SOFTWARE REQUIREMENTS SPECIFICATION
VERSION 1.0

Date : 6th February, 2014

TATHVA PAPER SUBMISSION PORTAL

Submitted by:

JAY SHANKAR YADAV (M120387CA) (Team Leader)

ABHINABA AUDHYA (M120360CA)

HARIPRIYA V (M120440CA)

VIKASH KUMAR (M120354CA)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF
CS3004 SOFTWARE ENGINEERING

<<Any comments inside double brackets such as these are *not* part of this SRS but are comments upon this SRS example to help the reader understand the point being made. >>

Table of Contents

List of Figures

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Tathva Paper Submission Portal. It will explain the purposes and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.2. Scope Of The Project

This software system will be the submission portal for papers to submitted by various users (competitors) during Tathva. This system will be designed to automate most of the steps starting from the paper submission to the end result publication which will come in handy. It will meet the various needs of the competition while remaining inherently easy to understand and use.

More specifically, this system is designed to help the tathva committee responsible for the particular competition in their various steps from the paper submission to manage and communicate with the competitors, validator(s),judge(s) and also administrator(s). The software will facilitate communication between the competitors, validator(s),judge(s) and administrator(s) through E-mail. Preformatted reply forms are used in every step of the competition. The system also maintains relational database containing a list of competitors, judge(s),validator(s) and administrator(s).

1.3. Glossary

Term	Definition
Active paper	The paper that is tracked by the system, the paper that may or may not be accepted.
Competitor(User)	Person submitting the paper for competition. In case of multiple persons, this refers to the principal one, with whom all the communication is made.
Database	Collection of all information monitored by the system.
Administrator	The person who updates information, check status and publishes results.
Validation	The report giving the green signal or go ahead showing the acceptance of the paper into the next level of competition.
Validator	The person who examines the paper and has ability to recommend approval of a paper into the competition or may reject the paper stating various reason .
Software Requirement Specification	A document that completely describes all the functions.
Judge	The person who finally evaluates the papers and depending on its merit awards it marks or ranks.

1.4 References

[1] SRS Example1
[2] SRS Example2
[3]SRS IEEE 830
[4]Umbrello_ Reference

1.5 Overview of the document

The next section (i.e. section 2) of this document gives the overview of the various functionalities of the system. It also specific the various informal requirements and is used to establish a context with the technical specification mentioned in the next section. The next section (i.e. section 3-Functional Requirement Section) details the various functionalities to be incorporated into the system in technical and is primarily written for the developers.

Both the sections are used to describe the same audience but have been written keeping in mind different audience and hence different languages are used.

2.0 Product perspective

2.0.1 **System interfaces** : Web Browser (Any latest Web Browser Supported)

2.0.2 **User interfaces**:

a) The logical characteristics of each interface between the software and its users

b) All aspects of optimizing interface with the person who must use the system.

Both a , b portions are not yet decided.(Designs are not ready).

2.0.3 **Hardware interfaces :** Mouse , Key-Board , Server.

2.0.4 **Software interfaces :** Ref 2.2 (Software interface included).

2.0.5 **Communications interfaces :** Internet and standard protocols used for communication.

2.1 Overall Description : *System Environment*

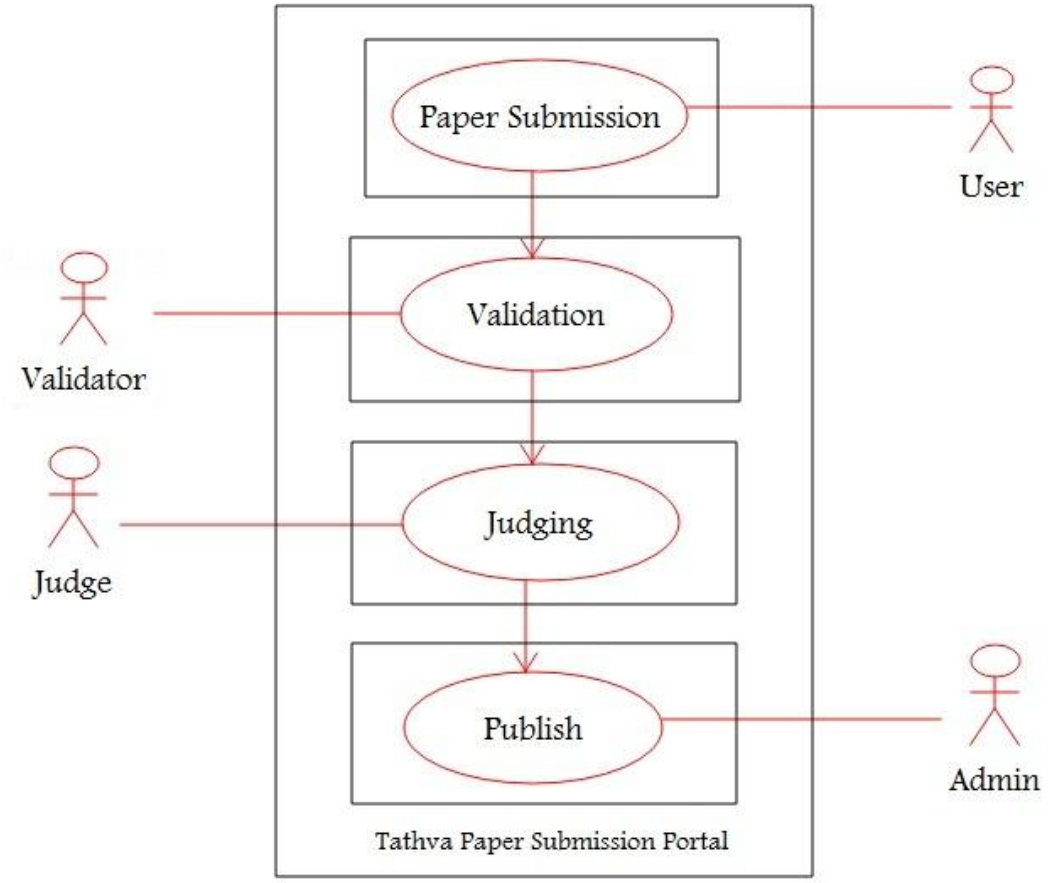


Figure 1 - System Environment

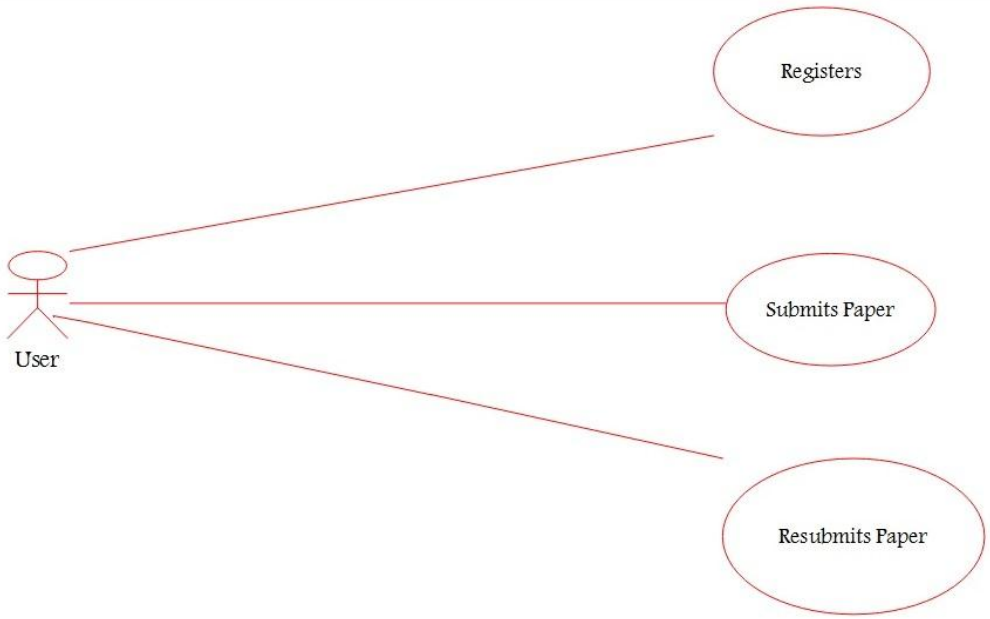
The Tathva Paper Submission Portal has four active actors and one cooperating system. The Admin , Judge , Validator , User accesses the Paper Submission Portal through the Internet. The Admin declares the date of event for paper submission. The User registers himself to take part in the paper submission event . The Validator validates the paper and sends an acknowledgement to the User. If the Paper is valid then it is shortlisted for further judgment. Judges evaluates the papers and declare the top three papers. Admin publish the Results. Only constraint is one user can submit only one paper.

2.2 Functional Requirements Specification

This section outlines the use cases for each of the active **Users** separately.

2.2.1 User Use Case

The user has the following sets of use cases:



Use case: **Register**
Diagram:

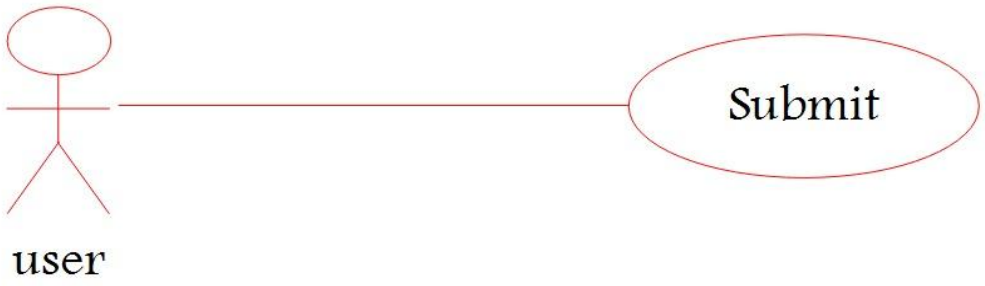
Brief Description
The User Registers himself for the Paper Submission Event.

Initial Step-By-Step Description

1. The User chooses the register option.
2. The system displays the fields to be filled.
3. The User fills up all the fields and clicks the register button.
4. The system checks all the fields and if all are correctly entered the information are stored in database and the User gets registered successfully .
5. The User gets a confirmation in his/her mail id..

Xref: Section 3.1.1, Register

Use case: **Submit**
Diagram:



Brief Description
The User Logs in to his account and submits paper.
Initial Step-By-Step Description

1. The User chooses the topic for the Paper.
2. The User uploads the Paper .
3. The system checks whether the particular USER has already uploaded any valid paper previously.
4. If so then System rejects the paper uploaded and sends an email to the User.
5. Else The System stores the uploaded Paper for further validation checking.
6. The User gets a system generated confirmation mail.
   Xref: Section 3.1.2, Submit

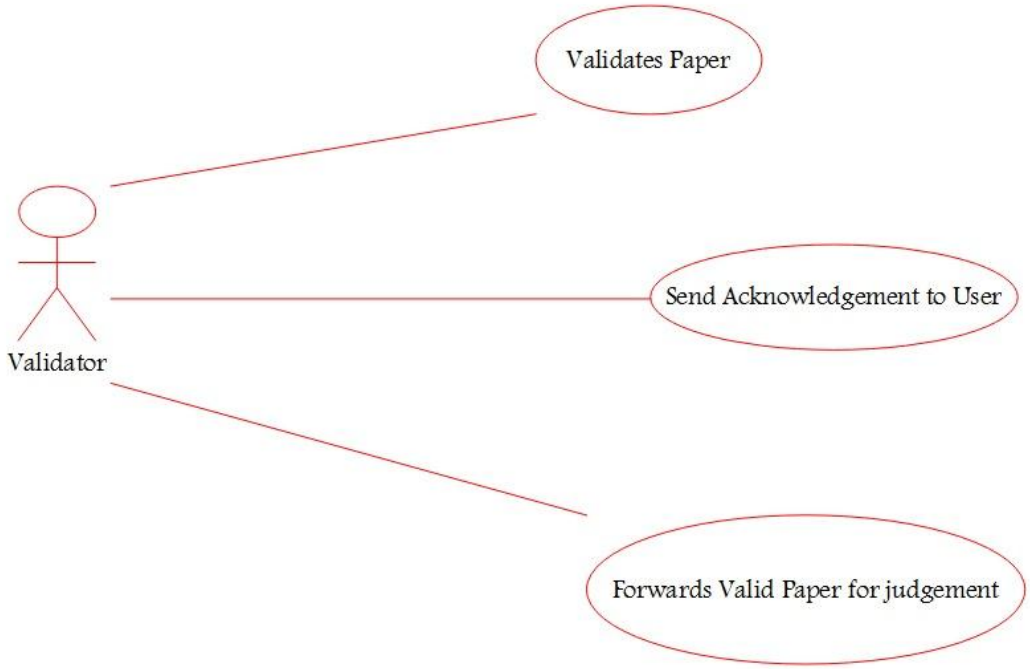
Use case: **Re-Submit**
Diagram:

Brief Description
After getting an email about invalid submission , The User Logs in to his account and can re-submit revised paper.
Initial Step-By-Step Description

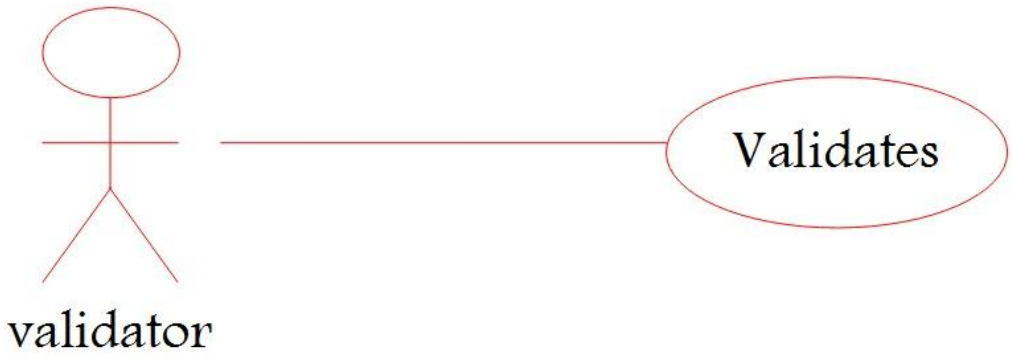
1. The User chooses the topic for the Paper.
2. The User uploads the Paper .
3. The User gets a system generated confirmation mail.
   Xref: Section 3.1.3, Re-Submit

2.2.2 Validator Use Case

The Validator has the following sets of use cases:



Use case: **Validate**
Diagram:

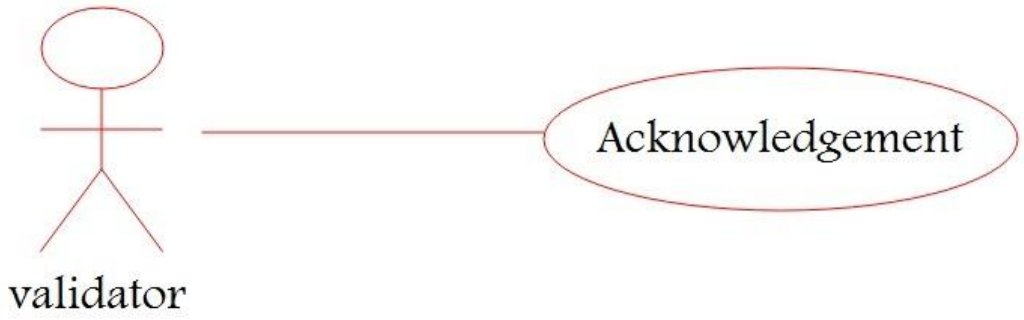


Brief Description
The Validator validates the submitted papers.
Initial Step-By-Step Description
Before this use case can be initiated, the Validator has already connected to the Paper Submission Portal.

1. The Validator chooses the *paper* that has not been checked.
2. The Validator checks plagiarism and other format issues and decides whether the paper submitted is valid .

Xref: Section 3.1.4, Validate

Use case : Acknowledgment
Diagram:



Brief Description
The Validator sends System Generated mail to the User.
Initial Step-By-Step Description
Before this use case can be initiated, the Validator has already connected to the Paper Submission Portal and checked the paper.

1. In both cases (valid or invalid) the Validator sends an acknowledgement mail(generated by system) to the User.

Xref: Section 3.1.5, Acknowledgment

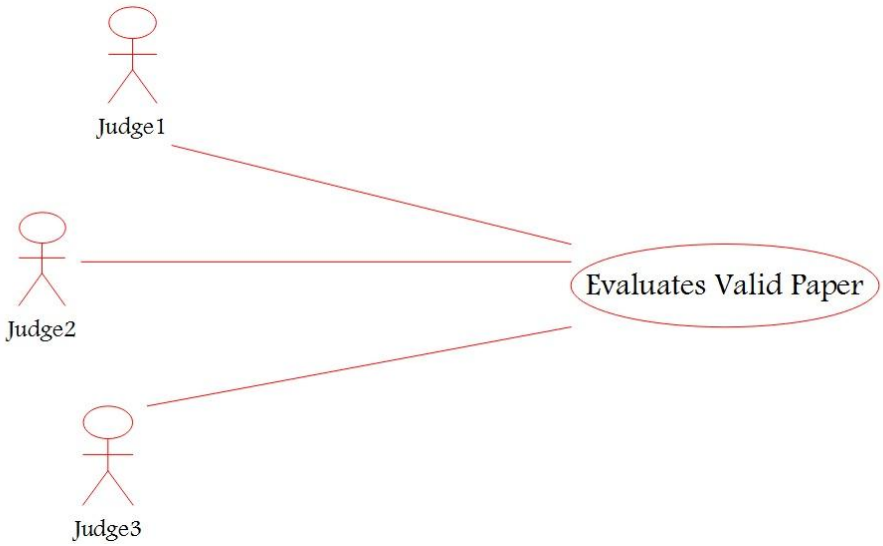
Use case : Forward
Diagram:

Brief Description
The Validator forwards valid papers for judgment.
Initial Step-By-Step Description
Before this use case can be initiated, the Validator has already connected to the Paper Submission Portal , checked the paper and already sent an acknowledgment to the user.

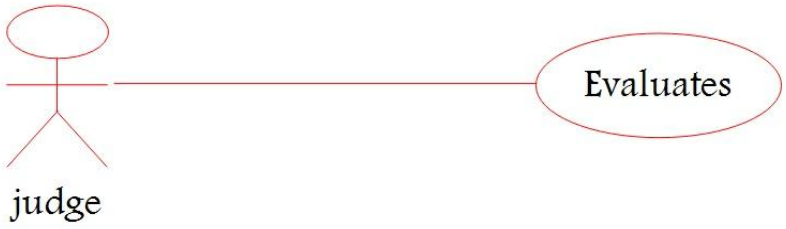
1. If valid then the corresponding papers are forwarded for judgment.

Xref: Section 3.1.6, Forward

2.2.3 Judge Use Case



Use case: Evaluates
Diagram:



Brief Description
The Judge evaluates papers.

Initial Step-By-Step Description
Before this use case can be initiated, the Validator has already validated the papers and the date of submission for papers is over.

1. The Judge evaluates all valid papers and provides scores for them.
2. System automatically adds them up and generates **Result.**

Xref: Section 3.1.7, Judge

2.2.4 Admin Use Cases
The Admin has the following sets of use cases:

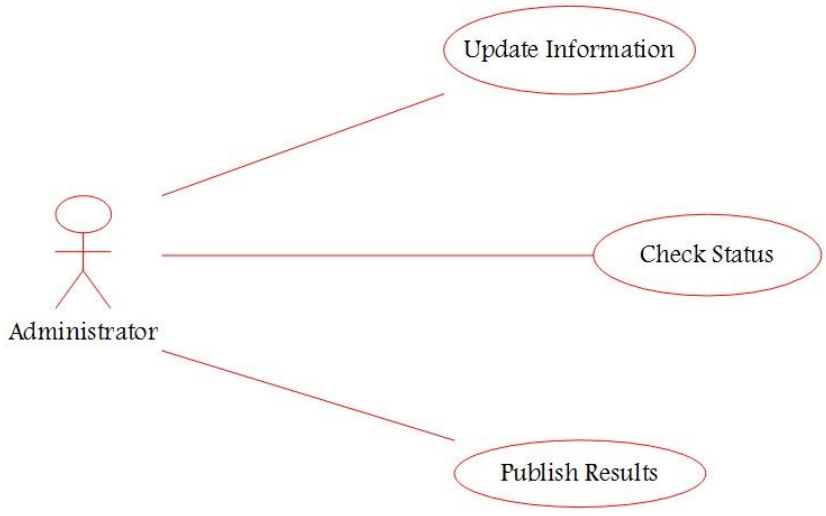


Figure 2 - Admin Use Cases

Update Information use cases
Use case: Update Validator
Diagram:

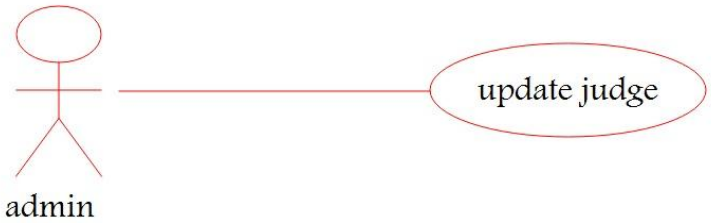
Brief Description
The Admin enters a new Validator or updates information about a current Validator.

Initial Step-By-Step Description
Before this use case can be initiated, the Admin has already accessed the main page of the System Admin.

1. The Admin selects to *Add/Update Validator*.
2. The system presents a choice of adding or updating.
3. The Admin chooses to add or to update.
4. The Admin can access every database and hence updates database accordingly.

Xref: Section 3.1.8, Add Validator; Section 3.1.9 Update Person

Use case: Update Judge
Diagram:



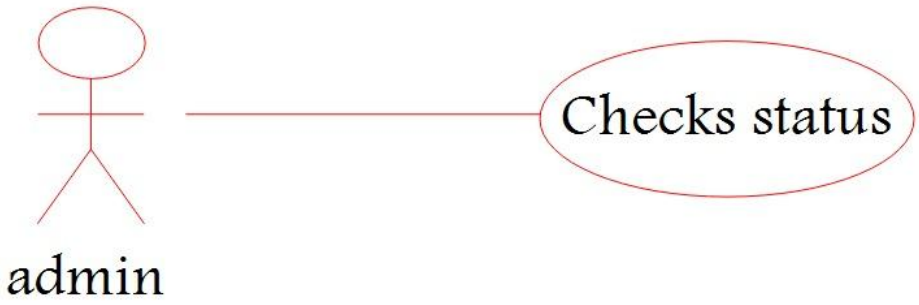
Brief Description
The Admin enters a new Judge or updates information about a current Judge.

Initial Step-By-Step Description
Before this use case can be initiated, the Admin has already accessed the main page of the System Admin.

1. The Admin selects to *Add/Update Judge*.
2. The system presents a choice of adding or updating.
3. The Admin chooses to add or to update.
4. The Admin can access every database and hence updates database accordingly.

Xref: Section 3.1.10, Add Judge; Section 3.1.9 Update Judge

Check Status use case:
Use case: Check Status
Diagram:

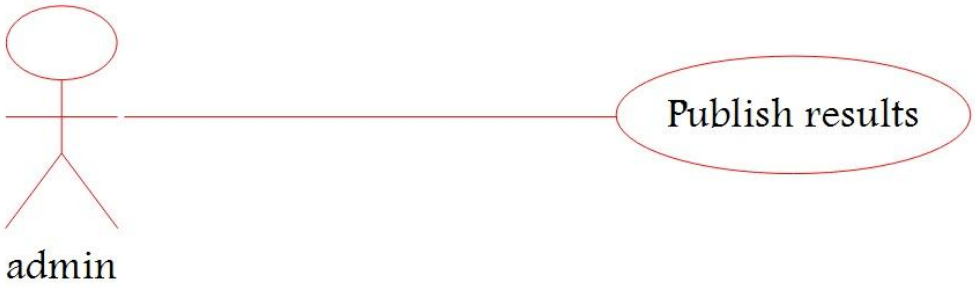


Brief Description
The Admin checks status of the current Paper submission event.
Initial Step-By-Step Description
Before this use case can be initiated, the Admin has already accessed the main page of the System Admin.

1. The Admin checks whether there are any problems regarding the procedures.
2. The Admin decide when to start or stop a full event.

Xref: Section 3.1.11 Check Status

Publish use case:
Use case: Publish
Diagram:



Brief Description
The Admin publish results of the current Paper submission event.
Initial Step-By-Step Description
Before this use case can be initiated, the Judges have already declared the results.

1. Admin access the Judge database
2. Admin publish the result in the portal.
3. Winners receive mail in their email-ids.

Xref: Section 3.1.11 Publish

State Transition diagram for Paper Submission Process

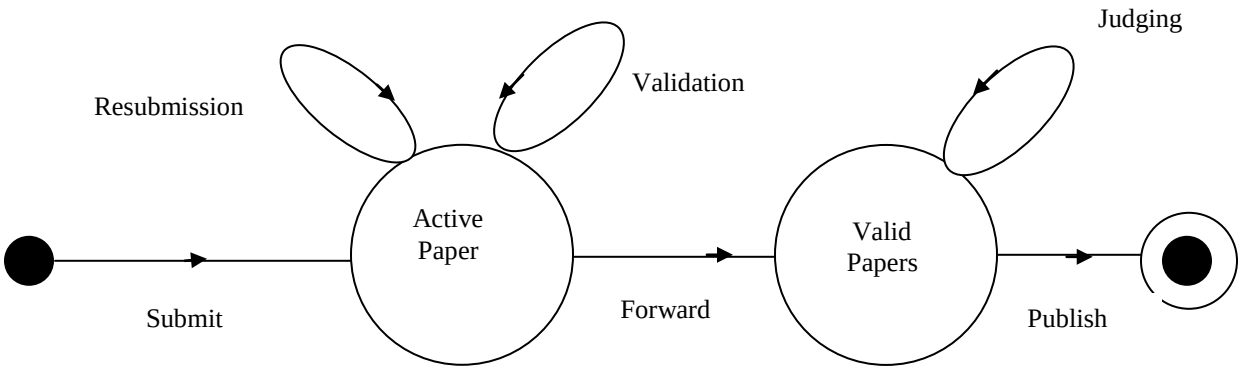


Figure 3 - Paper Submission Process

The *Paper Submission Process* state-transition diagram summarizes the use cases listed below. A User submits a paper for consideration. The Validator validates the Active paper . If the paper is found to be invalid then User has to re-submit . If the paper is valid the validator forwards the Active paper for further Judgment. At least three validators and three Judges should be there. Judge evaluates the valid papers and at the end of the event adds up the marks. Admin publishes the result in the website. Admin has the highest authority. He can access any database and delete any user at any-time(Not mentioned in the state diagram)

2.3 User Characteristics

The User is expected to be Internet literate and be able to use a search engine. The main screen of the Paper Submission Portal will have login/register option and a news area where updates regarding the event will be put up.

The Validator , Judge and Admin are expected to be Internet literate .

The detailed look of these pages is discussed in section 3.1 below.

2.4 Constraints

2.4.1 Bandwidth Constraints

- The Paper Submission Portal will be on a server with high speed Internet capability.
- Assuming that Internet is of high bandwidth , recommended no of competitors to be able to access the system at a time is 100. The reason is updating the database is a complex action and so many concurrency issues might occur.
- The software developed here assumes the use of a tool such as Apache for connection between the Web pages and the database.

2.4.2 Space Constraints

- The storage space required for this System to run including Database size is within 50MB. But Uploaded papers will require considerable amount of storage space which is depended on the number of competitors.
- All the databases are centralized and will be accessible from outside also.

2.4.3 Type Constraints

- All uploaded papers should be of pdf(Portable Document Format) format.

2.5 Assumptions and Dependencies

- As it is web based application so there is no Operating System dependencies.
- As it is a web based system so it can be hosted both in linux or windows based server.

3.0. Requirements Specification

3.1 Functional Requirements

The Logical Structure of the Data is contained in Section 3.2.1.

3.1.1 Register

Use Case Name	Register
XRef	Section 2.2.1, Register
Trigger	The User assesses the Paper Submission Portal for the first time.
Precondition	The User is not already registered.
Basic Path	1. The User chooses the register option. 2. The system displays the fields to be filled. 3. The User fills up all the fields and clicks the register button. 4. The system checks all the fields and if all are correctly entered the information are stored in database and the User gets registered successfully. 5. The User gets a confirmation in his/her mail id.
Alternate Path	In step 4, if all the fields are either not entered or are not correctly entered; the user is asked to fill in the details.
Postcondition	The person is registered.
Exception Paths	If the user is already registered the use case is abandoned. The person may choose not to register at any time.
Other	None

3.1.2 submit

Use Case Name	Submit
XRef	Section 2.2.1. Submit paper
Trigger	The user uploads his paper and submits it.
Precondition	The user is already registered.
Basic Path	This use case uses the *input* HTML tag with type=submit. This uploads the paper in the server.
Postcondition	The paper is uploaded.
Exception Paths	If the user has already submitted a valid paper the use case is abandoned. The attempt may be abandoned at any time.
Other	None

3.1.3Resubmit

Use Case Name	Resubmit
XRef	Section 2.2.1,,resubmit paper
Trigger	The user uploads his paper and submits it for resubmission.
Precondition	The user is registered and he either has not submitted any paper or his paper has been returned as not valid.
Basic Path	1. The User chooses the topic for the Paper.

	2. The User uploads the Paper . 3. The User gets a system generated confirmation mail.
Alternative Paths	None
Postcondition	The user's paper is uploaded into the server.
Exception Paths	The User may abandon the operation at any time.
Other	None

3.1.4 Validator

Use Case Name	validator
XRef	Section 2.2.2 validator
Trigger	The validator validates the paper.
Precondition	The paper has been submitted by a user.
Basic Path	1. The Validator chooses the *paper* that has not been checked. 2. The Validator checks plagiarism and other format issues and decides whether the paper submitted is valid
Alternative Paths	None.
Postcondition	The validator has validated the paper.
Exception Paths	The Validator may abandon the operation at any time.
Other	The Validator information includes paper id, Id and valid bit , checked bit.

3.1.5 Acknowledgment

Use Case Name	Acknowledgement
XRef	Sec 2.2.2,acknowledgment
Trigger	The validator sends his acknowledgment.
Precondition	The Validator has validated the paper.
Basic Path	1. In both cases (valid or invalid) the Validator sends an acknowledgement mail(generated by system) to the User.
Alternative Paths	None.
Postcondition	The user has been sent an acknowledgement.
Exception Paths	None.
Other	None.

3.1.6 Forward

Use Case Name	Forward
XRef	Section 2.2.2,forwards
Trigger	The Validator forwards and the valid paper ids are inserted in the judges table.
Precondition	The Validator has already connected to the Paper Submission Portal , checked the paper and already sent an acknowledgment to the user.
Basic Path	1.If valid then the corresponding papers are forwarded for judgment.
Alternative Paths	If the paper is not valid; then a negative acknowledgement is returned to the user.

Postcondition	The paper has either been forwarded to Judge or has been returned to the user.
Exception Paths	None.
Other	None.

3.1.7 Judge

Use Case Name	Judge
XRef	Section 2.2.3,judge
Trigger	The judge evaluates the paper.
Precondition	The Validator has already validated the paper and forwarded to Judge.
Basic Path	1. The Judge evaluates all valid papers and provides scores for them. 2. System automatically adds them up and generates result.
Alternative Paths	None
Postcondition	The paper is evaluated.
Exception Paths	The Judge may abandon the operation at any time.
Other	None

3.1.8 Admin

Use Case Name	Add Validator
XRef	Section 2.2.4, Update information; update validator.
Trigger	The admin chooses the add validator option.
Precondition	The Admin is already registered and has logged in.
Basic Path	1. The Admin selects to *Add/Update Validator*. 2. The system presents a choice of adding or updating. 3. The Admin chooses to add . 4. The Admin can access every database and hence updates database accordingly.
Alternative Paths	The admin chooses to add judge option.
Postcondition	The Validator database has been updated.
Exception Paths	The Admin may abandon the operation at any time.
Other	This use case extends *3.1.10, Add Judge.*

3.1.9 Admin

Use Case Name	Add Judge
XRef	Section 2.2.4,add judge.
Trigger	The Admin chooses Add Judge option.
Precondition	The Admin is already registered and has logged in.
Basic Path	1. The Admin selects to *Add/Update Judge*. 2. The system presents a choice of adding or updating. 3. The Admin chooses to add. 4. The Admin can access every database and hence updates database accordingly.

Alternative Paths	None.
Postcondition	The judge database has been updated.
Exception Paths	The Admin may abandon the operation at any time.
Other	None.

3.1.10 Check status

Use Case Name	Send Communication
XRef	Section 2.2.4, check status
Trigger	The admin chooses the check status option.
Precondition	The Admin has already logged into the account.
Basic Path	1. The Admin checks whether there are any problems regarding the procedures. 2. The Admin decide when to start or stop a full event.
Alternative Paths	None.
Postcondition	The Admin has received the present status.
Exception Paths	The Admin may abandon the operation at any time.
Other	None.

3.1.11 Publish Results

Use Case Name	Publish
XRef	Section 2.2.4, Publish Result
Trigger	Admin access the Judge's database and publishes the result.
Precondition	The Judge has evaluated all the papers.
Basic Path	1. Admin access the Judge database 2. Admin publish the result in the portal. 3. Winners receive mail in their email-ids.
Alternative Paths	None.
Postcondition	The result has been published and winners have been sent e-mails.
Exception Paths	None.
Other	None.

3.2 Detailed Non-Functional Requirements

3.2.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal databases is given below.

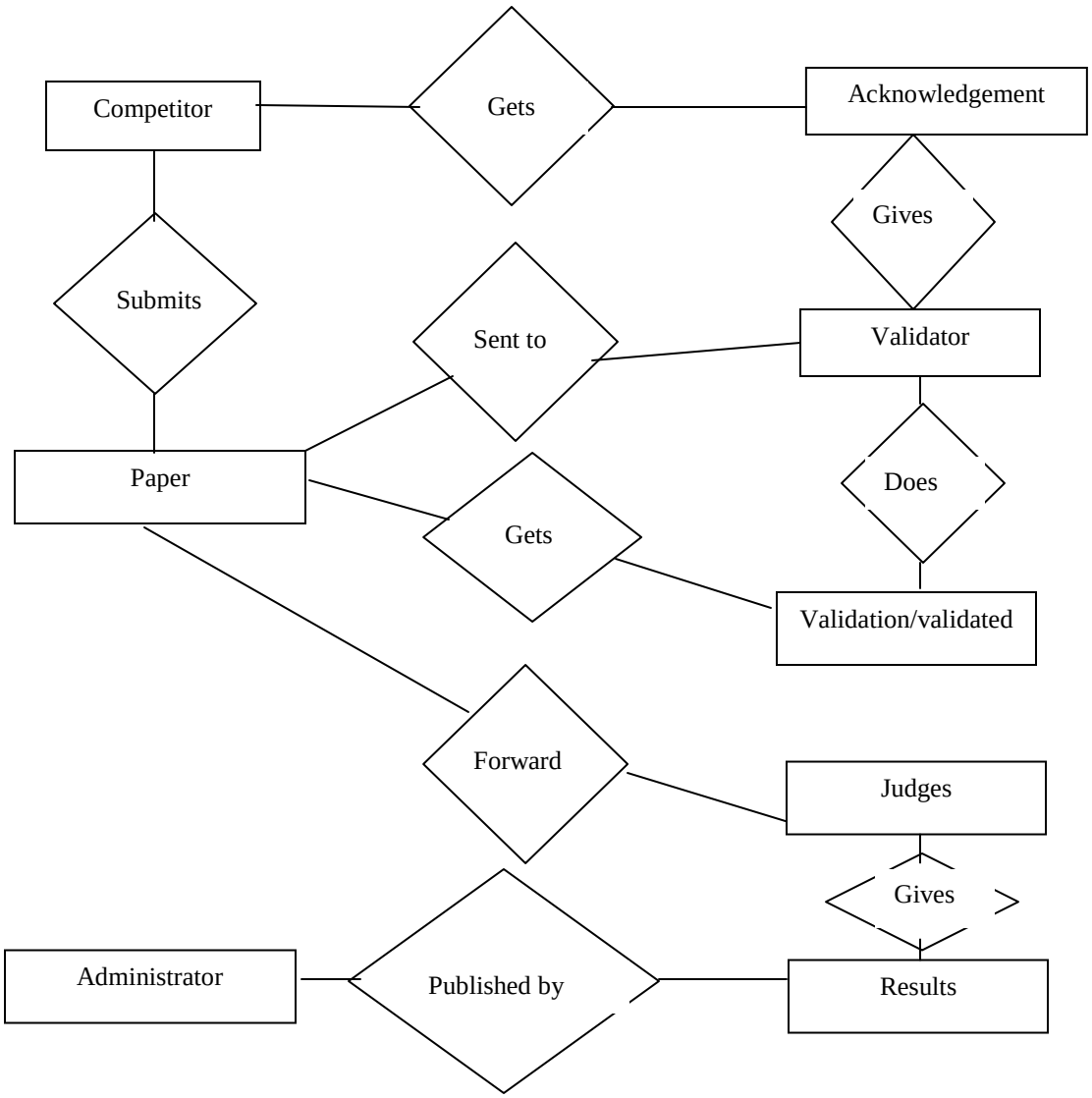


Figure 2 - Logical Structure of the Paper Submission Portal Data

Databases Used : Yet to be Designed.

3.2.2 Security

- The server on which the Online System resides will have its own security to prevent unauthorized *write*/*delete* access.
- There is no restriction on *read* access.

- Security regarding the write access is kept in mind while constructing the system itself. The Validator can write multiple databases while the judge can write only in the database dedicated to him/her. The Admin can access and change any database at any time.
- Concurrency control.
- No User can access other's User's account(Cannot update or delete other's information)

3.2.3 Others Non – Functional Requirements:

- Reliability :
    - The system should be online 24 hours during the time of event.
    - The system can provide service to at least 50 users at a time
    - Security (ref: 3.2.2)
- Efficiency :
    - Uploading of paper must be fast enough.
    - Storing paper in server without storing them in Database itself will help to achieve efficiency
- Portability :
    - As it a web based system so transferring the system from one server to another if required shouldn't be a problem.
- Maintainability :
    - Admin should find it easy to maintain and manage.
    - System should be Flexible enough to add other features (Ref : 3.3)

- Usability :
    - User should find it easy during registration and login.

3.3 System Evolution

In the future this system

- Will provide a more secured way of registration i.e Email Validation via link.
- Will provide a secured login.
- Validation process will be fully automated and will include plagiarism checker.
- Text message alerts will be included.