the system that is the services that are provided to the candidate who register for the conference.

2.External Interface Requirements :

The conference management system requires intuitive user interfaces accessible across devices and languages. Integration with secure payment gateways

is essential for processing registration fees securely. Additionally, a robust notification system must be in place to keep users informed of updates and deadlines via email or SMS.

3.System Features :

The conference management system should include a comprehensive set of features to facilitate efficient organization and execution of conferences. This encompasses functionalities such as attendee registration, paper submission, and peer review management to streamline the conference planning process. Additionally, the system should provide a user-friendly interface for browsing conference schedules, accessing session details, and viewing speaker profiles. Features like secure payment processing, notification management, and data analytics capabilities should also be incorporated to enhance user experience and ensure smooth operation throughout the conference lifecycle. Moreover, customizable branding options and support for multiple conference formats should be included to cater to the diverse needs of conference organizers.

NON-FUNCTIONAL REQUIREMENTS :

1.Performance Requirements :

The conference management system must exhibit robust performance to handle a large volume of concurrent users without significant slowdowns. Response times for critical functions like registration and submission should remain within acceptable limits even during peak usage periods. Additionally, the system should be scalable to accommodate growth in user base and data volume while maintaining optimal performance levels.

2.Safety Requirements :

The safety requirements for the conference management system are paramount to protect user data and ensure system integrity. This involves implementing robust encryption protocols to secure sensitive information during transmission and storage,

adhering to industry-standard security measures to prevent unauthorized access or data breaches, and conducting regular security audits and updates to address potential vulnerabilities and maintain system resilience against cyber threats.

3.Security Requirements :

The conference management system must enforce robust authentication and encryption protocols to safeguard user data and prevent unauthorized access. Additionally, adherence to industry-standard security practices and regular security audits are essential to mitigate potential vulnerabilities and ensure system resilience against cyber threats.

4.Software Quality Attributes :

The conference management system must prioritize several software quality attributes to ensure user satisfaction and system reliability. These include usability, with an intuitive interface and clear navigation paths, reliability through rigorous testing to minimize downtime and errors, and scalability to accommodate increasing user demands and data volumes while maintaining optimal performance levels. Additionally, the system should adhere to coding standards and best practices to facilitate maintenance and future enhancements.

5.Business Roles :

The conference management system caters to several key business roles essential for successful event organization. These roles include administrators responsible for system configuration, maintenance, and user management, organizers tasked with planning and executing conferences, speakers and authors who submit papers and presentations, and attendees who register for events and access conference materials. Each role requires specific functionalities within the system to fulfill their responsibilities effectively and contribute to the seamless operation of the conference management process.

.

RESULT :

Thus the Software Requirement Specification (SRS) document for Conference management system was implemented successfully.

EX NO :	IDENTIFY USE CASES AND DEVELOP THE USE CASE MODEL FOR CONFERENCE MANAGEMENT SYSTEM
DATE :	

AIM :

To identify the use cases and develop the use case model.

INTRODUCTION TO USE CASE DIAGRAM :

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It is represented using ellipse. Actor is any external entity that makes use of the system being modeled. It is represented using stick figure.

PURPOSE OF USE CASE DIAGRAM :

The purpose of a use case diagram is to illustrate the interactions between users and a system, capturing the system's functionalities from a user's perspective. It serves as a communication tool for stakeholders, aiding in requirements elicitation, system design, and testing. Use case diagrams help define system scope and assist in iterative development methodologies like Agile. Overall, they provide a clear, visual representation of system behavior and user interactions throughout the software development lifecycle.

COMPONENTS OF USE CASE DIAGRAM :

1.Actors :

Actors represent the roles played by external entities (such as users or other systems) interacting with the system. They are depicted as stick figures. In conference management system ,the actors include the candidate and the reviewer.

2.Use cases :

Use cases represent specific functionalities or actions that the system provides to its users. Each use case describes a set of interactions between the user and the system to achieve a specific goal or task . In conference management system ,examples of use cases include paper submission , review paper , send confirmation details , send revised paper and the registration.

3.Relationships :

Relationships illustrate the associations between actors and use cases. The primary relationships are "association" (connecting actors to use cases) and "include" or "extend" relationships (indicating the dependency of one use case on another).

BENEFITS OF USE CASE DIAGRAM :

1.Requirements Clarity:

Use case diagrams help in eliciting and documenting functional requirements, providing a clear understanding of the system's intended behavior from the user's perspective.

2.Communication Tool :

They serve as an effective communication tool between stakeholders, facilitating discussions about system functionality and requirements among developers, designers, clients, and users.

3.System Understanding:

Use case diagrams provide a visual representation of the system's behavior and interactions, aiding in understanding the overall system functionality and its relationship with users and external systems.

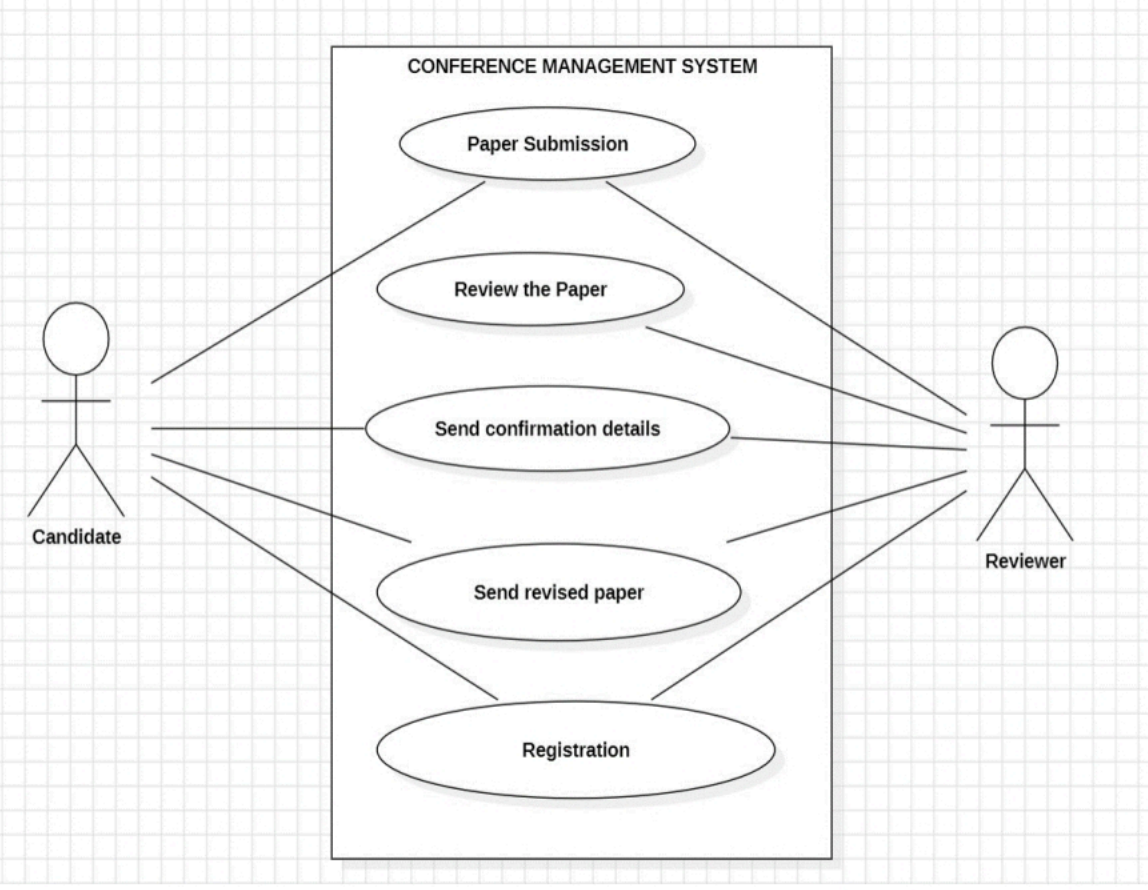
4.Scope Definition:

They assist in defining the scope of the system by identifying its boundaries and the interactions between the system and its users or external entities.

5.Design and Validation:

Use case diagrams aid in system design by providing insights into the major functionalities and interactions. They can also be used for validation purposes, ensuring that the system meets the specified requirements and user expectations.

USE CASE DIAGRAM FOR CONFERENCE MANAGEMENT SYSTEM :



Candidate:

Logins the conference system and submits the paper then do the registration process.

Reviewer:

Review the paper, select best candidate and send acknowledgement to them.

Paper Submission :

Candidate submits the paper.

Review the paper :

The paper is been reviewed by the reviewer and the paper is selected.

Paper Confirmation Details :

The reviewer can send the confirmation details to the candidate.

Revised and Camera Ready Paper :

After the paper is selected and the camera ready paper should be submitted to the reviewer by candidate.

Registration :

After submitting the revised paper the candidate wants to register.

RESULT :

Thus the use case diagram for conference management system was implemented and executed successfully .

EX NO :	IDENTIFY THE CONCEPTUAL CLASSES AND DEVELOP A DOMAIN MODEL AND DERIVE CLASS DIAGRAM FOR CONFERENCE MANAGEMENT SYSTEM
DATE :	

AIM :

To identify the conceptual classes and develop a domain model and derive class diagram for conference management system.

CONCEPTUAL CLASSES :

1.Conference:

This class represents the overarching entity of the conference itself. It includes attributes such as the conference title, date, location, theme, and organizers.

2.Attendee :

This class represents individuals who participate in the conference. It includes attributes such as attendee ID, name, contact information, affiliation, and registration status.

3.Speaker:

This class represents individuals who are scheduled to present at the conference. It includes attributes such as speaker ID, name, biography, presentation topics, and session details.

4.Session :

This class represents individual sessions within the conference program. It includes attributes such as session ID, title, description, time slot, location.

5.Paper :

This class represents academic papers or presentations submitted to the conference for review and potential inclusion in the program.

It includes attributes such as paper ID, title, abstract, authors, status, and review scores.

6.Reviewer :

This class represents individuals responsible for reviewing and evaluating submitted papers. It includes attributes such as reviewer ID, name, expertise, assigned papers, and review feedback.

7.Registration :

This class represents the registration details of attendees for the conference. It includes attributes such as registration ID, attendee information, registration type (e.g., early bird, standard), fees, and payment status.

8.Organizer:

This class represents individuals or organizations responsible for planning and managing the conference. It includes attributes such as organizer ID, name, contact information, and role.

DOMAIN MODEL FOR CONFERENCE MANAGEMENT SYSTEM :

candidateInfo :

- Attributes :
  - Cand_Id : Integer
  - Can_Name : String
  - Age : Integer
  - Qualification : String
  - Institution_Name : String
  - Gender : String
  - Phone_no : Integer
  - Email_id : String
  - Paper_Title : String

- Operations :
    - submitPaper()
    - sendCamReadyPaper()
    - register()

reviewer :

- Attributes :
    - Reviewer_id : Integer
    - Reviewer_name : String
    - Status : String
    - No_of_Paperrecd : Integer
- Operations :
    - reviewPaper()
    - sendConfirmMsg()

conferenceInfo :

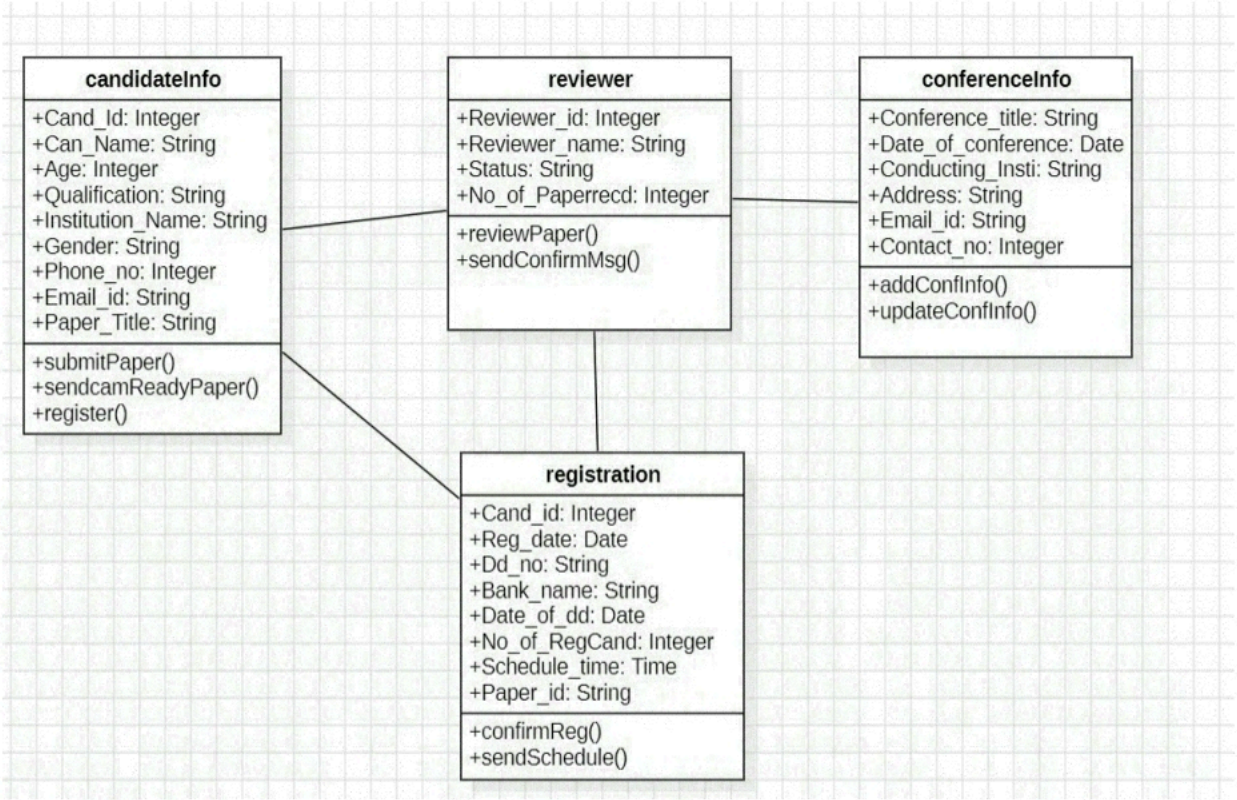
- Attributes :
    - Conference_title : String
    - Date_of_conference : Date
    - Conducting_Insti : String
    - Address : String
    - Email_id : String
    - Contact_no : Integer
- Operations :
    - addConfInfo()
    - updateConfInfo()

registration :

- Attributes :
  - Cand_id : Integer
  - Reg_date : Date
  - Dd_no : String
  - Bank_name : String
  - Date _of_dd : Date
  - No _of_RegCand : Integer
  - Schedule_time : Time
  - Paper_id : String
- Operations :
  - confirmReg()
  - sendSchedule()

This domain model captures the key entities and their attributes within the Conference Management System , such as Candidate Information , Reviewer , Conference Information and Registration. The operations associated with each entity represent the functionalities performed within the system including paper submitting , paper reviewing , sending confirmation details , sending camera ready paper , registration , adding and updating conference information , registration confirmation and sending the schedules.

CLASS DIAGRAM FOR CONFERENCE MANAGEMENT SYSTEM :



RESULT :

Thus the class diagram for Conference Management System was implemented and executed successfully.

EX NO :	USING THE IDENTIFIED SCENARIOS , FIND THE INTERACTION BETWEEN OBJECTS AND REPRESENT THEM USING UML SEQUENCE AND COLLABORATION DIAGRAMS
DATE :	

AIM :

To find the interaction between objects and represent them using UML Sequence and Collaboration diagram.

INTERACTION DIAGRAM :

Introduction :

- An interaction diagram is a type of diagram used in Unified Modeling Language (UML) to visualize the interactions between objects or components in a system.
- It helps in understanding how various objects collaborate to accomplish a specific task or scenario within the system.

TYPES OF INTERACTION DIAGRAM :

There are two main types of interaction diagrams: sequence diagrams and collaboration diagrams.

- **Sequence diagram:**

  A sequence diagram is a type of interaction diagram in the Unified Modeling Language (UML) that illustrates the dynamic behavior of a system by depicting the interactions between objects or components in a specific scenario or use case. It visually represents the sequence of messages exchanged between objects over time to achieve a particular functionality within the system.

- **Collaboration diagram :**

A collaboration diagram, also known as a communication diagram, is another type of interaction diagram in the Unified Modeling Language (UML). It depicts the interactions and relationships among objects or actors within a system to achieve a particular functionality or scenario.

Sequence diagram :

- A sequence diagram illustrates the interactions between objects in a sequential manner, showing the order in which messages are exchanged between objects over time.
- Objects are represented as vertical lifelines, and messages between objects are depicted as arrows.
- The sequence of messages and their timing helps in understanding the flow of control and the sequence of operations during the execution of a use case or scenario.
- Sequence diagrams are particularly useful for modeling the dynamic behavior of a system and for understanding the interactions between objects in different scenarios.

COMPONENTS OF A SEQUENCE DIAGRAM :

1.Lifelines:

Lifelines represent the participating objects or actors in the sequence diagram. Each lifeline is depicted as a vertical line, typically topped with the name of the object or actor it represents. Lifelines show the existence and lifetime of objects during the interaction.

2.Messages :

Messages are the means of communication between objects or actors in a sequence diagram. They represent the flow of control or data between lifelines. There are different types of messages:

- **Synchronous Messages:** Represented by solid arrows, synchronous messages indicate that the sender waits for a response from the receiver before proceeding.
- **Asynchronous Messages:** Represented by dashed arrows, asynchronous messages indicate that the sender does not wait for a response from the receiver and continues its execution immediately after sending the message.
- **Return Messages:** Represented by dashed arrows with a label indicating the return value, return messages depict the response sent by the receiver to the sender in response to a synchronous message.

3.Activation Boxes :

Activation boxes, also known as execution occurrences, represent the period during which an object is active or executing a particular operation. They are depicted as boxes placed on top of the lifeline, indicating the duration of time during which the object is processing a message.

4.Fragments :

Fragments represent conditional or iterative behavior within a sequence diagram. They include :

- **Alternative Fragments (alt):** Represents alternative paths of execution based on conditions. It is denoted by a rectangle with a guard condition, and different scenarios are shown within different partitions.
- **Option Fragment (opt):**Option fragments represent optional behaviors. They depict interactions that may or may not occur based on a condition. If t condition is true, the interactions within the option fragment are executed; otherwise, they are skipped.
- **Loop Fragment (loop):**Loop fragments represent iterative behaviors. They indicate that the interactions within the fragment will be repeated multiple times based on a condition. The loop fragment contains a guard condition that determines whether the loop should continue or terminate.

- **Parallel Fragment (par):** Parallel fragments represent parallel behaviors. They depict interactions that occur concurrently or in parallel. Objects within a parallel fragment can execute interactions independently of each other.
- **Strict Sequence Fragment (seq):** Strict sequence fragments represent strict sequential behaviors. They indicate that the interactions within the fragment must occur in a specific order without any deviations.

SEQUENCE DIAGRAM FOR CONFERENCE MANAGEMENT SYSTEM :

Candidate :

Logins the conference system and submits the paper then do the registration process.

Reviewer :

Review the paper, select best candidate and send acknowledgement to them.

Submit paper :

Candidate submits the paper.

Review paper:

The paper is been reviewed by the reviewer and the paper is selected.

Send confirmation details :

The reviewer can send the confirmation details to the candidate.

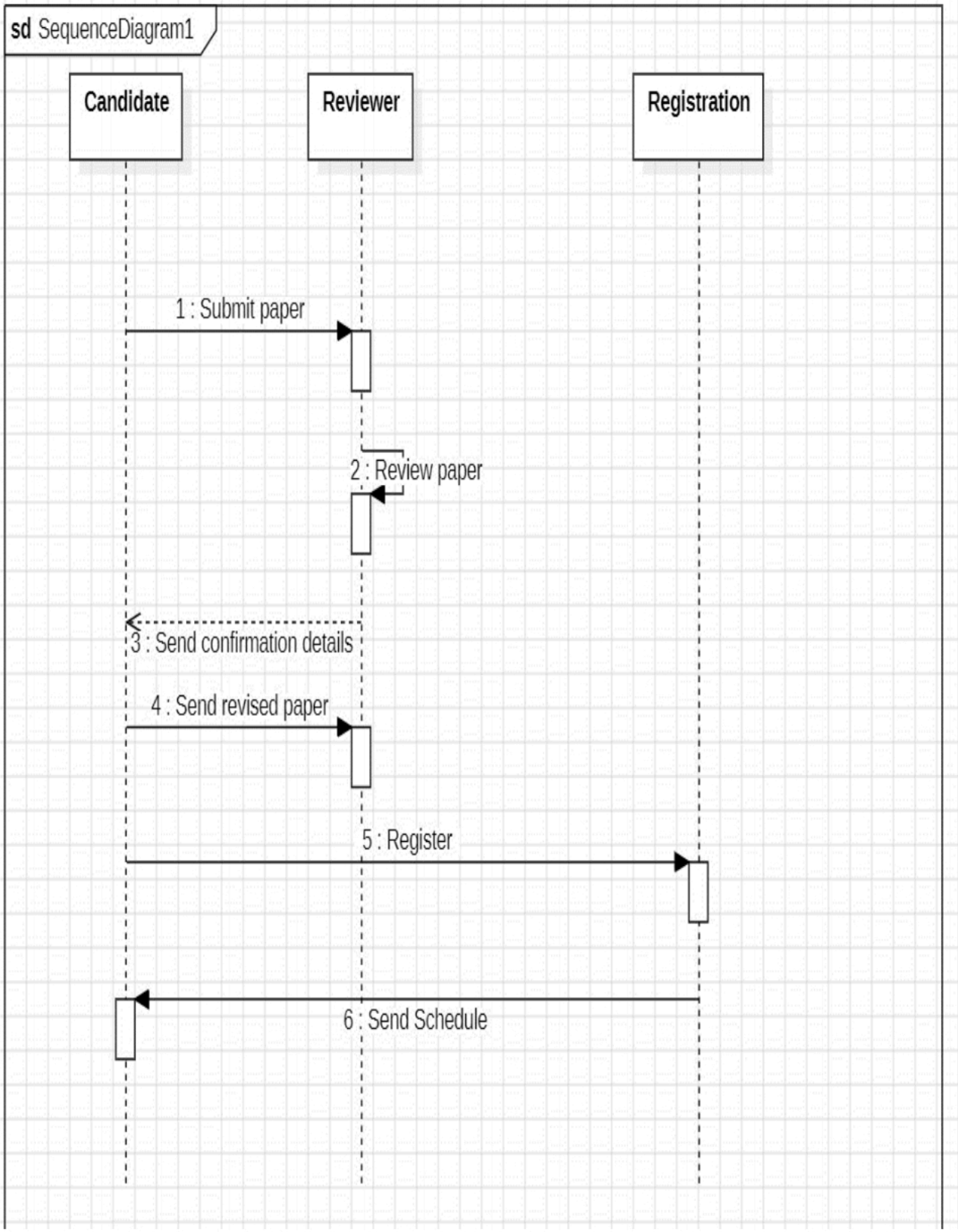
Revised and camera ready paper:

After the paper is selected and the camera ready paper should be submitted to the reviewer by candidate.

Registration:

After submitting the revised paper the candidate wants to register.

Sequence Diagram:



Collaboration Diagram:

- A collaboration diagram, also known as a communication diagram, is a type of interaction diagram in Unified Modeling Language (UML) that illustrates the interactions and relationships among objects or components within a system.
- Collaboration diagrams focus on the structural organization of objects and the messages exchanged between them to accomplish specific tasks.

COMPONENTS OF A COLLABORATION DIAGRAM :

1.Objects:

Objects represent the instances of classes or components within the system. Each object is depicted as a rectangle with its name or identifier written inside. Objects interact with each other by sending and receiving messages.

2.Links:

Links represent the connections or relationships between objects. They are depicted as lines connecting the objects involved in the interaction. Links can be directed or undirected, depending on the nature of the relationship between objects.

3.Roles:

Roles specify the responsibilities or behaviors associated with objects within the context of a collaboration. They define the expected actions or interactions that objects should perform to achieve specific objectives.

4.Notes and comments :

Notes and comments provide additional information or explanations about certain aspects of the collaboration diagram. They are typically added to clarify the roles of objects, the nature of relationships, or the purpose of specific interactions.

5.Optional components :

- **Constraints:** Represented using square brackets ([ ]), constraints specify conditions or constraints that must be satisfied during the interaction between objects**.**
- **Multiplicity:** Indicated using numerical values near the ends of links, multiplicity specifies the cardinality or number of instances participating in a particular relationship**.**

COLLABORATION DIAGRAM FOR CONFERENCE MANAGEMENT SYSTEM :

1.Submit paper :

Candidate submits the paper.

2.Review paper:

The paper is been reviewed by the reviewer and the paper is selected.

3.Send confirmation details :

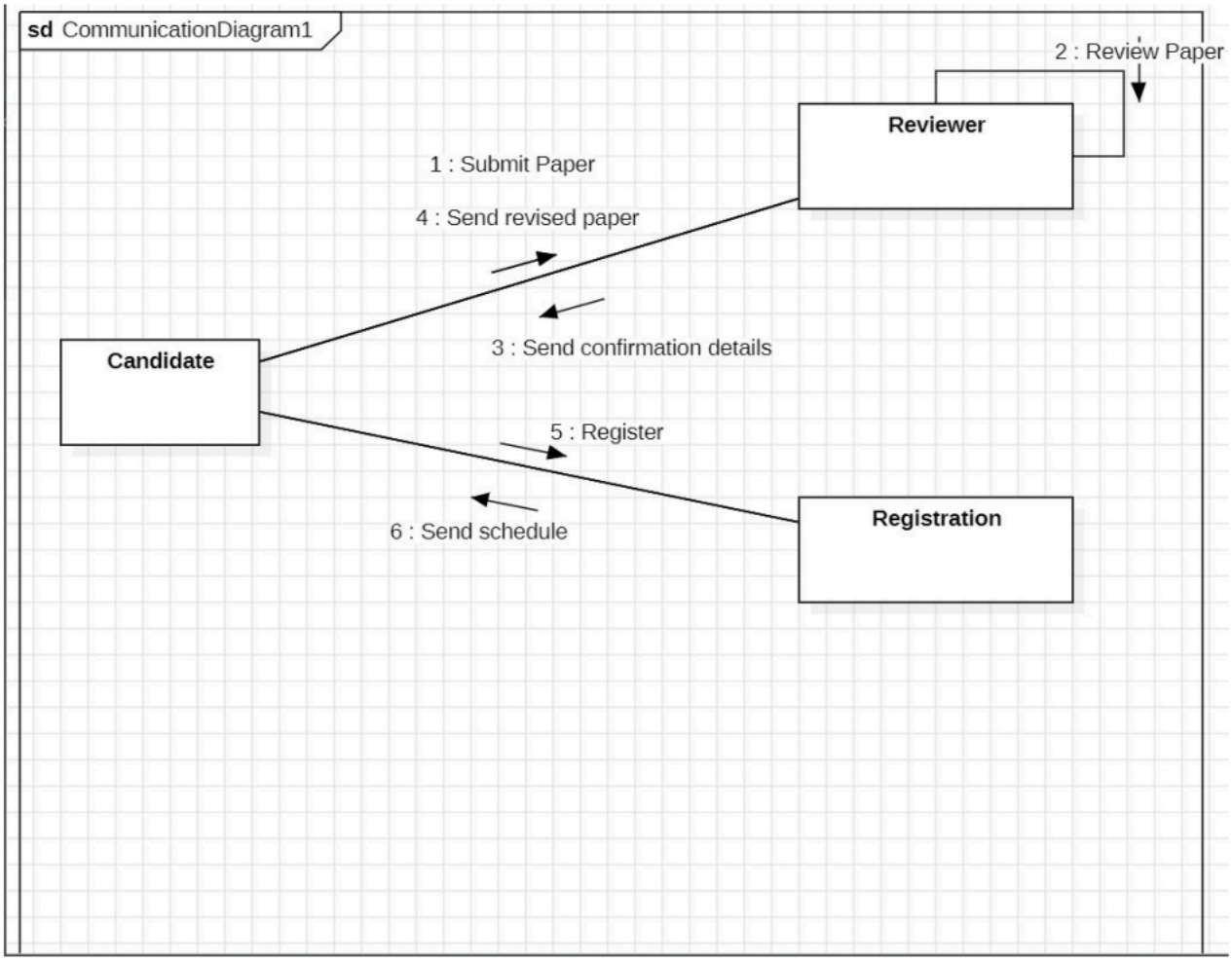
The reviewer can send the confirmation details to the candidate.

4.Send revised paper :

After the paper is selected and the camera ready paper should be submitted to the reviewer by candidate.

5.Registration:

After submitting the revised paper the candidate wants to register



RESULT :

Thus to find the interaction between objects and represent them using UML sequence and collaboration diagram in Conference Management System was implemented successfully.

EX NO :	DRAW RELEVANT STATE CHART AND ACTIVITY DIAGRAM FOR THE SAME SYSTEM
DATE :	

AIM :

To draw Activity diagram for Conference Management System.

ACTIVITY DIAGRAM :

The activity diagram for a conference management system is a visual representation illustrating the sequential flow of activities and interactions within the system. At its inception, the diagram initiates with the login/authentication step, ensuring secure access for users.

Following successful authentication, users can view the conference schedule, providing them with essential information about sessions, speakers, and timings. Subsequently, attendees have the option to register for the conference, providing necessary details and payment if required.

Authors may submit papers or proposals for consideration, which undergo a rigorous review process involving assignment to reviewers and subsequent decision-making by conference organizers.

Accepted submissions are acknowledged, while rejected ones are communicated back to the authors. Concurrently, organizers manage various aspects such as scheduling sessions, handling attendee registrations, and coordinating resources essential for the conference, including venue arrangements, equipment, and catering services.

As the conference unfolds, organizers oversee the seamless execution of events, sessions, and presentations. Post-conference activities involve gathering attendee feedback for evaluation and future planning.

The activity diagram encapsulates the entire conference management lifecycle, from initiation to conclusion, within a structured visual framework, aiding in

understanding the flow of activities and interactions within the system. Each node and connection represents a specific activity or decision point, facilitating clarity in system understanding and process optimization.

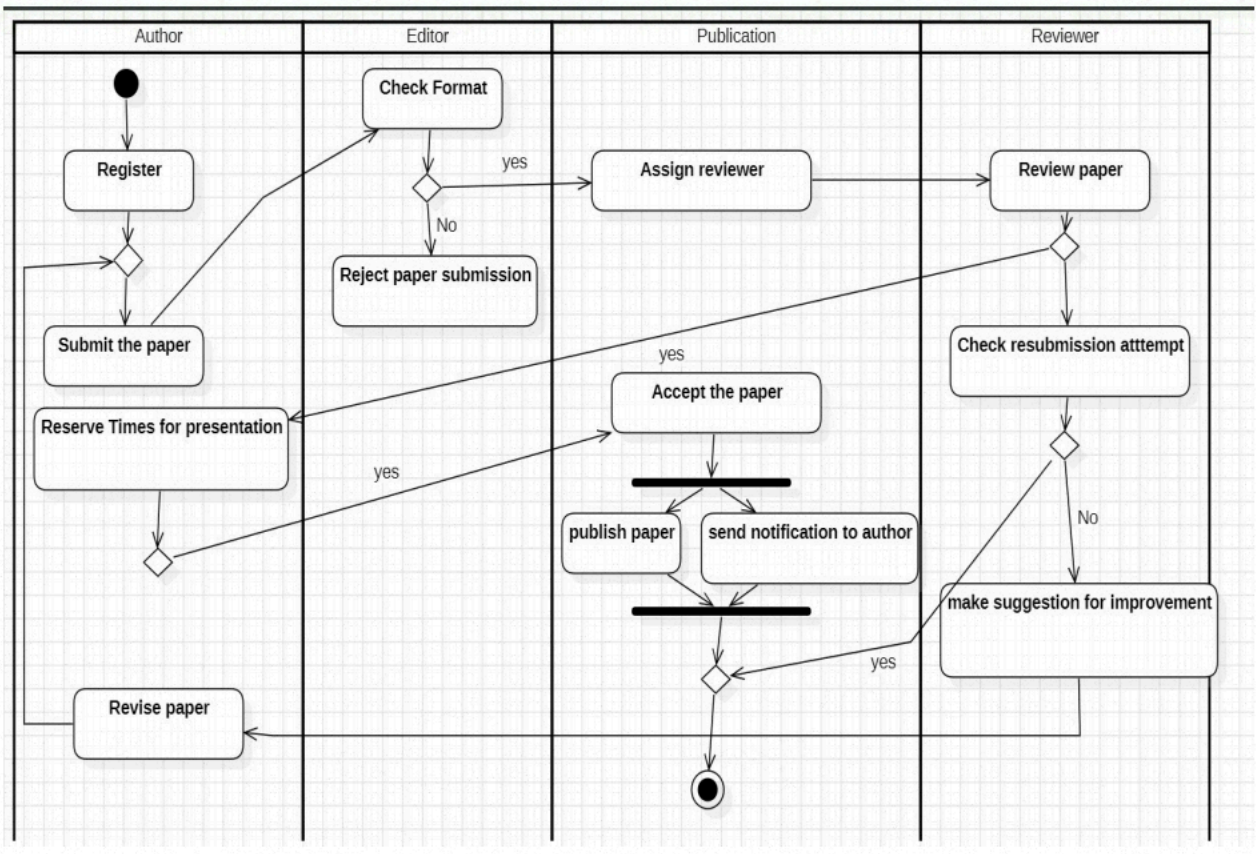
Ultimately, the activity diagram serves as a blueprint for designing, implementing, and refining the conference management system, ensuring efficiency and effectiveness in its operation.

Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

The purposes can be described as:

- ♣ Draw the activity flow of a system.
- ♣ Describe the sequence from one activity to another.
- ♣ Describe the parallel, branched and concurrent flow of the system.

ACTIVITY DIAGRAM FOR CONFERENCE MANAGEMENT SYSTEM :



RESULT :

Thus the activity diagram for Conference Management System was developed successfully.

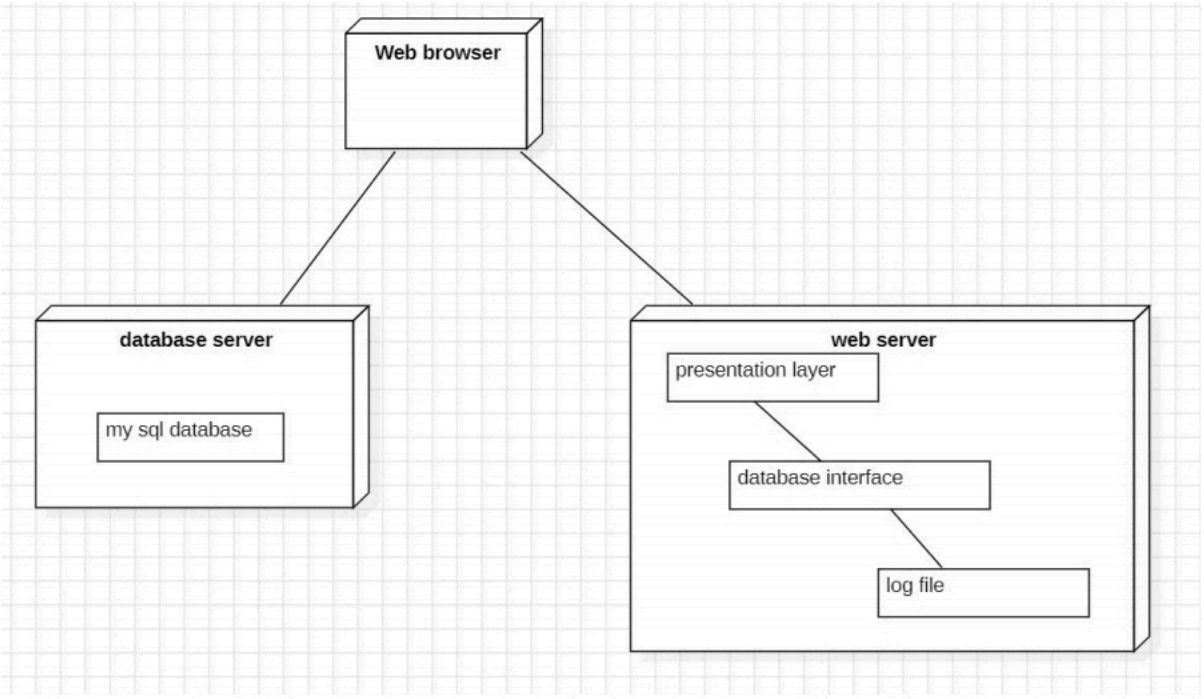
EX NO :	IMPLEMENT THE SYSTEM AS PER THE DETAILED DESIGN
DATE :	

AIM :

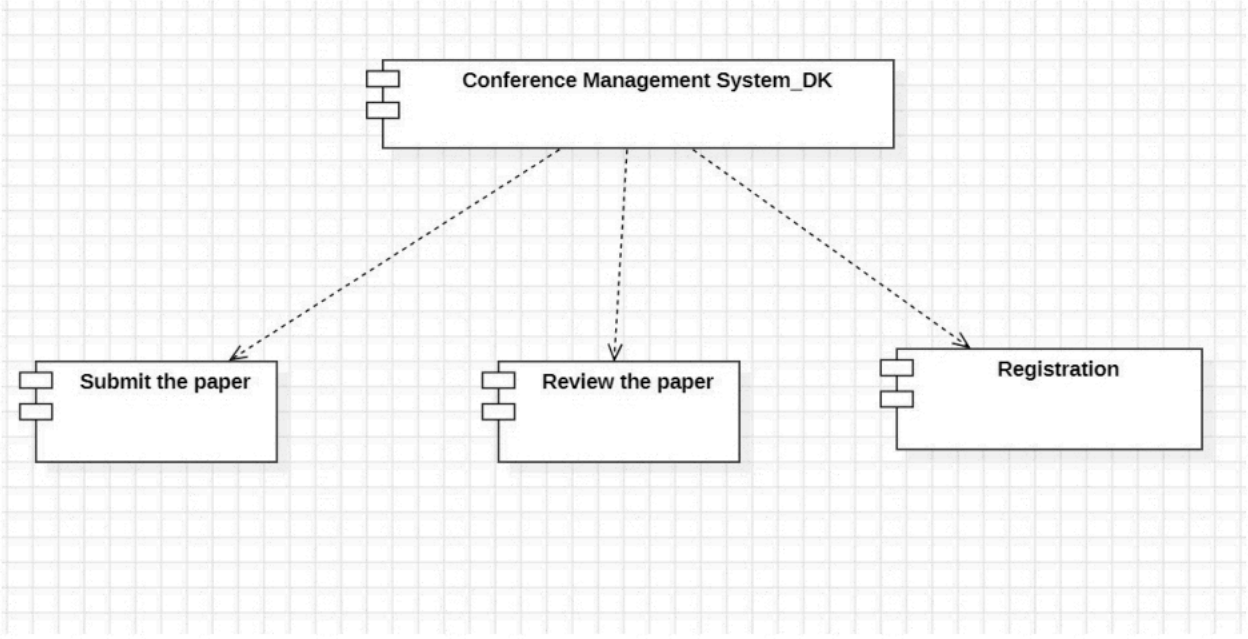
To implement the Conference Management System as per the detailed design.

IMPLEMENTATION DIAGRAM:

Deployment diagram :



Component diagram :



IMPLEMENTATION CODE:

candidateInfo.java

```
import java.util.*;
public class candidateInfo {
    public candidateInfo() {
    }
    public Integer Cand_Id;
    public String Can_Name;
    public Integer Age;
    public String Qualification;
    public String Institution_Name;
    public String Gender;
   public Integer Phone_no;
   public String Email_id;
   public String Paper_Title;
   public void submitPaper() {

    }
   public void sendcamReadyPaper() {

    }
```

```
public void register() {
    }
}
```

conferenceInfo.java

```
import java.util.*;
public class conferenceInfo {
    public conferenceInfo() {
    }
    public String Conference_title;
    public Date Date_of_conference;
    public String Conducting_Insti;
    public String Address;
    public String Email_id;
   public Integer Contact_no;
   public void addConfInfo() {

    }
  public void updateConfInfo() {

    }
}
```

registration.java

```
import java.util.*;

public class registration {
    public registration() {
    }
```

```
    public Integer Cand_id;
    public Date Reg_date;
    public String Dd_no;
   public String Bank_name;
   public Date Date_of_dd;
   public Integer No_of_RegCand;
   public Time Schedule_time;
   public String Paper_id;
   public void confirmReg() {

    }
  public void sendSchedule() {

    }
}
```

reviewer.java

```
import java.util.*;
public class reviewer {
   public reviewer() {
    }
    public Integer Reviewer_id;
    public String Reviewer_name;
    public String Status;
    public Integer No_of_Paperrecd;
    public void reviewPaper() {
    }


}
```

```
public void sendConfirmMsg() {


    }
}
```

RESULT :

Thus the implementation of Conference Management System was executed and the above codes were generated successfully.

Ex.No:08	Test the software system for all scenarios identified as per the use case diagram
Date:	

Aim:

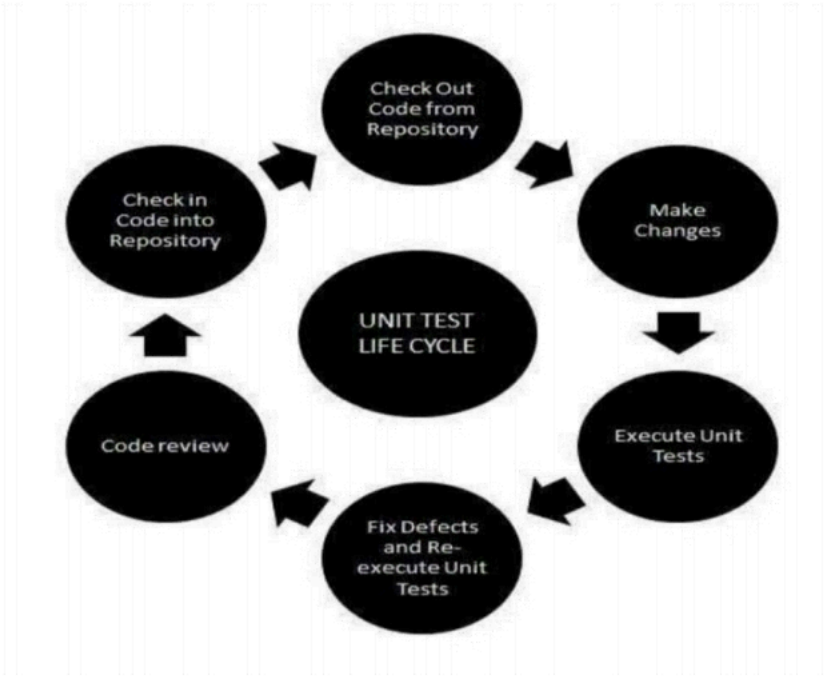
To test the software system for all scenarios identified in the use case diagram.

Introduction:

In the context of higher education institutions, an efficient Conference System is crucial for handling the process of inviting speakers, managing attendee registrations, scheduling presentations, and selecting participants. This system fosters collaborations between conference organizers and speakers, ensuring a smooth and successful event.

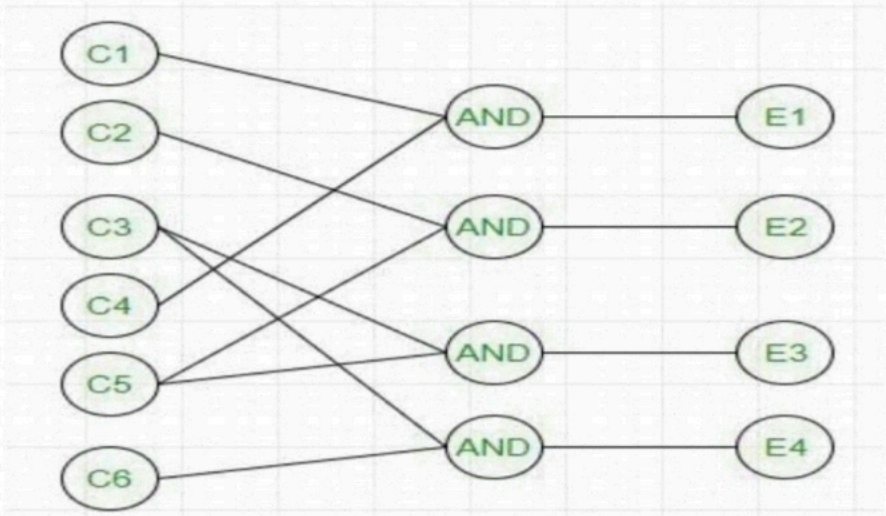
i) Unit Testing:

- Unit testing involves testing individual components or modules of the software to ensure they function correctly in isolation.
- Each module, such as client management, project management, employee management, etc., would undergo unit testing.
- Example: Testing the "register()" function to ensure that a new client is successfully added to the system



ii) Black Box Testing:

- Black box testing is a technique where the internal workings of the system are not known to the tester. The tester only tests the system's functionality based on its specification.
- Testers would input various sets of data into the system and verify that the expected output is produced.
- Example: Testing the "submitPaper()" functionality to ensure that it returns the expected results based on different search criteria.

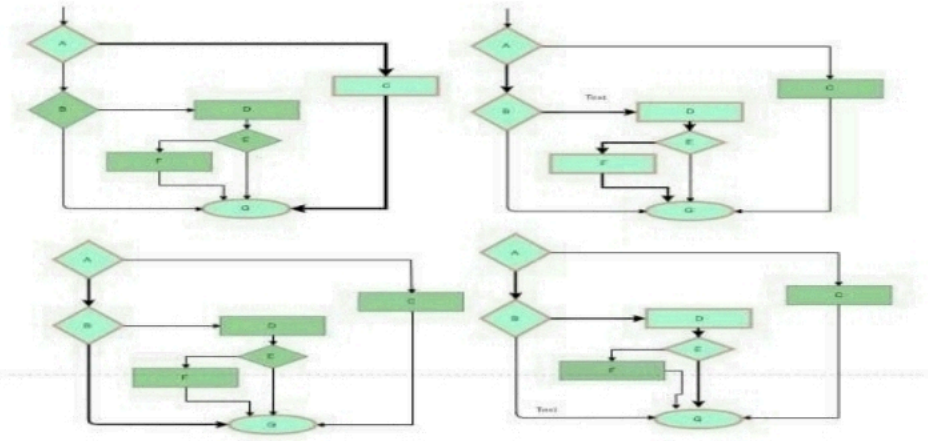


		1	2	3	4
CAUSES	C1	1	0	0	0
	C2	0	1	0	0
	C3	0	0	1	1
	C4	1	0	0	0
	C5	0	1	1	0
	C6	0	0	0	1
EFFECTS	E1	X	-	-	-
	E2	-	X	-	-
	E3	-	-	X	-
	E4	-	-	-	X

iii) White Box Testing:

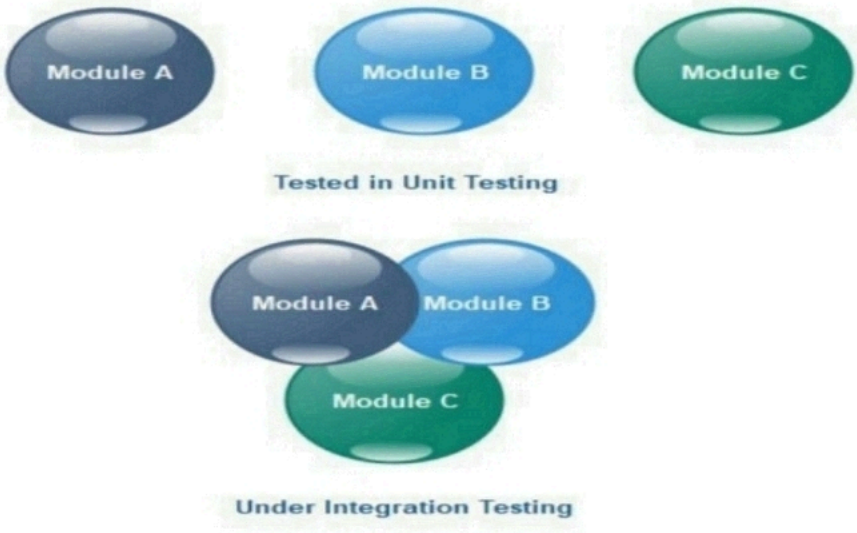
- White box testing involves testing the internal logic and structure of the software code.

- Testers would examine the code of individual modules to ensure that all code paths are tested.
- Example: Testing the "reviewPaper()" function to ensure that it adequately checks the quality of the processed data.



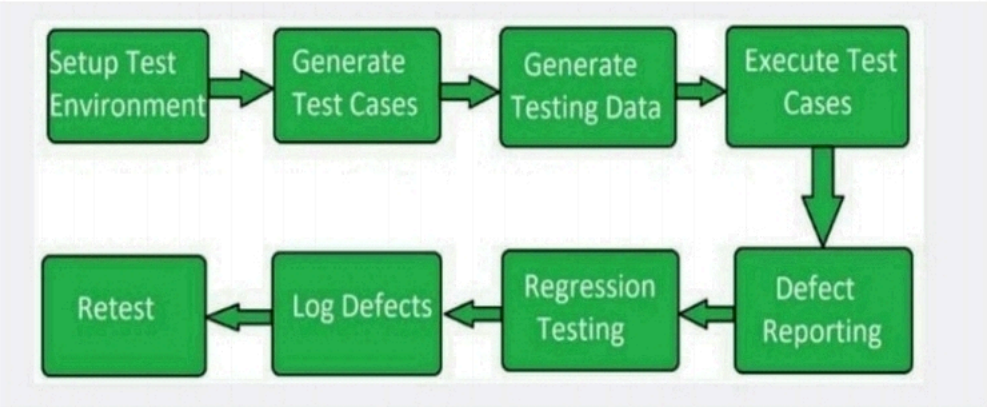
iv) Integration Testing:

- Integration testing verifies that different modules of the software work together as expected.
- Testers would test the interaction between different modules, such as client management, project management, and employee management.
- Example: Testing the interaction between the "register()" and "submitPaper()" functions to ensure that a project can be associated with a client successfully.



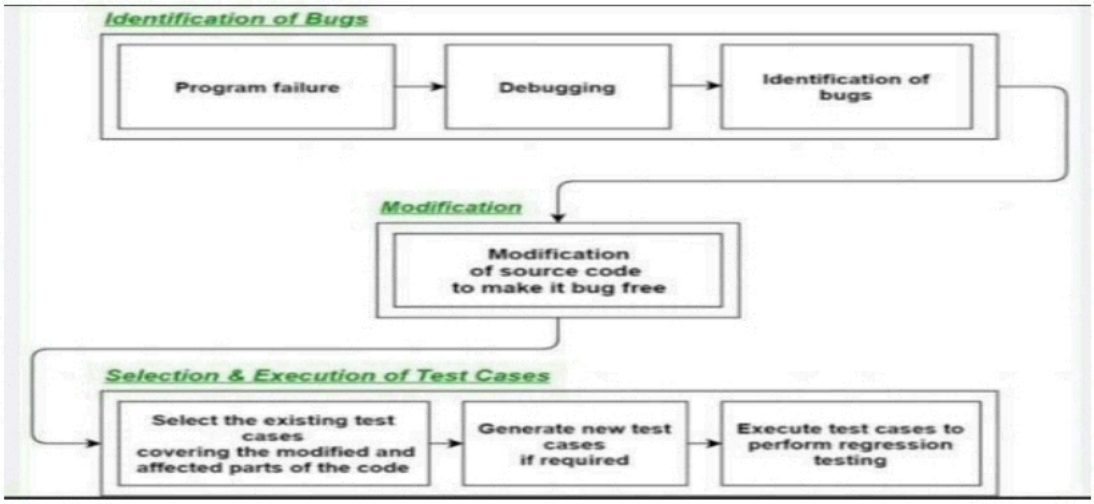
v) System Testing:

- System testing involves testing the entire system as a whole to verify that it meets the specified requirements.
- Testers would perform end-to-end testing of the entire system, including all modules and their interactions.
- Example: Testing the entire process from client negotiation to project delivery and payment to ensure that it functions as expected.



vi) Regression Testing:

- Regression testing ensures that new changes or additions to the system do not adversely affect existing functionalities.
- Testers would re-run previously conducted tests after new changes or additions are made to the system to ensure that existing functionalities are not affected.
- Example: After adding a new feature to the system, testers would re-run all existing tests to ensure that the new feature did not introduce any bugs or errors.



Result:

Thus the software system for all scenarios identified in the use case diagram was tested successfully.

Ex. No:09	Improve the Reusability and Maintainability of the Software System by Applying Appropriate Design Patterns
DATE:	

Aim:

To improve the reusability and maintainability of the software by applying appropriate design patterns.

Conference Management System:

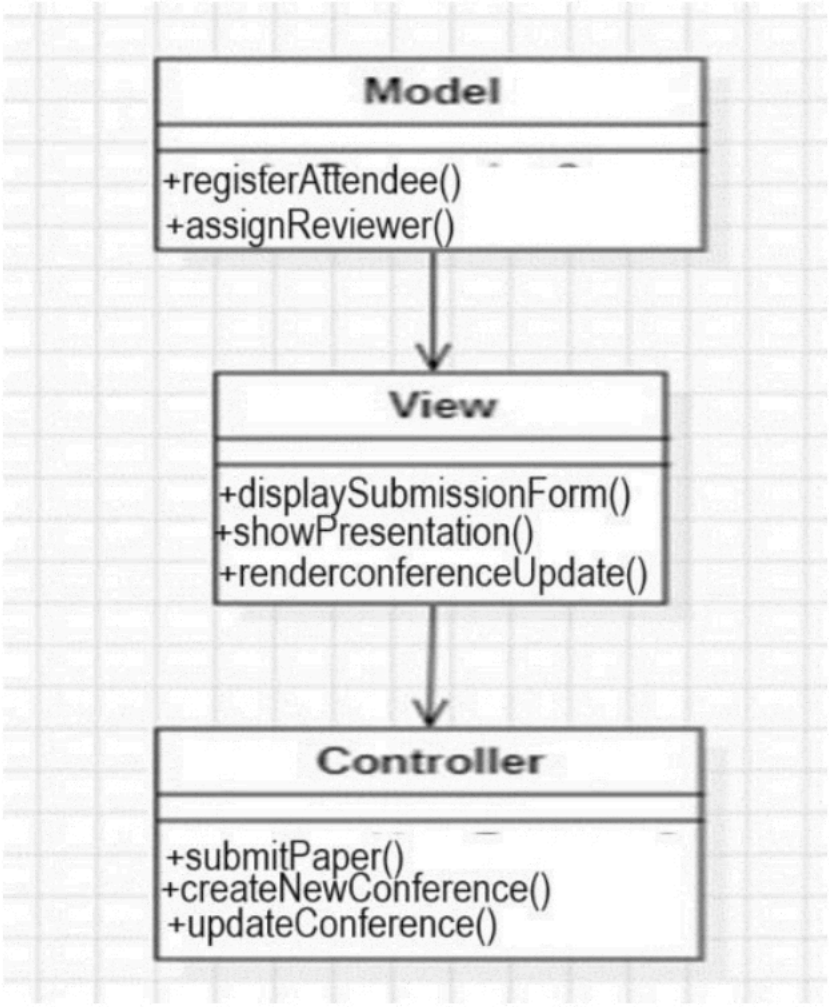
A Conference Management System is a software platform created to simplify and oversee conference organization and participation.It facilitates communication among attendees, organizers, reviewers, and administrators, making tasks like conference registration, paper submission, session scheduling, and system maintenance more efficient.

To enhance the reusability and maintainability of the Management System, several design patterns can be incorporated:

1.Model-View-Controller Pattern:

- The MVC pattern separates the representation of information from the user's interaction with it.
- In the Conference Management system,it divides the system into three interconnected components: Model (data and logic), View (interface), and Controller (manages input and updates).

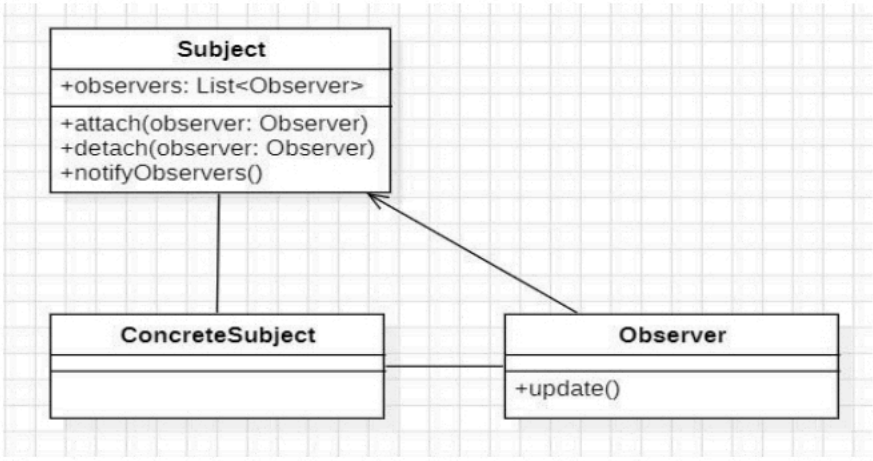
- Context: MVC separates trade data management, UI presentation, and user interaction.
- Problem: Without MVC, code complexity increases, making maintenance difficult.
- Solution: MVC divides the system into Model, View, and Controller, simplifying development and improving code organization



2. Observer Pattern:

- In the conference management system, we can apply the Observer pattern to notify attendees of new sessions or updates to their submissions.
- Attendees can subscribe to session categories or specific speakers to receive relevant notifications.

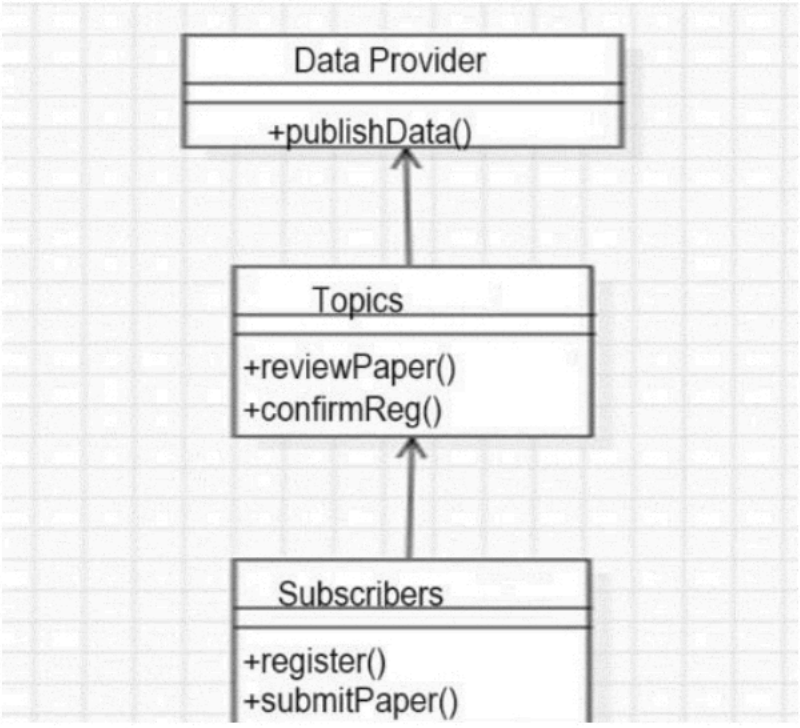
- Context: Used when objects need to be notified of changes in another object's state.
- Problem: Establishes a one-to-many relationship without tightly coupling objects, ensuring automatic updates.
- Solution: Define a subject interface with methods for attaching, detaching, and notifying observers. Observers register with subjects to receive updates on state changes



3. Publish-Subscribe Pattern:

- The Publish-Subscribe Pattern facilitates instantaneous updates and notifications among numerous components within conference management systems.
- Its implementation bolsters scalability, flexibility, and responsiveness within the architecture, optimizing the management of conferences and events**.**

- Context: Real-time updates are crucial in trading operations.
- Problem: Distributing real-time market data without the Publish-Subscribe Pattern leads to data latency and inefficiencies
- Solution: The pattern enables efficient data distribution, letting providers publish to specific topics, ensuring relevant updates reach subscribing modules, enhancing system scalability and responsiveness.



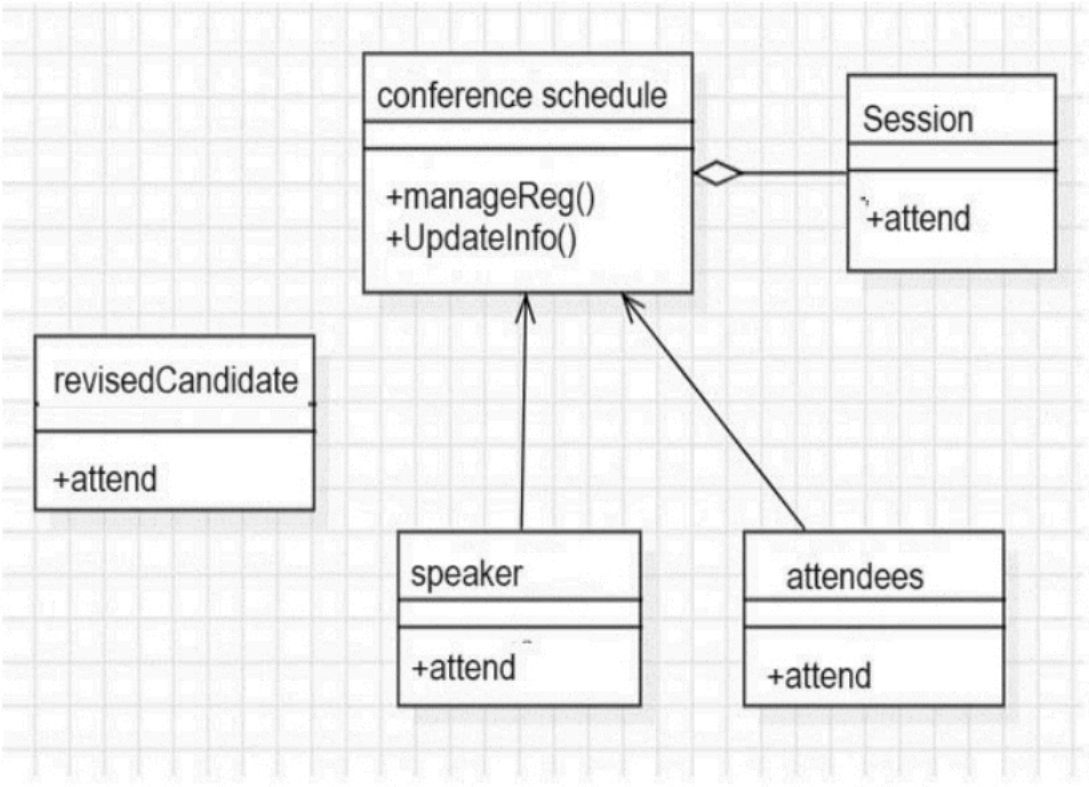
4. Strategy Pattern:

- In the conference management system, we can implement the Strategy pattern to define different session scheduling algorithms or paper evaluation strategies.
- These strategies can be interchangeable, enabling organizers to customize the conference management process according to their specific preferences or requirements.

➢ Context: Used to manage a family of algorithms interchangeably.

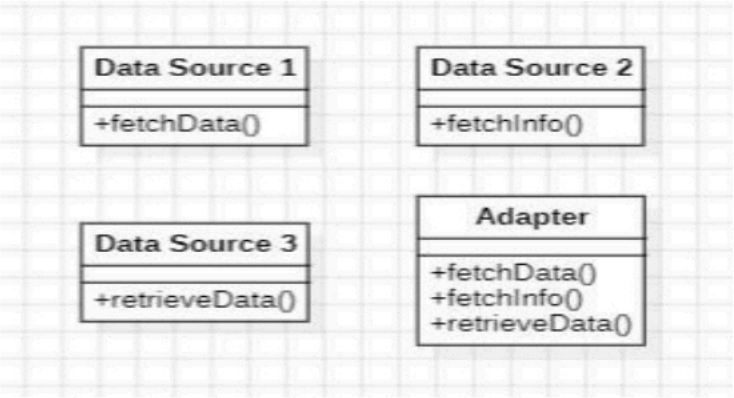
➢ Problem: Facilitates dynamic algorithm selection and switching, promoting code flexibility and reusability.

➢ Solution: Define a strategy interface with algorithm methods. Implement concrete strategy classes for each algorithm and allow runtime switching between them.



5.Adapter Pattern:

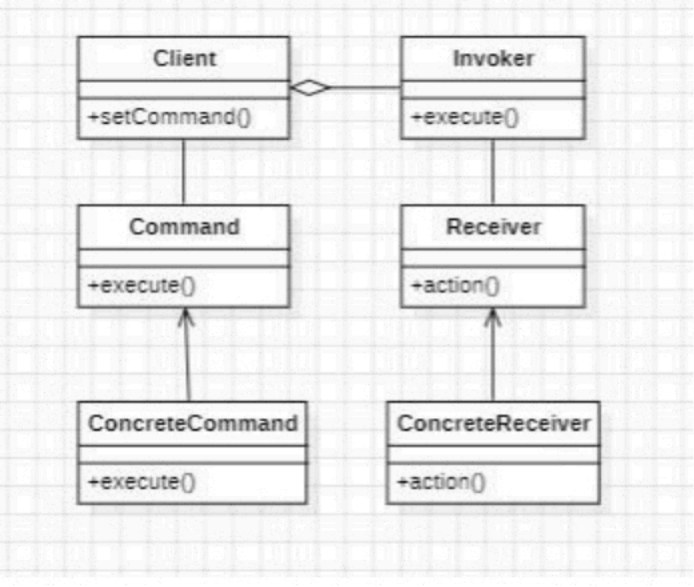
- The Adapter Pattern is utilized in conference management systems to seamlessly integrate various data sources. It addresses the challenge of incompatible interfaces between these sources and the management system by offering a wrapper that translates different data formats and protocols.
- This approach guarantees smooth communication and data exchange, thereby enhancing interoperability and flexibility within the conference management ecosystem



6. Command Pattern:

- In the conference management system, we can implement the Command pattern to encapsulate actions performed by organizers, such as scheduling sessions, updating session details, or managing attendee registrations.
- Each command represents a specific action, like adding a new session or modifying session information

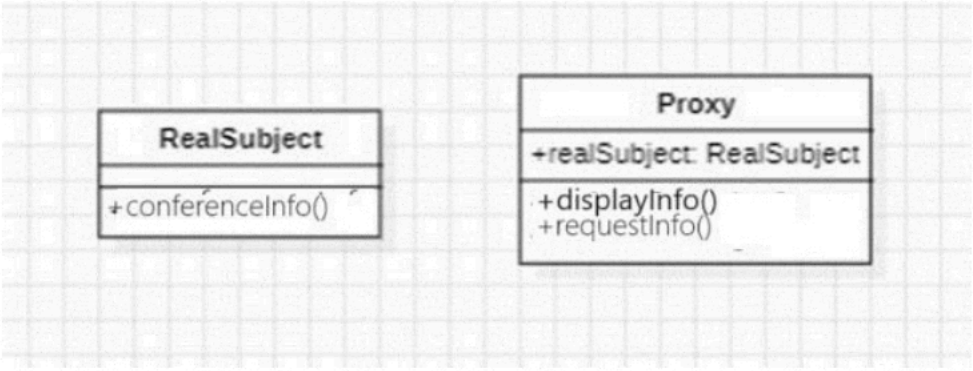
- Context: Used to encapsulate requests as objects, enabling parameterized and queued requests.
- Problem: Decouples sender and receiver, supports undoable operations, and provides structured request handling.
- Solution: Define command objects that encapsulate requests and parameters. Clients create and queue commands, and invokers execute them when needed.



7. Proxy pattern:

- The Proxy Pattern is employed in conference management systems to manage access to sensitive data or resource-intensive operations.
- It addresses the challenge of direct access to such resources by introducing a surrogate object that serves as a stand-in, overseeing access and incorporating additional functionalities such as caching or logging.

- ➢ Context: Need to control access to sensitive trading data or manage expensive operations.
- ➢ Problem: Direct access to such resources can lead to security risks or performance issues.
- ➢ Solution: Proxy Pattern provides a surrogate object managing access, adding functionalities like caching or logging, enhancing security, performance, and resource management in the trading system

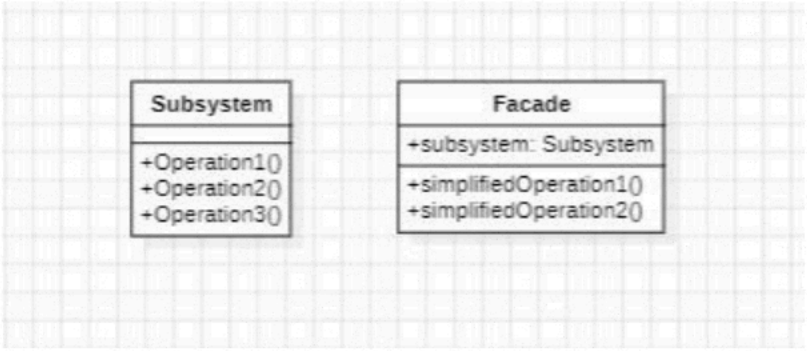


8. Facade pattern:

- In the conference management system, we can create a façade to streamline interactions among attendees, organizers, reviewers, and administrators.
- This façade would simplify processes like event registration, paper submission, session scheduling, and system administration by abstracting away complexities such as authentication and submission procedures.

- ➢ Context: Direct interactions with subsystems cause code complexity and increased dependencies.
- ➢ Problem: Managing multiple subsystem interactions leads to complex code and dependencies.

- Solution: Facade Pattern simplifies interactions, hiding subsystem complexities for easier system management and maintenance.



By incorporating these design patterns into the Foreign Trading System, we can improve the reusability and maintainability by promoting code reuse, encapsulating complexity, and decoupling components. This leads to a more modular, flexible, and maintainable architecture, making it easier to extend, modify, and maintain the system over time.

Result:

Thus, the reusability and maintainability of the software system by applying appropriate design pattern was improved successfully.