AIM:

To implement a software for Conference management system

(I) PROBLEM STATEMENT:

The Conference Management System is an online website in which candidate can submit the paper and register themselves and then attend the conference. The paper will be reviewed. The details of the conference, date and time will be made available to them through the website. After getting the confirmation details the candidate should submit the revised and camera ready paper. Then the registration process will be done.

(II) SOFTWARE REQUIREMENT SPECIFICATION:

1.0 INTRODUCTION

This software specification document consist full set of features and function for online conference management system. In this we give specification about the system requirements that are apart from the functionality of the system to perform the candidate paper valuation. It tells the usability, reliability defined in use case specification.

1.1 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.

1.2 SCOPE

The scope of this conference management process is to select the best candidate from the list of candidates based on their performance in the process.
1.3 DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS

- **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.

- **REVIEWER** - Reviewer will reviews the paper and sending acknowledgement to the candidate

- **DATABASE** - Database is used to verify login and store the details of selected candidates.

- **HTML** - Markup Language used for creating web pages.

- **J2EE** – Java 2 Enterprise Edition is a programming platform java platform for developing and running distributed java applications.

- **HTTP** - Hyper Text Transfer Protocol.

- **TCP/IP** – Transmission Control Protocol/Internet Protocol is the communication protocol used to connect hosts on the Internet.

1.4 REFERENCES

IEEE Software Requirement Specification format.

1.5 TECHNOLOGIES TO BE USED

- HTML
- JSP
- Java

1.6 TOOLS TO BE USED

- Eclipse IDE (Integrated Development Environment)
- Rational Rose tool (for developing UML Patterns)

1.7 OVERVIEW

SRS includes two sections overall description and specific requirements –
Overall Description will describe major role of the system components and inter-connections.

Specific Requirements will describe roles & functions of the actors.

2.0 OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The process of the candidates is to login the conference system and submit the paper through online. Then the reviewer reviews the paper and sends the acknowledgement to the candidate either paper selected or rejected.

2.2 SOFTWARE INTERFACE

- Front End Client - The exporter online interface is built using JSP and HTML.
- Web Server – Apache Tomcat Server (Oracle Corporation)
- Back End - Oracle 11g database

2.3 HARDWARE INTERFACE

The BPO system’s server is directly connected to the client systems via ftp. The client systems have access to the database in the server.

2.4 SYSTEM FUNCTIONS

This process of conference management system are described sequentially through following steps,

• The candidate login to the conference management system.
• The paper title is submitted.
• The paper is been reviewed by the reviewer.
• The reviewer sends acknowledgement to the candidate.
• Based on the selection, the best candidate is selected.
• Finally the candidate registers all details.
2.5 USER CHARACTERISTICS

- **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.

2.6 CONSTRAINTS

- Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.

- The user has to be careful while submitting the information. Much care is required.

2.7 ASSUMPTIONS AND DEPENDENCIES

- The candidate and reviewer must have basic knowledge of computers and English Language.

- Provide privacy and security for the documents and candidate information

(III) USECASE 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.

The conference management system use cases are:

1. Paper submission
2. Review the paper
3. Send confirmation details
4. Send revised paper
5. Registration
ACTORS:

Actors are as follows:

1. Candidate
2. Reviewer

ACTORS DOCUMENTATION:

- **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.

![UML USE CASE DIAGRAM](image)

**Fig.3.1 UML USE CASE DIAGRAM**
(IV) ACTIVITY DIAGRAM

Fig. 4. UML ACTIVITY DIAGRAM FOR CONFERENCE MANAGEMENT SYSTEM
An activity diagram is a variation or special case of a state machine in which the states or activity representing the performance of operation and transitions are triggered by the completion of operation.

The purpose is to provide view of close and what is going on inside a use case or among several classes. An activity is shown as rounded box containing the name of operation.

(V)UML CLASS DIAGRAM:

A class diagram in the unified modeling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. Top compartment have the class name, middle compartment the attributes and the bottom compartment with operations.

This class diagram has three classes candidate, reviewer and database.

- **Candidate** – Its attributes are name, collegename, department, paper title. The operations performed in the candidate class are login, submit the paper, submit revised and camera ready paper and registration.

- **Reviewer** – Its attributes are name, department, reviewer ID The operations performed are review the paper and send the paper confirmation details.

- **Database** – The operations performed are storing candidate details and verifying login.
Fig. 5.1 UML CLASS DIAGRAM
(VI) UML SEQUENCE DIAGRAM:

A sequence diagram illustrates a kind of format in which each object interacts via message. It is generalize between two or more specialized diagram.

Fig. 6.1 SEQUENCE DIAGRAM
Communication diagram illustrate that object interact on a graph or network format in which object can be placed where on the diagram. In collaboration diagram the object can be placed in anywhere on the diagram. The collaboration comes from sequence diagram.

Fig. 6.2 COLLABRATION DIAGRAM
(VII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

![Deployment Diagram]

(IX) IMPLEMENTATION OF DOMAIN OBJECTS LAYER AND TECHNICAL SERVICE LAYER

//Source file: F:\vaish\Candidateinfo.java

public class Candidateinfo
{


public integer candid;
public string candname;
public integer age;
public string qualification;
public string institutionname;
public string gender;
public integer phoneno;
public string emailid;
public string paperTitle;
public reviewer theReviewer;
public registrations theRegistrations;
/**
* @roseuid 5142F919001F
*/

public Candidateinfo()
{
}
/**
* @roseuid 5142F3D10119
*/

public void submitpaper()
{
}
/**
* @roseuid 5142F3D202CE
*/
public void sendcamreadypaper()
{
}
/**
 * @roseuid 5142F3D4031C
 */
public void register()
{
}
}
/**
void CandidateInfo.register(){
}
CandidateInfo.CandidateInfo()
void CandidateInfo.submitPaper(){
}
void CandidateInfo.sendCamReadyPaper(){
}
*/

//Source file: D:\uma\conferenceInfo.java
public class conferenceInfo
{
    private string conferenceTitle;
private date dateOfConference;
private string conductingInsti;
private string address;
private string emailid;
private integer contactno;
public review theReviewer;
/**
@roseuid 513441B80222
*/
public conferenceInfo()
{
}
/**
@roseuid 513440F60167
*/
public void addConfInfo()
{
}
/**
@roseuid 5134410002BF
*/
public void updateConf()
{
}
public class registration {

    private integer candId;
    private date regDate;
    private string DDNo;
    private string bankname;
    private date dateofDD;
    private integer noofregCan;
    private time scheduleTime;
    private string paperId;
    public candidateInfo theCandidateInfo;
    public review theReviewer;

    /**
     * @roseuid 513441B80177
     */
    public registration()
    {
    }

    /**
     * @roseuid 5134416402FD
     */
    public void confirmreg()
    {
    }
public void sendSchedule()
{
}

//Source file: F:\vaish\reviewer.java
public class reviewer
{
   public integer reviewerid;
   public string reviewername;
   public string status;
   public integer noofpaperrecd;
   public conferenceinfo theConferenceinfo;
   public registrations theRegistrations;
   /**
   @roseuid 5142F89301F4
   */
   public reviewer()
   {
   }
   /**
   @roseuid 5142F3E300AB
   */
public void reiviewerpaper()
{
}
/**
 * @roseuid 5142F3E6008C
 */
public void sendconfirmmsg()
{
}

/**
 * String reviewer.reviewPaper()
 * { return null; }
 *
 * String reviewer.sendConfirmMsg()
 * }
 */

CONCLUSION:

Thus the mini project for conference management system has been successfully executed and codes are generated.