# HOADIS: Hypermedia Open Architecture Departmental Information System

**Project Report – Software Lab**
**B.Tech Part II(IDD), IV Semester 2008-09**



**Submitted by:**

Rohit Yadav (07020003) Ajay Chhatwal (07020002)

**Project Guide:**

Dr. A.K. Tripathi
*(Professor)*
*DEPARTMENT OF COMPUTER SCIENCE AND ENGINEEERING*
*INSTITUTE OF TECHNOLOGY*
*BANARAS HINDU UNIVERSITY*
*VARANASI*


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**INSTITUTE OF TECHNOLOGY**
**BANARAS HINDU UNIVERSITY**
**VARANASI-221005, (U.P.)**
**INDIA**


# CERTIFICATE

This is to certify that **Mr. Rohit Yadav and Mr. Ajay Chhatwal,** students of Department of Computer Science and Engineering, Institute of Technology BHU, Varanasi have worked on the project **"Departmental Information System"** under my direct supervision and guidance for the 4th semester of B.Tech (IDD) Part II program 2008-09, the findings of which have been incorporated in this report. They have worked diligently, meticulously and methodically. The report submitted by them embodies the original work done by them during the development of the project.

**Dr. A. K. Tripathi**
Professor
Department of Computer Science & Engineering
Institute Of Technology
Banaras Hindu University

# ACKNOWLEDGEMENT

We would like to convey our deepest gratitude to **Dr. A.K. Tripathi**, Professor, Computer Science Department, Institute of Technology, B.H.U., who guided us throughout the project. His superb guidance, and constant encouragement are the motive force behind this project work.

We would also like to thank **Dr. A. K. Agrawal**, Head of the Department, Computer Science and Engineering, Institute of Technology, BHU, for allowing us to avail all the facilities of the Department necessary for this project.

We are very thankful to all the technical and non-technical staff of the Department of Computer Engineering for their assistance and co-operation.

**Rohit Yadav** 07000003

**Ajay Chhatwal** 07000002

B. Tech (IDD) Part II

Computer Science and Engineering

Institute of Technology

Banaras Hindu University,

Varanasi - 221005

# Table of Contents

# ABSTRACT

We describe here the design and development of a hypermedia based open architecture departmental information system (HOADIS) for an educational institution.

**HOADIS is intended to be a flexible, extensible and seamless hypermedia information system that allows students to find up-to-date information on courses, teachers, announcements, events, lectures etc**. It will also enable the teachers and students to maintain **online attendance records, online notice board, online assignment submission, examination results** etc.

***KEYWORDS***

*Information System, Data Processing, Information Technology, MVC, Open Architecture, Rich Internet Application, Hypermedia.*

# 1. Introduction

It is important for university students to have information about their department. This information should be up to date, and easily accessible, and digestible. When students have a question, they should have the means to find their own answers.

Printed material such as the *Schedule of Classes* describing when and where courses will be taught this term and the *Course Bulletin* describing all the courses offered by the department are very useful, but are only updated once a term or once a year. Augmenting this printed material with an interactive computerized information system allows the student to get up to date information, not only about the current term, but also about current plans for future terms.

Besides information on courses and instructors, the students need a central location to find information about activities in the department such as speakers, presentations, and events. This allows the student to become more actively involved in the department. Making all this information available on-line saves on printing costs and frees the human resources to deal with less routine matters. [1]
To satisfy this need in our department we have designed and implemented the hypermedia open architecture [2]-[4] departmental information system: **HOADIS**.

## 1.1 Theoretical Foundations

A theory of information systems needs to consider the role of information and information systems in organizations. In the theory proposed by Langefors [9] [10] it is a fundamental assumption or choice that organizations can be considered as simple cybernetic networks similar to mechanical systems or machines.

Accordingly an information system is viewed as an instrument for controlling and managing another system or activity, with the aim of achieving optimal utilization of resources and maximum economic returns for the organization as a whole.

Langeforsian approach aims to introduce effective methods for information system design and to utilize the potential of computers and information technology in controlling organizations and improving their performance. [8] The assumptions are: first, it must be possible to define goals, precisely and operationally, for the operation of the system; further, the system must be controllable and able to react in predictable and deterministic ways; and finally, it must be possible and feasible to formularize controlling a system.

Langefors [12] recognized that it is never possible to satisfy the assumptions completely. Still he believed that in most cases it will be possible to obtain a good

enough approximation to these assumptions and therefore it will be possible to design information systems according to the principles of the theory of information systems.

Langefors [11] proposed that information system work be divided into a limited number of method categories. Each category includes distinct tasks requiring different skills and therefore making it possible to obtain economics of specialization. [11]

The work of Langefors has had decisive impact on ISD (Information System Design) by consistently insisting on theoretical work as a fundamental background for ISD. Langefors suggests a simple differentiation of ISD into five method areas, which are systematic but not chronologic (Fig. 1).

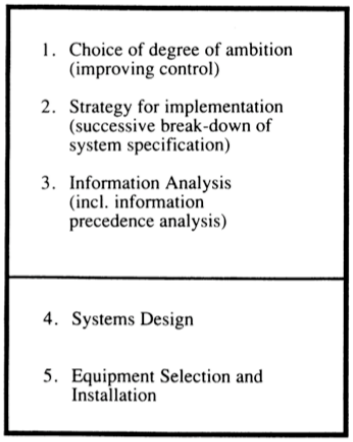



**Figure 1. Method areas in analysis and design of information systems, according to Langeforsian approach[11]**

The major shortcoming of the Langeforsian approach can be attributed to the choice of mechanical system models for organizations.

## 1.2 Architectural Description

## Pluggable and Open Architecture

HOADIS has a pluggable and open architecture. To add a module the administrator simple has to do the following:

Add the name and URL of the module in the links table along with the access privilege for the module. The link is automatically displayed in the side navigator for all user roles having the required access privilege. There is no limitation on the type of module. It may even be hosted externally and may use any kind of framework and development environment. The only constraint is that it should be hypermedia based i.e. it should use HTTP protocol.

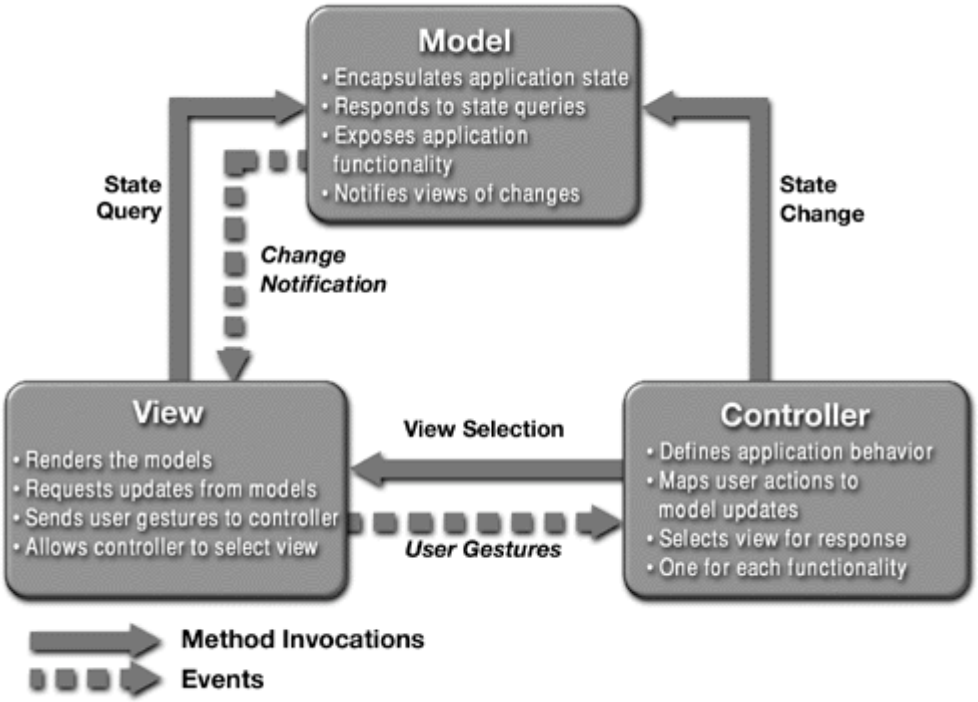
## MVC Architecture

We have used the model-view-control architecture in the design of HOADIS. The Model-View-Controller architecture is a widely used architectural approach for interactive applications. It divides functionality among objects involved in maintaining and presenting data to minimize the degree of coupling between the objects. The architecture maps traditional application tasks--input, processing, and output--to the graphical user interaction model. It also maps into the domain of multitier Web-based enterprise applications.

MVC organizes an interactive application into three separate modules: one for the application model with its data representation and business logic, the second for views that provide data presentation and user input, and the third for a controller to dispatch requests and control flow.

The MVC design pattern provides a host of design benefits. MVC separates design concerns (data persistence and behaviour, presentation, and control), decreasing code duplication, centralizing control, and making the application more easily modifiable. MVC also helps developers with different skill sets to focus on their core skills and collaborate through clearly defined interfaces. For example, a web application project may include developers of custom tags, views, application logic, database functionality, and networking. An MVC design can centralize control of such application facilities as security, logging, and screen flow. New data sources are easy to add to an MVC application by creating code that adapts the new data source to the view API. Similarly, new client types are easy to add by adapting the new client type to operate as an MVC view. MVC clearly defines the responsibilities of participating classes, making bugs easier to track down and eliminate.

Most Web-tier application frameworks use some variation of the MVC design pattern. The view is the actual HTML or XHTML page, and the controller is the code that gathers dynamic data and generates the content within the HTML or XHTML. Finally, the model is represented by the actual content, which is often stored in a database or in XML nodes, and the business rules that transform that content based on user actions.




The MVC architecture divides applications into three layers--model, view, and controller--and decouples their respective responsibilities. Each layer handles specific tasks and has specific responsibilities to the other areas.

- **Model**
  A model represents business data and business logic or operations that govern access and modification of this business data. Often the model serves as a software approximation to real-world functionality. The model notifies views when it changes and provides the ability for the view to query the model about its state. It also provides the ability for the controller to access application functionality encapsulated by the model.

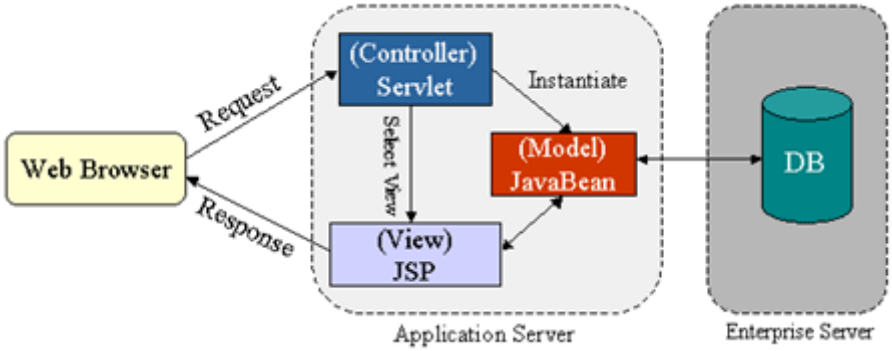
- **View**
  A view renders the contents of a model. It accesses data from the model and specifies how that data should be presented. It updates data presentation when the model changes. A view also forwards user input to a controller.

- **Controller**
  A controller defines application behaviour. It dispatches user requests and selects views for presentation. It interprets user inputs and maps them into actions to be performed by the model. In a Web application, they are HTTP

GET and POST requests to the Web tier. A controller selects the next view to display based on the user interactions and the outcome of the model operations. An application typically has one controller for each set of related functionality. Some applications use a separate controller for each client type, because view interaction and selection often vary between client types.

**Use of MVC in HOADIS**

We have used the MVC architecture in HOADIS using the following J2EE specification architectural model (known as the Model2 architecture in J2EE nomenclature):




**Model2 Architecture**

**Model**

The model is a collection of Java beans that are used to store data and can be associated with HttpServletRequest or HttpSession objects.

**View**

JavaServer Page represents the view, with data being transported to the page in the HttpServletRequest or HttpSession. It can retrieve information from the Java beans associated with the HttpServletRequest or HttpSession objects.

**Controller**

The controller servlet communicates with the database, placing data in the Java beans and then loads the HttpServletRequest or HttpSession with appropriate Java beans. Then, it forwards the HttpServletRequest and HttpServletResponse to the JSP using a RequestDispatcher.

The Servlet is a Java class, and it communicates and interacts with the model but does not need to generate HTML or XHTML output; the JSPs do not have to communicate with the model because the Servlet provides them with the information—they can concentrate on creating output.

## 1.3 References


[1] A. Johnson, F. Fotouhi, N. Goel, R. Weinand, and J. Lechvar, "CASPER: A Hypermedia Departmental Information System" *IEEE Trans. Education*, vol.39, no. 4, pp. 471–477, Nov. 1996.

[2] J. Conklin, "Hypertext: An introduction and survey" *IEEE Trans. Comput*., vol. 20, no. 9, pp. 17–41, Sept 1987.

[3] J. Nielsen, "Hypertext and Hypermedia", San Diego: Academic, 1990.

[4] I. Tom& S. Khan, T. Muldner, M. Nassar, G. Novak, and P. Proszynski, "Hypermedia-Introduction and survey," *J. Microcomput. Applicat*., vol. 14, pp, 63-103.

[5] J. J. Cimino, S. Socratous , "Just tell me what you want!: the promise and perils of rapid prototyping with the World Wide Web." , *J Am Med Inform Assoc. 1996*;3:719-23 (supl).

[6] J. J. Cimino,1 S. Sengupta, 1 Paul D. Clayton, V. L. Patel, A. Kushniruk, X. Huang, "Architecture for a Web-Based Clinical Information System that Keeps the Design Open and the Access Closed".

[7] Edmond Mesrobian, Richard Muntz, Eddie Shek, Silvia Nittel, Mark LaRouche, and Marc Kriguer, "OASIS: An Open Architecture Scientific Informatioin System". *IEEE Trans. Education*, July 1996.

[8] A. Olerup, "Design Approaches: A Comparative Study of Information System Design and Architecture Design".

[9] B. Langefors, "Some Approaches to the Theory of Information Systems." *BIT,* 3(1963).

[10] B. Langefors, "Theoreotical Analysis of Information Systems." 4th Edn(1st Edn 1966). *Studentlitteratur*, Lund (1973).

[11] B. Langefors, "Management information system design." *IAG Journal* (Amsterdam), 2 (4) (1969).

[12] B. Langefors, “System for Foretagsstyrning[Systems for management control]”, 2nd edn. *Studentlitteratur,* Lund (1970).

# 2. Analysis

In order to analyze the project we worked out on Software Requirements Specification (SRS) that enabled us to work out on designing a robust solution.

## 2.1 SRS: IEEE Format

Software Requirement Specification follows next page where we describe and discuss the requirements of the project and explore the technologies to be used.

# HOADIS

## Software Requirements Specification


**Developed By**

Ajay Chhatwal (07020002)
Rohit Yadav (07020003)

**Project Guide**

Dr. A. K. Tripathi
Professor
Department of Computer Science & Engineering
Institute Of Technology
Banaras Hindu University

# Table of Contents

# 1. Introduction

## 1.1 Purpose:

To develop a web enabled Information System for educational institution using hypermedia and employing an open architecture.
It is intended to be a hypermedia information system that allows students to find up-to-date information on courses, teachers, announcements, events etc.
It would also enable the teachers to keep attendance records, calculate grades etc.

## 1.2 Scope:

- *Ease of Use*
  The system is intended for users with varying levels of computer training and experience. It will, therefore, have a user interface that provides a consistent look-and-feel as the user and is easy to use.
- *Multiple Functions*
  Within the main functions of HOADIS we envisioned multiple subordinate functions, some of which will be designed locally and some of which will be drawn from a variety of Web-based resources.
- *Extensible and Open Architecture*
  We desire an architecture which will allow us to add components as quickly as we think of them. Open architecture allows people to write modules without having to go through the whole information system.
- *Access over the network*
  The system will be accessible over the intranet to ensure ease of access to the users.
- *Multiple Access Levels*
  The different types of users (Students, Teachers, Alumni, Non-teaching Staff, Guest and Administrator/Super User) with different user privileges will be using the system. Therefore, the system will have different access levels for different users.
- *Security*
  Our desire to operate in the Web environment complicates security tasks. We use the traditional approach of having users access the application through an identification-authentication-authorization process once at the beginning of the session. Coupled with this is the need to keep track of the user access level in order to determine the modules which a user is allowed to access.
- *Platform Independence*

Motivated by the premise that heterogeneity of software applications and hardware systems is here to stay we design the system without any restriction on the platform being used and it will use cross-platform technologies wherever possible.

### 1.3 Definitions, Acronyms and Abbreviations

**HOADIS:** Hypermedia Open Architecture Departmental Information System.

**Information System:** An information System is a system of persons, hardware, software, communication devices, network and data resources that processes (can be storing, retrieving, transforming information) data and information for a specific purpose and activities that process the data and information in an organization, and it includes the organization's manual and automated processes.

**Hypermedia:** An extension to hypertext providing multimedia facilities, such as sound and video which allows the information to be processed in a non-linear manner.

**Open Architecture:** Open architecture is a type of software architecture that allows adding, upgrading and swapping components.

**Superusers:** Users with high access priviledges; they have power to read and write into databases and do maintanence and upgradation.

**HTML:** Hypertext Markup Language

**PHP:** Preprocessed Hypertext P

**JSP:** Java Server Pages

### 1.4 References

[1] A. Johnson, F. Fotouhi, N. Goel, R. Weinand, and J. Lechvar, "CASPER: A Hypermedia Departmental Information System" *IEEE Trans. Education*, vol.39, no. 4, pp. 471–477, Nov. 1996.

[2] J. Conklin, "Hypertext: An introduction and survey" *IEEE Trans. Comput*., vol. 20, no. 9, pp. 17–41, Sept 1987.

[3] J. Nielsen, "Hypertext and Hypermedia", San Diego: Academic, 1990.

[4] I. Tom& S. Khan, T. Muldner, M. Nassar, G. Novak, and P. Proszynski, "Hypermedia-Introduction and survey," *J. Microcomput. Applicat*., vol. 14, pp, 63-103.

[5] J. J. Cimino, S. Socratous , “Just tell me what you want!: the promise and perils of rapid prototyping with the World Wide Web.” , *J Am Med Inform Assoc. 1996*;3:719-23 (supl).

[6] J. J. Cimino,1 S. Sengupta, 1 Paul D. Clayton, V. L. Patel, A. Kushniruk, X. Huang, “Architecture for a Web-Based Clinical Information System that Keeps the Design Open and the Access Closed”.

[7] Edmond Mesrobian, Richard Muntz, Eddie Shek, Silvia Nittel, Mark LaRouche, and Marc Kriguer, “OASIS: An Open Architecture Scientific Informatioin System”. *IEEE Trans. Education*, July 1996.

[8] A. Olerup, “Design Approaches: A Comparative Study of Information System Design and Architecture Design”.

[9] B. Langefors, “Some Approaches to the Theory of Information Systems.” *BIT,* 3(1963).

[10] B. Langefors, “Theoreotical Analysis of Information Systems.” 4th Edn(1st Edn 1966). *Studentlitteratur*, Lund (1973).

[11] B. Langefors, “Management information system design.” *IAG Journal* (Amsterdam), 2 (4) (1969).

[12] B. Langefors, “System for Foretagsstyrning[Systems for management control]”, 2nd edn. *Studentlitteratur,* Lund (1970).

[13] IEEE SRS Format.

[14] www.wikipedia.org – Online Encyclopaedia.

### 1.5 Technologies to be used

- HTML, CSS
- Javascript, AJAX
- Java Server Pages (JSP) & Servlets
- MySQL

### 1.6 Overview

Rest of SRS contains the general description of HOADIS.

# 2. General Description

## 2.1 Product Perspectives

The product is independent and totally self-contained.

## 2.2 Product Functions

- To include generic information about an educational institution like courses offered, faculty, students, research, facilities, placement etc.
- To provide a calendar for planning a semester/year that includes information about exams, results, holidays, events and other relevant information planned.
- To provide a notifier that will notify users about upcoming events, announcements, and any relevant information.
- To store data about past, present and future events and activities that will be mapped with the calendar module. To provide an Online Notice Board
- To provide a Scheduler/Timetable which will store and display timetables for various batches and be used to add new information like events (seminars, presentations, extra classes, holidays), message, competitions, etc.
- To store attendance records of all the students in current session and automatically generate blacklist of students, who might fall short in attendance, in advance.
- To provide a Grade Calculator to generate grades based on rules according to relative or absolute grading systems.
- To facilitate declaration of results of examinations, competitions etc.
- To provide Online Courseware
- To provide a Gallery containing organized collection of media (videos, audio, photos) featuring the institution, people and events.
- To provide a platform for various groups/community/societies in the institution like ACM Chapter, IEEE Chapter, Linux User Group etc.
- To provide for searching data in the system.
- To provide an Institutional Blog
- To provide Web-Space to Every user so that Students, Teachers and alumni can have web space for their sites featuring: profile, brief resume, website/blog link, projects, interests etc.

## 2.3 User Characteristics

The users of HOADIS would comprise of students, teachers, office staff, alumni as well as guest users. It is assumed that users would have basic proficiency in operating computers and would have a working knowledge of English language. Also, the maintainers of the system (administrators) would have a working knowledge of databases and system maintenance.

### 2.4 General Constraints

- GUI is completely in English.
- HOADIS requires a webserver preferably Apache2, MySQL server(>version 5.0), PHP server(>version 5.1).
- Some Javascript functions can fail in Internet Explorer.

The data transfer rate between the system and remote users will be constrained by the capabilities of the networking infrastructure and hardware used in the institution.

### 2.5 Assumptions and Dependencies

- This software will be designed according to the current versions of hypertext and hypermedia communication.
- The availability of a database management system on the server has been assumed.
- Availability of a web-server with support (through plug-ins, if required) for server-side programming tools like JSP, Servelets and PHP etc. has been assumed.

## 3. The Specific Requirements

### 3.1 Information

This section features information required by software developer as part of specific requirement.

#### 3.1.1 Functional Requirements

Following functional are required for HOADIS:

**a. Account Management:**

1. Introduction: Login/Register/Profile module requires a method to interact with database to add and access user information to allow login in and sign ups.

2. Inputs:
- Username
- Password
- Account Type: Student, Teacher, Staff, Alumni, Guest, Admin
- For Profiles: Address, Roll No., Interests, Date of Birth, Gender, Security Question, About, Email, Year, Contact Number.

3. Processing: Processing is done by PHP/JSP script which fetches data from forms using POST/GET methods. Super users(Admin, Staff and Teachers) have power to alter information, ban/sanction new accounts.

4. Outputs: User gets registered/logged in. The information is stored in a SQL database.

**b. Information Method:**

1. Introduction: A set of generic information produced dynamically by mouse event over hypertext.

2. Inputs:
- Mouse event over hyperlink.

3. Processing: Browser opens up the file linked with the hyperlink.

4. Outputs: The required information is displayed on the client area.

**c. Event Method:**

1. Introduction: All the set of events: exams, seminars, competitions etc. will be dealed.
2. Inputs:
- User types: Admin, Staff, Teachers can add things on Online NoticeBoard/new events.
- Notifier automatically gets input of newly added information.

3. Processing: An event manager manages all the event information and sends updates to users via notifier.

4. Outputs: The required information is displayed through notifier and events section of HOADIS.

**d. Result Declaration:**

1. Introduction: This routine automates result declaration.
2. Inputs:
    - Super users input data or result.

3. Processing: A PHP/JSP based sever script is employed to calculate grades etc. and result put in a database.

4. Outputs: The required information is stored in a database and can be accessed by a user.

**e. Attendance Recorder Routine:**

1. Introduction: This routine automates attendance notification/calculation/storage.
2. Inputs:
    - Super users input data.

3. Processing: A PHP/JSP based sever script is employed to add attendance information into SQL databases. In case attendance is below a certain threshold, a memo is sent to the notifier.

4. Outputs: The required information is stored in a database and users can see their attendance records.

**e. Streaming Method:**

1. Introduction: This method automates online courseware/multimedia sharing.
2. Inputs:
    - Super users upload presentations, pictures, PDFs, video/audio pod casts.

3. Processing: Flash or related streaming technology is employed to stream multimedia content via HOADIS. Newer additions sent to notifier.

4. Outputs: Set of hypertext links appearing for downloading courseware etc.

**f. Forums:**

1. Introduction: This module gives platform for students to join/make new groups like Linux User Group, ACM Student Chapter etc.
2. Inputs:
    - A user becomes moderator of a group. Users post thread on the forum.

3. Processing: User threads are added along with other data entries.

4. Outputs: Information is added to the forum.

**f. Blog:**

1. Introduction: This module gives both students and teachers a platform to express themselves.

2. Inputs:
    - A group of users post blogs about current events etc.

3. Processing: Information added to database. A new page is created.

4. Outputs: Blog is updated or made.

**g. WebSpace:**

1. Introduction: This module gives both students and teachers their own web space to host their websites/blogs.

2. Inputs:
    - Users upload their HTML based websites
3. Processing: Information added to database. A new page is created.

4. Outputs: Users got their webspace.

### 3.2 Hardware Limitations

HOADIS runs smoothly, irrespective of hardware. A minimum neccesity is required though it arise due to PHP/JSP servers, Apache server etc. It's recommended to use a Linux OS with a minimum of 64 MB RAM.

### 3.3 Attributes

The attribute places specific requirement of the software.

- **Availability:** Except of power cuts, the system will work fine regardless of platforms.
- **Security:** HOADIS is to be made secure with password encryption using Blowfish algorithm of PHP based PASSWORD() API and robust programming to defend SQL injection and other hacking techniques.
- **Maintainability:** HOADIS is modular based information system, so its easy for the administrator to maintain the software integrity.
- **Transferability/Conversion:** HOADIS is a portable web application; transferability can be done if bare minimum necessities of database and web servers are met.

### 3.4 Interfaces

- **User interface:** HOADIS will have easy to use GUI based interface.
- **Hardware interface:** An ethernet connection will be required by the server to connect to its peers.
- **Software Interfaces:** A Web Browser preferably Firefox should be used at end-user side.
- **Communication Interfaces:** HOADIS uses simple HTTP protocol over ethernet.
- **Data Base:** A lot of databases for storing information is used.

# 3. Design

After exploring the requirements of the project we work on a robust design based on the MVC architecture. For that we made a Software Design Description (SDD) document

## 3.1 SDD: IEEE format

The Software Design Description follows next page where we describe and discuss the design of the project based on MVC architecture.

# HOADIS

## Software Design Descriptions


**Developed By**

Ajay Chhatwal (07020002)
Rohit Yadav (07020003)

**Project Guide**

Dr. A. K. Tripathi
Professor
Department of Computer Science & Engineering
Institute Of Technology
Banaras Hindu University

# Table of Contents

# 1. Introduction

## 1.1 Purpose

To develop a web enabled Information System for educational institution using hypermedia and employing an open architecture.
It is intended to be a hypermedia information system that allows students to find up-to-date information about courses, teachers, announcements, events etc.

## 1.2 Scope

- *Ease of Use*
  The system is intended for users with varying levels of computer training and experience. It will, therefore, have a user interface that provides a consistent look-and-feel as the user and is easy to use.
- *Multiple Functions*
  Within the main functions of HOADIS we envisioned multiple subordinate functions, some of which will be designed locally and some of which will be drawn from a variety of Web-based resources.
- *Extensible and Open Architecture*
  We desire an architecture which will allow us to add components as quickly as we think of them. Open architecture allows people to write modules without having to go through the whole information system.
- *Access over the network*
  The system will be accessible over the intranet to ensure ease of access to the users.
- *Multiple Access Levels*
  The different types of users (Students, Teachers, Alumni, Non-teaching Staff, Guest and Administrator/Super User) with different user privileges will be using the system. Therefore, the system will have different access levels for different users.
- *Security*
  Our desire to operate in the Web environment complicates security tasks. We use the traditional approach of having users access the application through an identification-authentication-authorization process once at the beginning of the session. Coupled with this is the need to keep track of the user access level in order to determine the modules which a user is allowed to access.
- *Platform Independence*

Motivated by the premise that heterogeneity of software applications and hardware systems is here to stay we design the system without any restriction on the platform being used and it will use cross-platform technologies wherever possible.

### 1.3 Definitions and Acronyms

**HOADIS:** Hypermedia Open Architecture Departmental Information System.

**Information System:** An information System is a system of persons, hardware, software, communication devices, network and data resources that processes (can be storing, retrieving, transforming information) data and information for a specific purpose and activities that process the data and information in an organization, and it includes the organization's manual and automated processes.

**Hypermedia:** An extension to hypertext providing multimedia facilities, such as sound and video which allows the information to be processed in a non-linear manner.

**Open Architecture:** Open architecture is a type of software architecture that allows adding, upgrading and swapping components.

**Superusers:** Users with high access priviledges; they have power to read and write into databases and do maintanence and upgradation.

**HTML:** Hypertext Markup Language

**PHP:** Preprocessed Hypertext P

**JSP:** Java Server Pages

## 2 References

[1] A. Johnson, F. Fotouhi, N. Goel, R. Weinand, and J. Lechvar, “CASPER: A Hypermedia Departmental Information System” *IEEE Trans. Education*, vol.39, no. 4, pp. 471–477, Nov. 1996.

[2] J. Conklin, “Hypertext: An introduction and survey” *IEEE Trans. Comput*., vol. 20, no. 9, pp. 17–41, Sept 1987.

[2] J. Nielsen, “Hypertext and Hypermedia”, San Diego: Academic, 1990.

[3] I. Tom& S. Khan, T. Muldner, M. Nassar, G. Novak, and P. Proszynski, "Hypermedia-Introduction and survey," *J. Microcomput. Applicat*., vol. 14, pp, 63-103.

[4] J. J. Cimino, S. Socratous , "Just tell me what you want!: the promise and perils of rapid prototyping with the World Wide Web." , *J Am Med Inform Assoc. 1996*;3:719-23 (supl).

[5] J. J. Cimino,1 S. Sengupta, 1 Paul D. Clayton, V. L. Patel, A. Kushniruk, X. Huang, "Architecture for a Web-Based Clinical Information System that Keeps the Design Open and the Access Closed".

[6] Edmond Mesrobian, Richard Muntz, Eddie Shek, Silvia Nittel, Mark LaRouche, and Marc Kriguer, "OASIS: An Open Architecture Scientific Informatioin System". *IEEE Trans. Education*, July 1996.

[7] A. Olerup, "Design Approaches: A Comparative Study of Information System Design and Architecture Design".

[8] B. Langefors, "Some Approaches to the Theory of Information Systems." *BIT,* 3(1963).

[9] B. Langefors, "Theoreotical Analysis of Information Systems." 4th Edn(1st Edn 1966). *Studentlitteratur*, Lund (1973).

[10] B. Langefors, "Management information system design." *IAG Journal* (Amsterdam), 2 (4) (1969).

[11] B. Langefors, "System for Foretagsstyrning[Systems for management control]", 2nd edn. *Studentlitteratur,* Lund (1970).

[12] IEEE SDD Format.

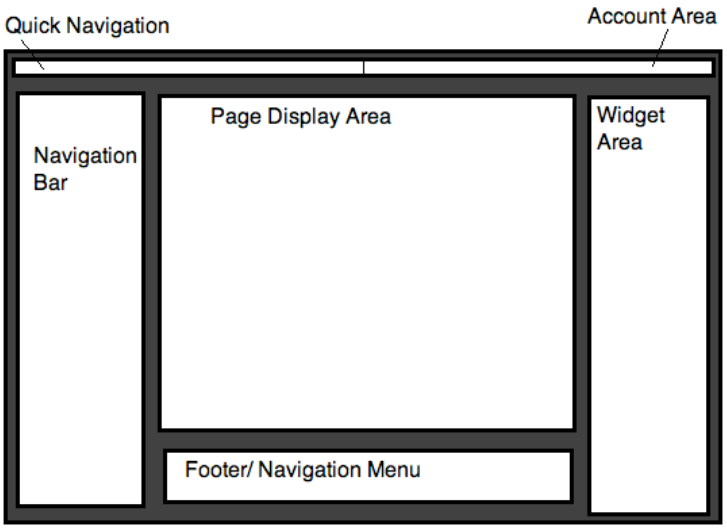
[13] www.wikipedia.org – Online Encyclopaedia.

[14] An Integrated Approach to Software Engineering by Pankaj Jalote.

# 3. Structure of Software

In purpose to satisfy the requirements stated in the SRS the structure of HOADIS is as presented below:

## 3.1 Layout Design




This is basic layout structure it shows the layout of the proposed web application.

## 3.2 Software designing approach to be followed

By the comparative study of all approaches of software design(analysing their advantages and disadvantages ) we found that Incremental approach will most beneficial for development of the web application: HOADIS.

The project is broken into about 15 modules as stated in the SRS. This allows independent development and faster production.

Basic principles of incremental development are

- A series of mini-Waterfalls are performed, where all phases of the Waterfall development model are completed for a small part of the systems, before proceeding to the next incremental, or
- Overall requirements are defined before proceeding to evolutionary, mini-Waterfall development of individual increments of the system, or
- The initial software concept, requirements analysis, and design of architecture and system core are defined using the Waterfall approach, followed by iterative Prototyping, which culminates in installation of the final prototype (i.e., working system).

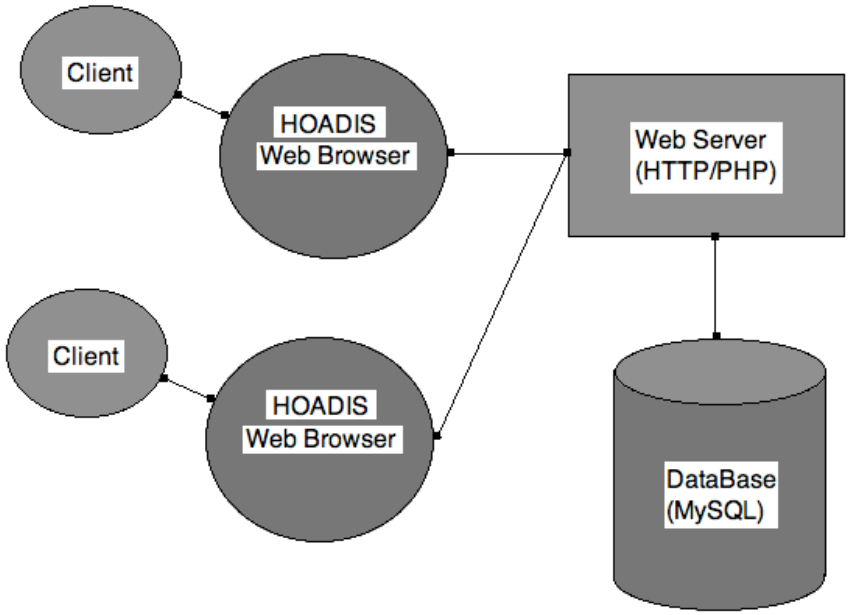
**Advantages**

- Steady development of Web Application.
- Planned approach to develop HOADIS.
- Involvement of user in development of HOADIS is a very effective approach as they can help in finding bugs and giving useful suggestions as the information system is being developed for these users.
- Modular approach will help in group development as the information system is being developed in group and will be favorable for bug detection.
- Since Waterfall approach is best suited for small software, it is ideal to use in the modules as they are literally small software.

**Disadvantage**

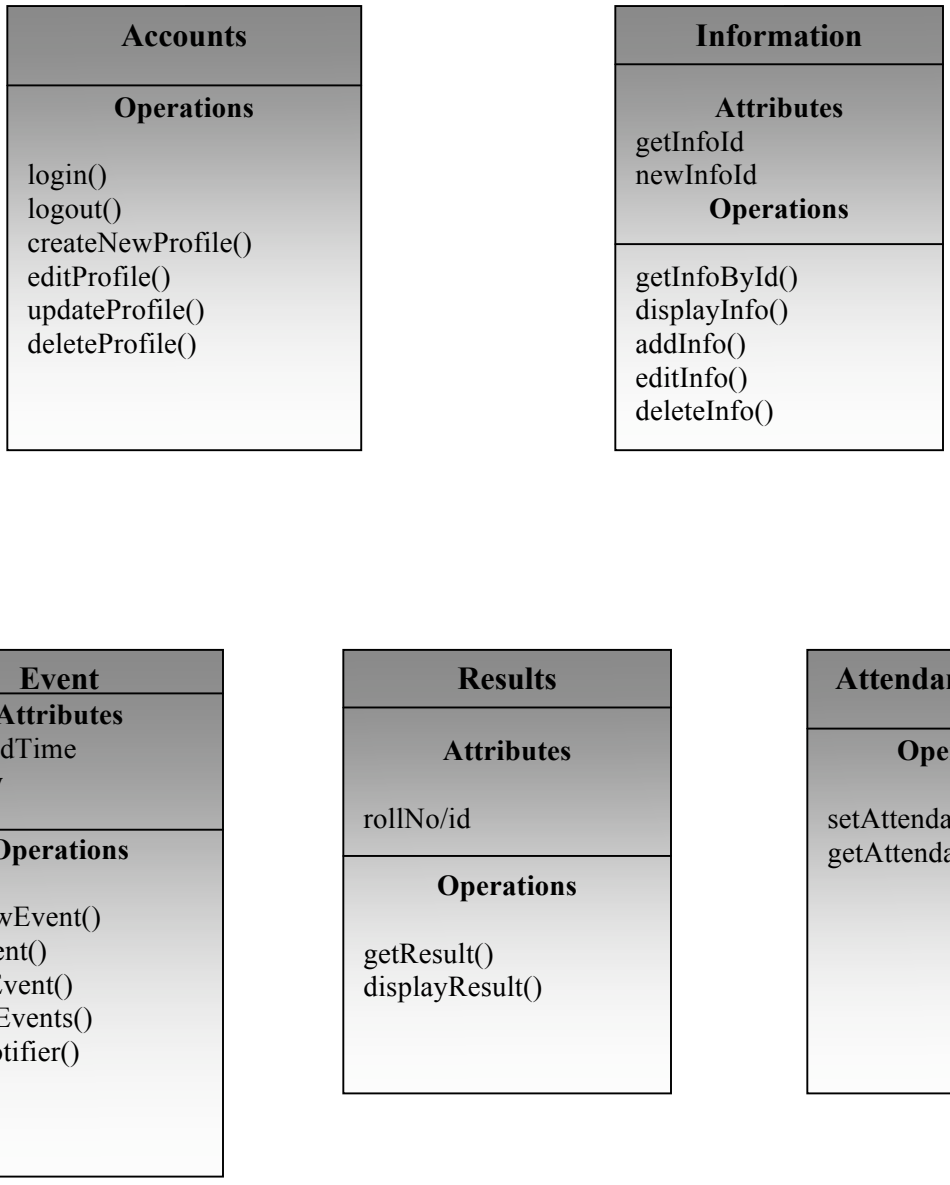
- The modules can be difficult to integrate if a common programming protocol is not adopted.

## 3.3 Data Flow Diagram




- Administrator is responsible for programming and maintanence of the information system.
- A web browser provides front-end to a user which connects to the HOADIS Web Server like HTTP and PHP.
- HOADIS Web Server fetches data from a Database Server like MySQL.
- The data is processed by the Web Server and sent to the Web Browser.
- Users find the information they intended..

## 3.4 Class Diagrams




## 3.5 Other Details

The coding of the front end and back end is done in HTML/CSS, PHP, JavaScript, SQL and AJAX.
The whole software is proposed made in modules that makes it pluggable, removable and easily upgradable.
The approach would be Object Oriented Programming.

## 4. Design description information content

| S.No. | Design Entity Name | Type | Function or purpose | Interface |
|---|---|---|---|---|
| 1 | Theme | module | Contains information Front End Layout | Web Browser |
| 2 | User Authentication Service | module | Account Management | Web Server |
| 3 | Information Manager (Servlets) | module | Fetches and displays general information. | Web Server |
| 4 | User Profile | module | Allows each type of user to edit and view his/her profile. | Web Server |
| 5 | Notifier | module | Notifies New events | Web Server |
| 6 | Event Manager | module | Updates/Adds/Deletes new events and updates notifier | Web Server |
| 7 | Online Notice Board | module | Notice Board | Web Server |
| 8 | Time table | module | Schedules time table, | Web Server |
| 9 | Attendance | module | Manages records for attendance | Web Server |
| 11 | Result Manager | module | Manages Result declaration | Web Server |
| 12 | Online Courseware | module | Manages online courseware etc. | Web Server |
| 13 | Online Assignment Submission | module | Online submission of electronics Lab/Class assignment like source files etc. | Web Server |
| 14 | Gallery and Microblogging | module | Platform to share photos, videos etc. Share your ideas through microblogging. | Web Server |

# 4. Testing

Testing is an important part of software development. After completing the project, we ran the project to test some of its functionality and integrity.

## 4.1 User Authentication Test

This is to test login, logout and register modules.

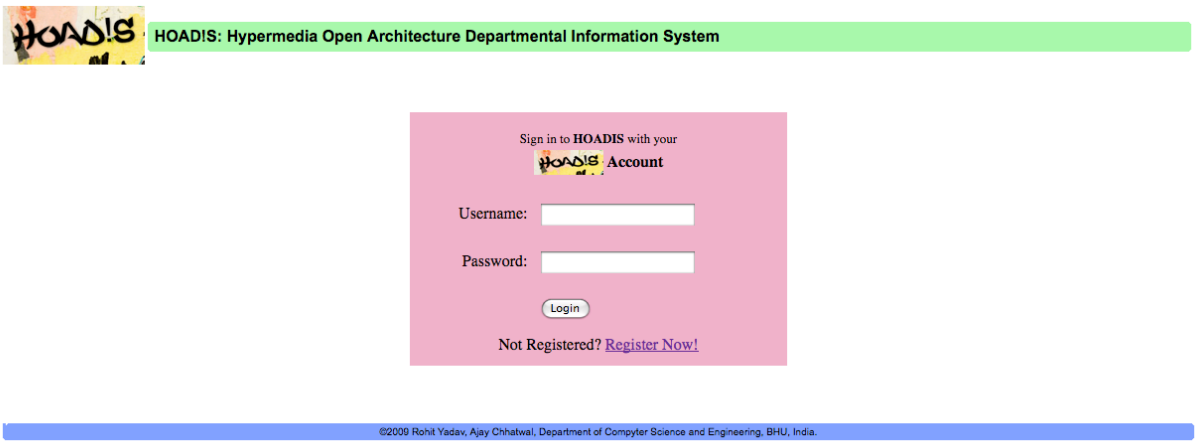



**Fig 4.1.1 Login Page**

We clicked on “Register Now” to create a new account and used it to login HOADIS.

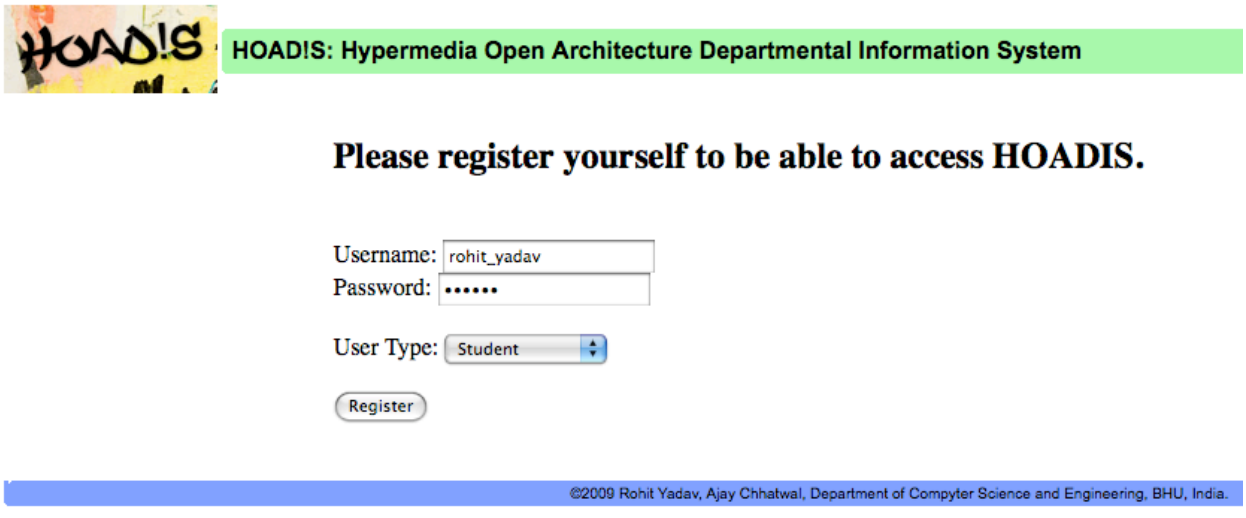



**Fig 4.1.2 Registration Page**

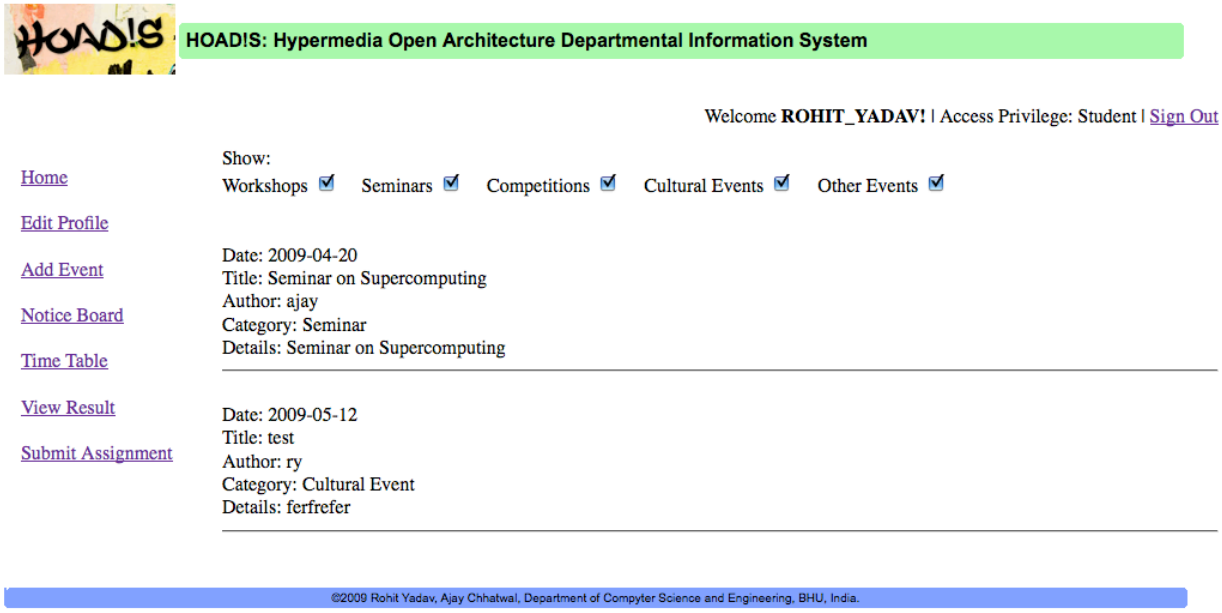



**Fig 4.1.3 Home Page**

## 4.2 Profile Test

We checked the profile module, which keeps user data.

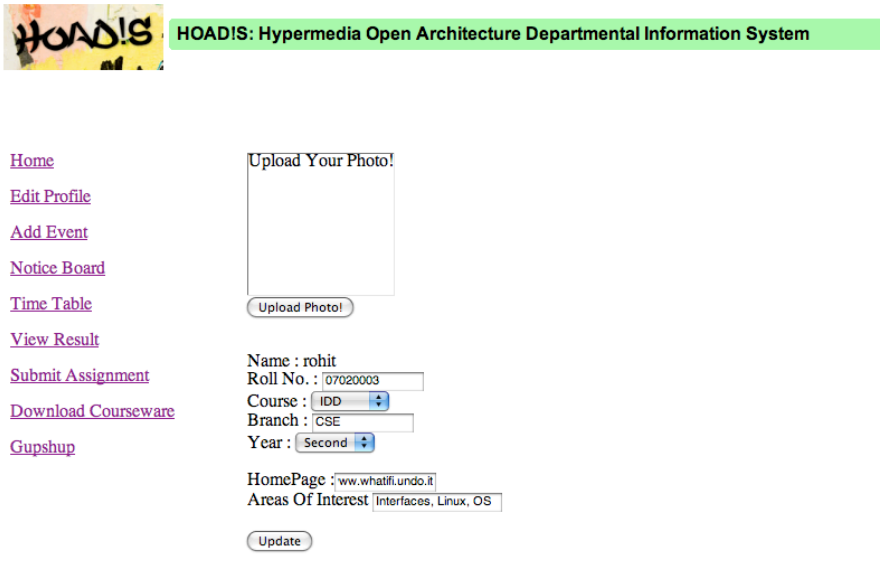



**Fig 4.2.1 Profile Page**

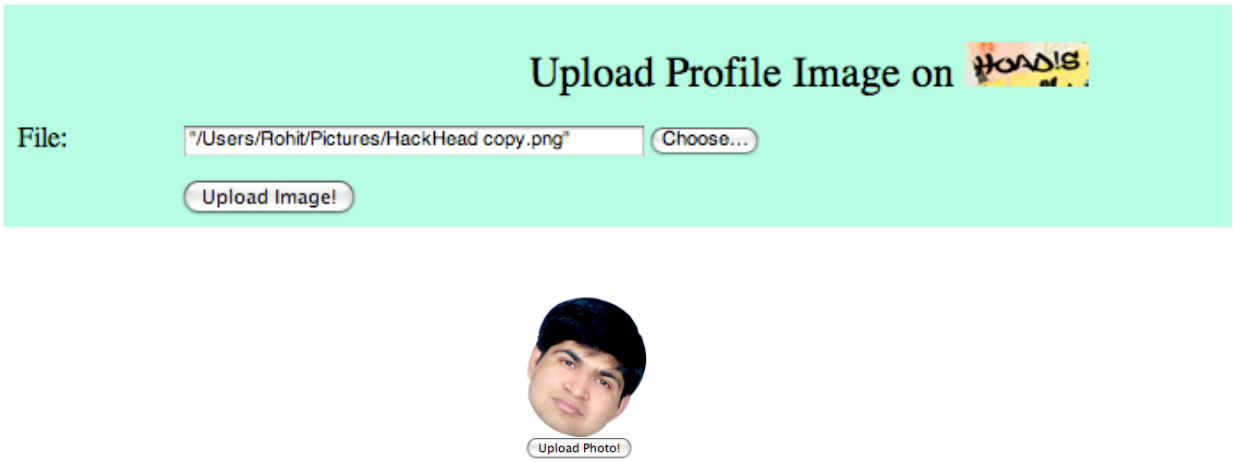



**Fig 4.2.2 Upload Profile Image Page**

## 4.3 Event Test

We tested the event module by adding a fake event to the information system. Then we checked and found that it shows up on the notice board and on notification widget.

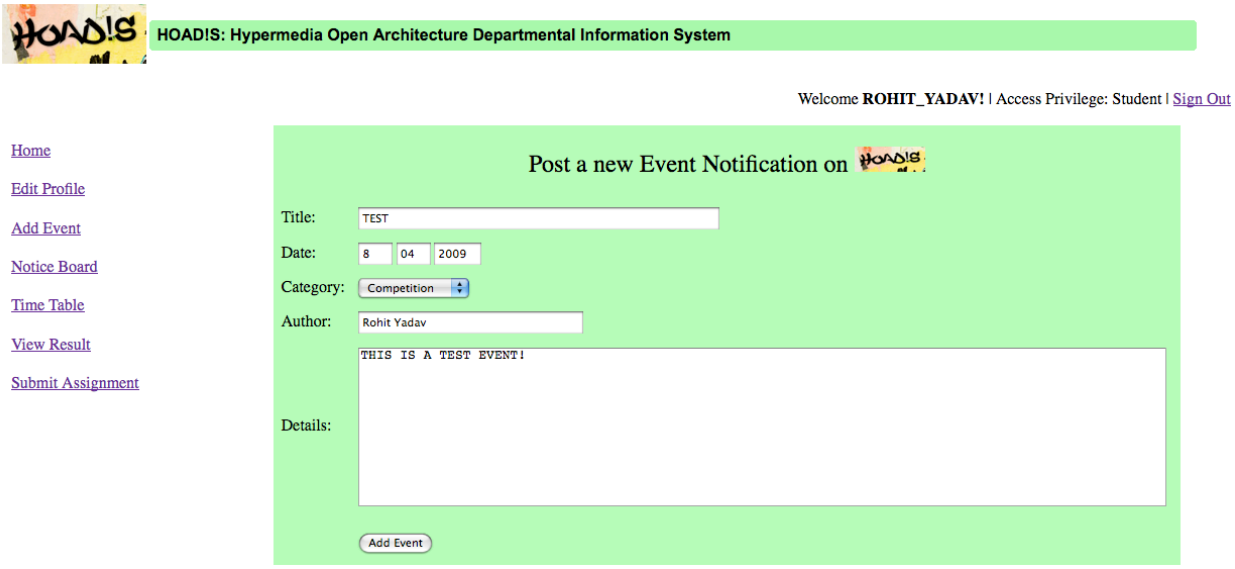



**Fig 4.3.1 Add Event Page**

## 4.4 Notice board test

We checked that dynamic notice board powered by JavaScript.

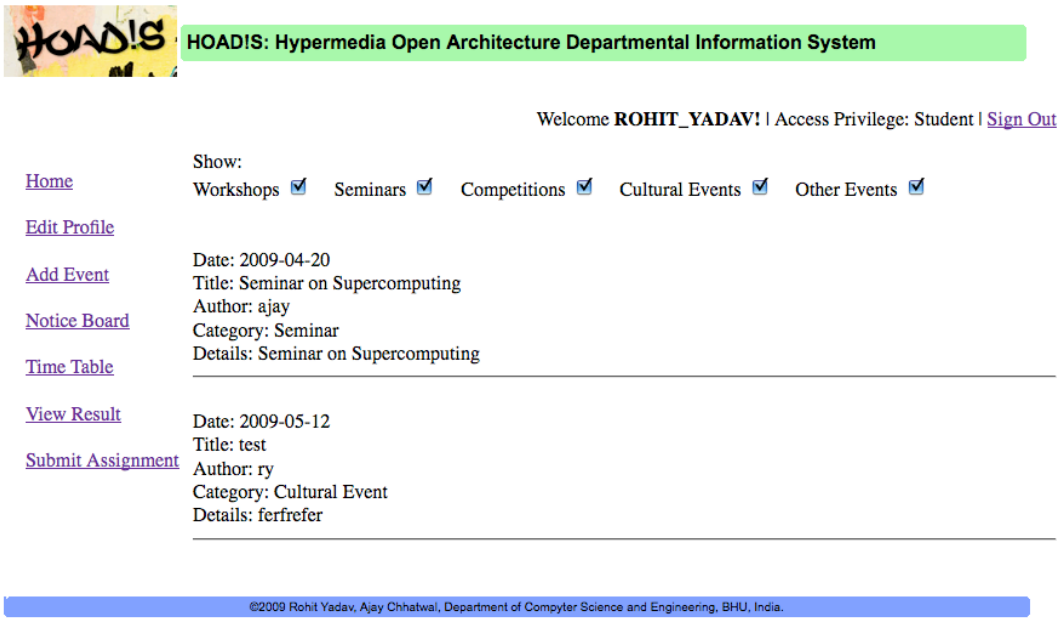



**Fig 4.4.1 Add Event Page**

## 4.5 Timetable Test

The timetable module worked fine and was read only for students and writable for faculty and staff.

## 4.6 Result Module Test

The Result declaration module worked fine and was read only for students and writable for faculty and staff.

## 4.7 Online Assignment Submission Test

The Online Assignment Submission module worked and files were uploaded in the assignment folder on the server machine.

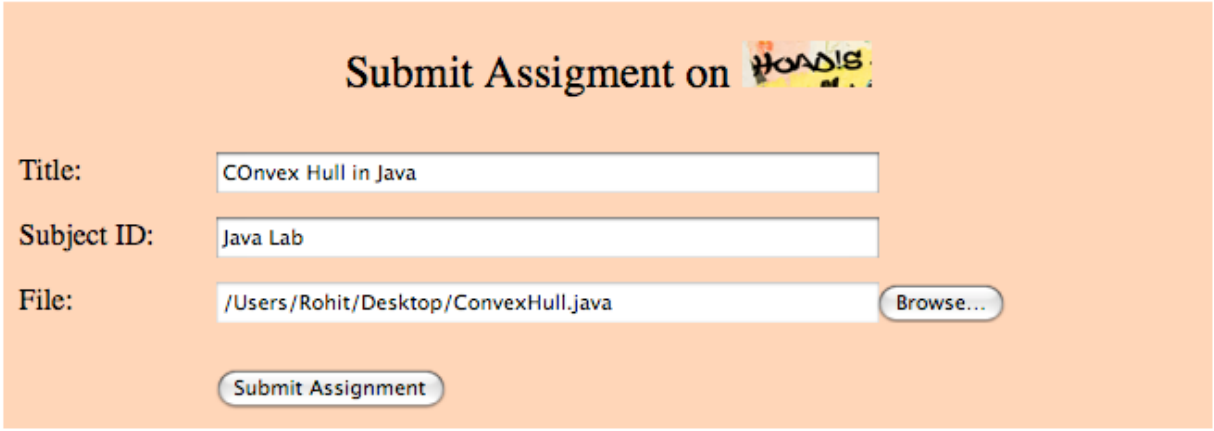



**Fig 4.6.1 Submit Assignment Page**

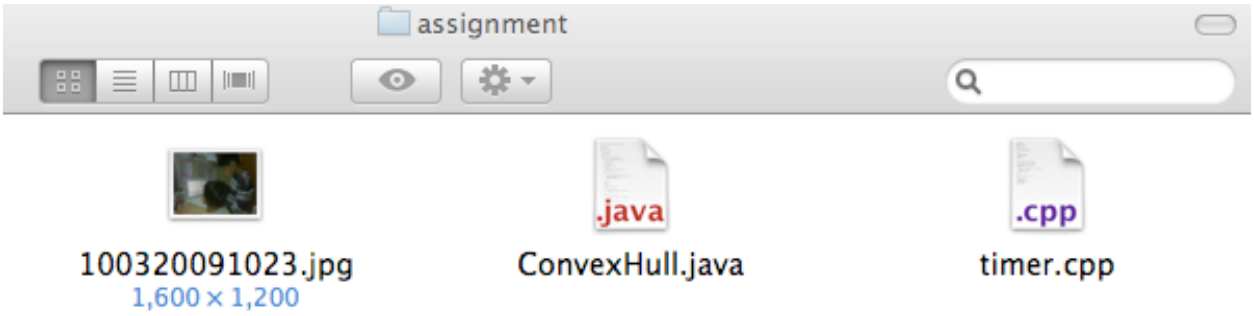



**Fig 4.6.2 Files on server in *assignment* directory**

## 4.8 Courseware Sync. Test

The courseware uploading ran successfully and files were found in courseware directory on the server.


**Fig 4.8.1 Courseware Upload Page**

## 4.9 Departmental MicroBlog(GupShup) and Gallery Test

Both the modules ran successfully and users were able to share their ideas and explore gallery.

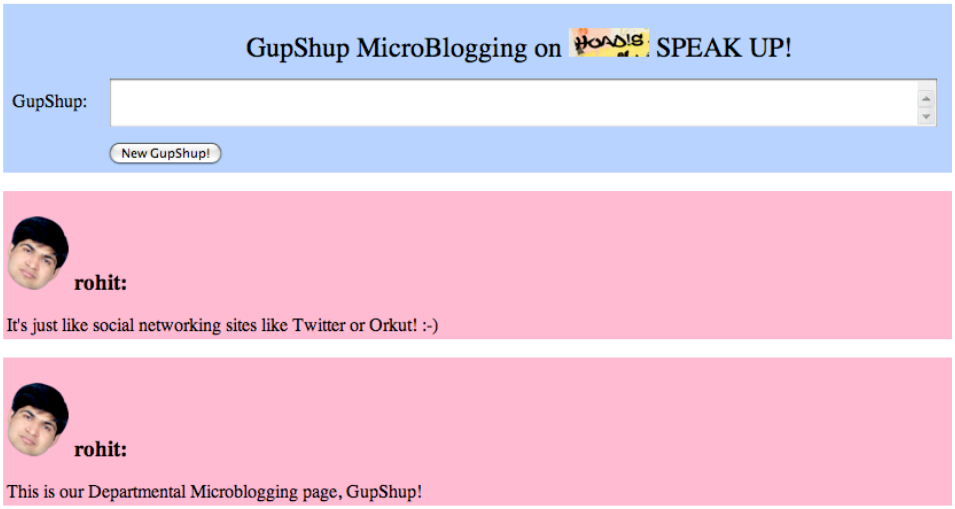



**Fig 4.9.1 GupShup: Departmental MicroBlogging Page.**

## 4.10 Administrative Maintenance test

In this test we verified if the admin was able to maintain and perform operations on the departmental information system like adding courses, updating theme, cleaning server directory, etc.

## 4.11 Role Based Access Test

We tested by logging into different account types: admin, faculty, staff and student. We found that role based access and privileges were working fine.

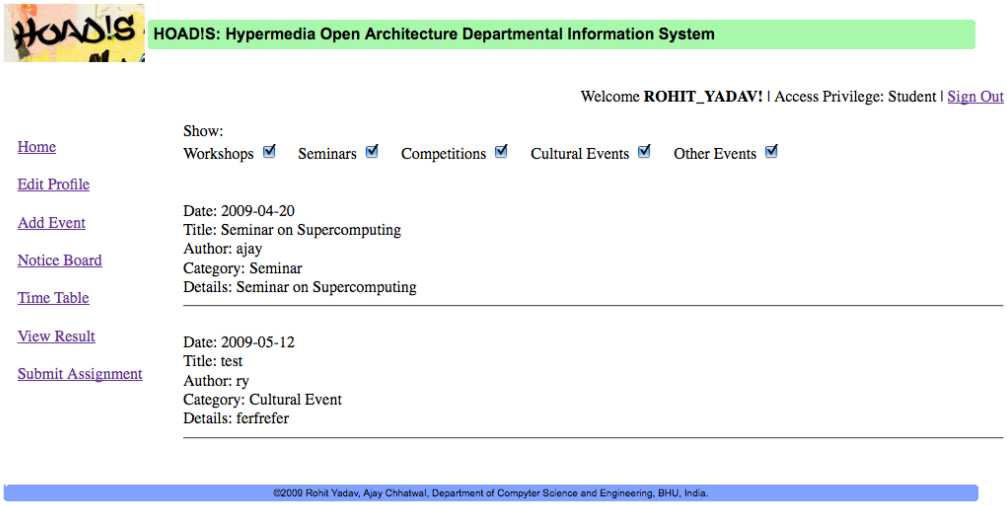



**Fig 4.11.1 User logged in as Student.**

## 4.12 Error Handling Test

We tested the error-handling page that appears if the requested services or module malfunctions or is not available.

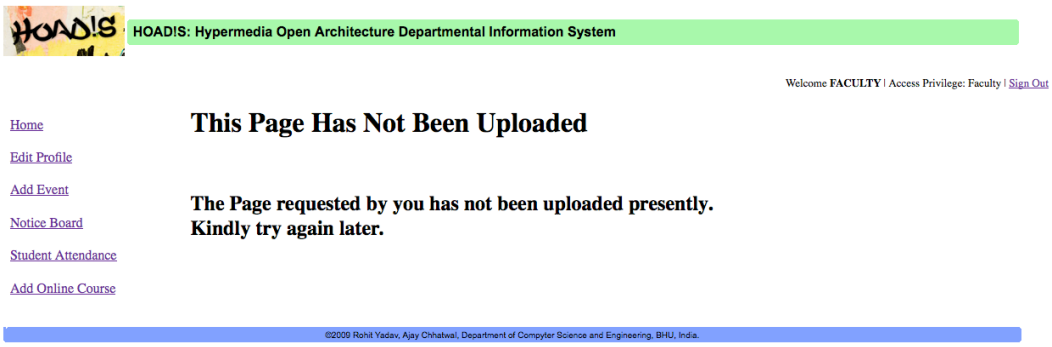



**Fig 4.12.1 Error Handling Page.**

# 5. Conclusions and Further Work

The project is complete but finds further work as the field of web application is developing very fast and arising new technologies are ubiquitous. Many more modules can be integrated and improved like integration of email and other web 3.0 services.

The Internet has changed technology and communications. It has shown how standards work and how they can benefit even competing products. Different machines and different systems can work together based on common standards. The power of the standard will streamline products and services as well as operations.

Finally, we can conclude that though the project finds potential work to be done, a robust framework has been made and using this framework an enhanced version of HOADIS can be worked out in future.

# 6. References

[1] A. Johnson, F. Fotouhi, N. Goel, R. Weinand, and J. Lechvar, "CASPER: A Hypermedia Departmental Information System" *IEEE Trans. Education*, vol.39, no. 4, pp. 471–477, Nov. 1996.

[2] J. Conklin, "Hypertext: An introduction and survey" *IEEE Trans. Comput*., vol. 20, no. 9, pp. 17–41, Sept 1987.

[3] J. Nielsen, "Hypertext and Hypermedia", San Diego: Academic, 1990.

[4] I. Tom& S. Khan, T. Muldner, M. Nassar, G. Novak, and P. Proszynski, "Hypermedia-Introduction and survey," *J. Microcomput. Applicat*., vol. 14, pp, 63-103.

[5] J. J. Cimino, S. Socratous , "Just tell me what you want!: the promise and perils of rapid prototyping with the World Wide Web." , *J Am Med Inform Assoc. 1996*;3:719-23 (supl).

[6] J. J. Cimino,1 S. Sengupta, 1 Paul D. Clayton, V. L. Patel, A. Kushniruk, X. Huang, "Architecture for a Web-Based Clinical Information System that Keeps the Design Open and the Access Closed".

[7] Edmond Mesrobian, Richard Muntz, Eddie Shek, Silvia Nittel, Mark LaRouche, and Marc Kriguer, "OASIS: An Open Architecture Scientific Informatioin System". *IEEE Trans. Education*, July 1996.

[8] A. Olerup, "Design Approaches: A Comparative Study of Information System Design and Architecture Design".

[9] B. Langefors, "Some Approaches to the Theory of Information Systems." *BIT,* 3(1963).

[10] B. Langefors, "Theoreotical Analysis of Information Systems." 4th Edn(1st Edn 1966). *Studentlitteratur*, Lund (1973).

[11] B. Langefors, “Management information system design.” *IAG Journal* (Amsterdam), 2 (4) (1969).

[12] B. Langefors, “System for Foretagsstyrning[Systems for management control]”, 2nd edn. *Studentlitteratur,* Lund (1970).

[13] IEEE SRS Format.

[14] IEEE SDD Format.

[15] An Integrated Approach to Software Engineering by Pankaj Jalote.

[16] wikipedia.org for general information.

[17] tizag.com for JSP/MySQL.

[18] Beginning JavaScript, Third Edition by Paul Wilton and Jeremy McPeak.

[19] Beginning JavaScript, Third Edition by Paul Wilton and Jeremy McPeak.

[20] JavaServer Pages™, 2nd Edition by Hans Bergsten.