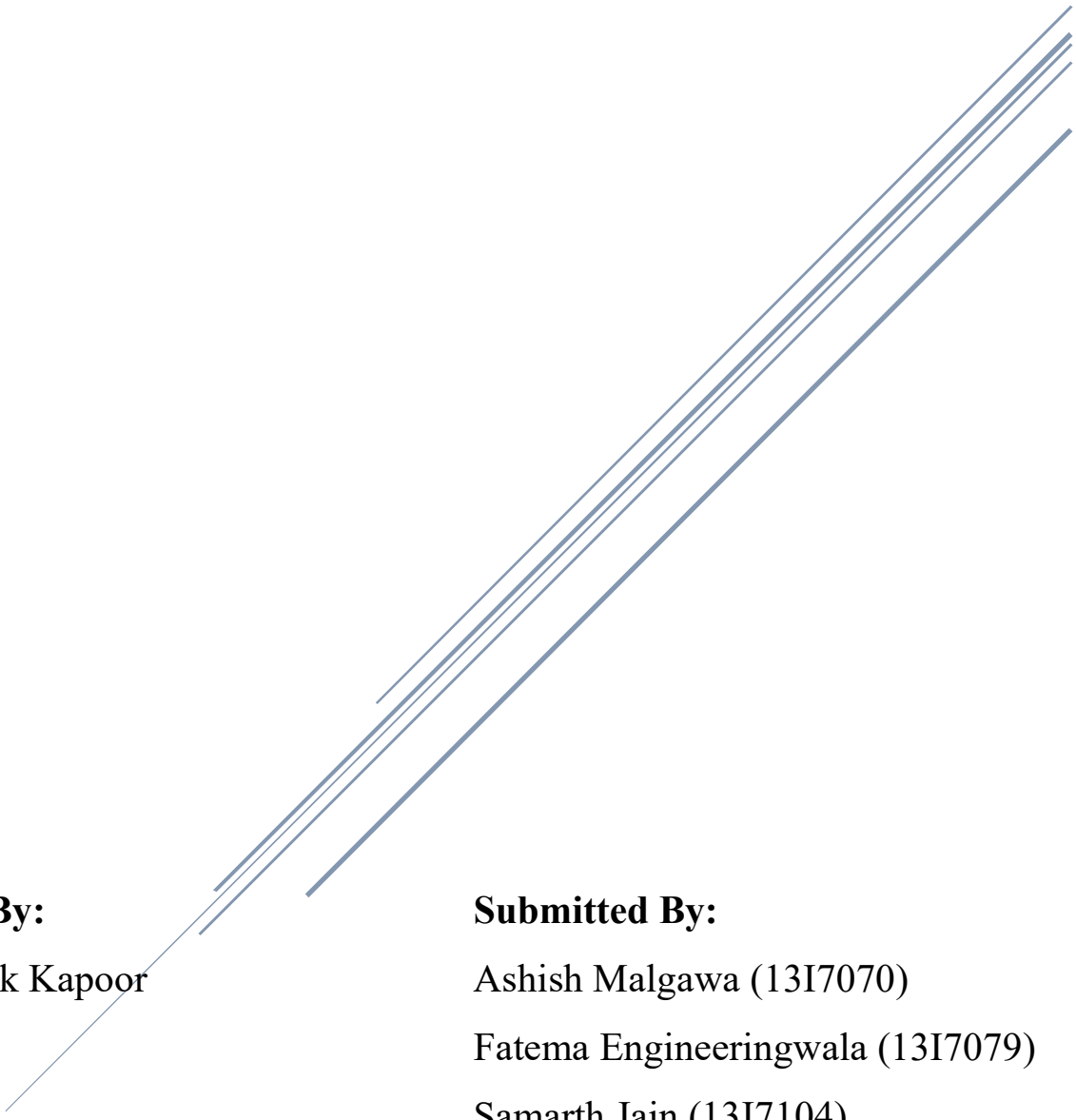
LEARNEST

Software Requirement Specification (SRS)



Guided By:

Dr. Vivek Kapoor

Submitted By:

Ashish Malgawa (13I7070)

Fatema Engineeringwala (13I7079)

Samarth Jain (13I7104)

Department of Computer Engineering / Information Technology

Institute of Engineering & Technology

Devi Ahilya Vishwavidyalaya, Indore (M.P.)

(www.iet.dauniv.ac.in)

April 2016

Table of Contents

1. Introduction

The purpose of this section is to provide the reader with general, background information about the website "LEARNEST" - an education portal. The Main objective of the Project is to provide study material and lectures to the students who are studying in the Educational organization (DAVV). We are trying to develop software that will provide instant data delivery. This can be achieved through World Wide Web.

1.1 Purpose

The document is the Software Requirement Specification for Education Portal.
The information described in this document is the features and requirements of the project To be developed for Online Education and is named as "LEARNEST- Education Portal" Herein after referred as "Portal". The objective of this "Portal" is to "Launch an interactive, Rich and informative portal focused on education by developing, implementing and Enriching a learning platform and content".
This is a database project which aims at creating a Courses portal for a campus or University. This allows registered users of the system to access the materials (Video lectures and related documents) published for their course. In our project development we

Considered DAVV as a typical educational institution which maintains student database. It will also provide data security as we are using the Secured data bases for maintaining the documents (database).We can also conserve the time and human resources for doing the same task. The data can be maintained for longer
Period with no loss of data.

The main objective of this document is to illustrate the requirements of the design and Development of the "Portal".
The document is developed after consultation with the UNIVERSITY and considering the complete requirement specifications of the given Project. This Software Requirements Specification provides a complete description of all the functions and specifications of proposed "Portal".

1.2 Document Conventions

Following conventions were used while creating UML Use Case Diagram

1.2.1 Use Cases

A use case describes a sequence of actions that provide a measurable value to an actor. A use case is drawn as a horizontal ellipse on a UML use case diagram

1. Use Case Names Begin With a Strong Verb
2. Name Use Cases Using Domain Terminology
3. Place Your Primary Use Cases In The Top-Left Corner Of The Diagram
4. Imply Timing Considerations By Stacking Use Cases.

1.2.2 Actors

An actor is a person, organization, or external system that plays a role in one or more interactions with your system (actors are typically drawn as stick figures on UML Use Case diagrams).

1.2.3 Relationships

There are several types of relationships that may appear on a use case diagram:

- An association between an actor and a use case
- An association between two use cases
- A generalization between two actors
- A generalization between two use cases

Associations are depicted as lines connecting two modeling elements with an optional open-headed arrowhead on one end of the line indicating the direction of the initial invocation of the relationship. Generalizations are depicted as a close-headed arrow with the arrow pointing towards the more general modeling element.

1.3 Intended Audience and Reading Suggestions

This document is meant for users, developers, testers, and documentation writers. The SRS document aims to explain in an easy manner, the basic Idea behind the Online Education Portal and how the developers aim to achieve their goals. It also aims to introduce to the users the main features of the Online Education Portal.

1.4 Product Scope

This software system will be an online education portal for any university wishing to manage their academic needs online. More specifically to design and develop a simple and intuitive system which shall cater the academic needs of any institute. The system shall provide features to the user of an educational institute to access documents uploaded by various faculties for particular courses, have discussion threads, instant notifications, assignment submissions, blogs, share old papers, etc.
This document is intended for all the users to understand the usage and maintenance of the portal who will know about the basic knowledge of computer. Database maintenance and steps to follow authentication to become a registered user for the portal. The user having the basic knowledge of internet can get the information (lectures, notes etc.) about a subject or topic. In our project we are giving the information globally. It is a powerful tool which allows the students to access the material provided by the faculties easily.

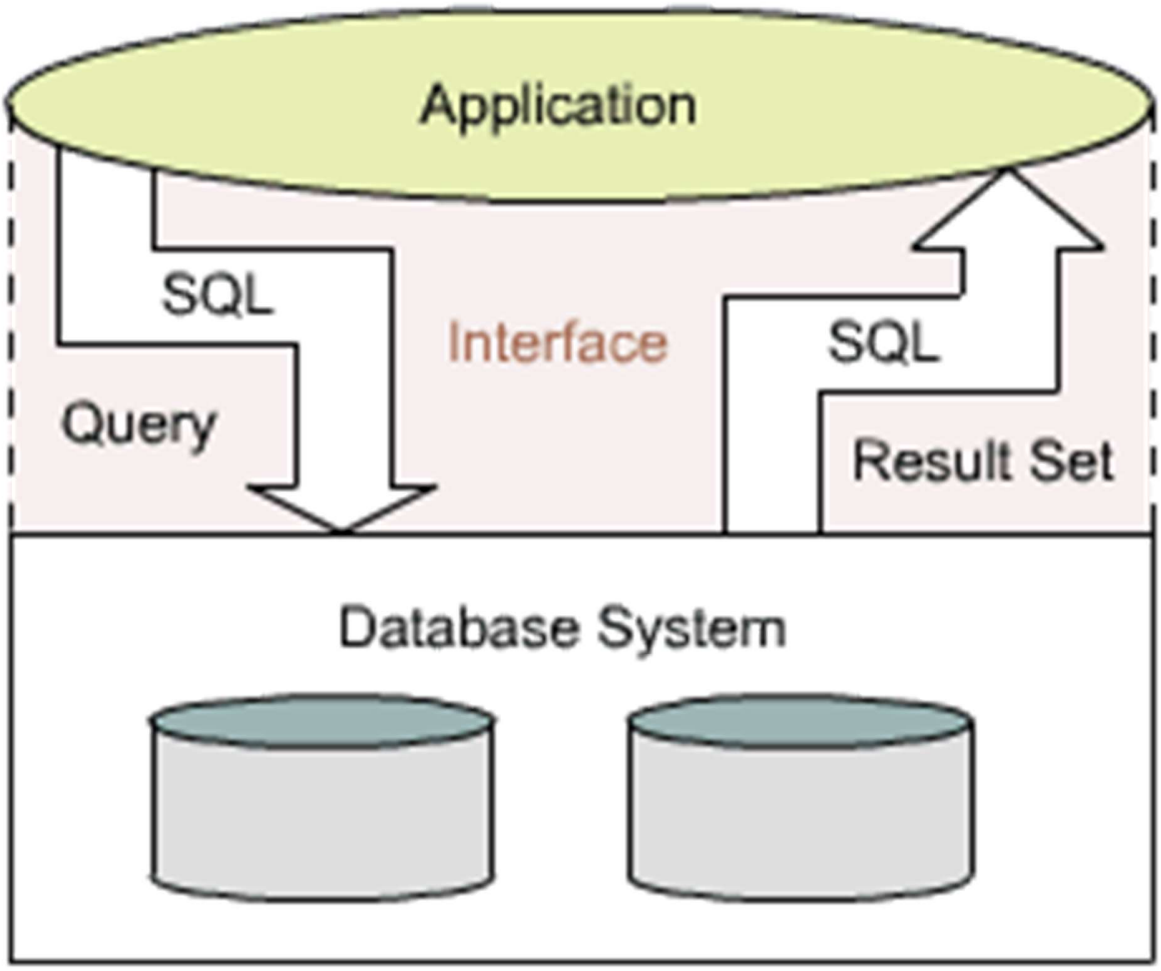
1.5 References

- http://agilemodeling.com/style/useCaseDiagram.htm
- https://en.wikipedia.org/wiki/Main_Page
- https://www.quora.com/

2. Overall Description

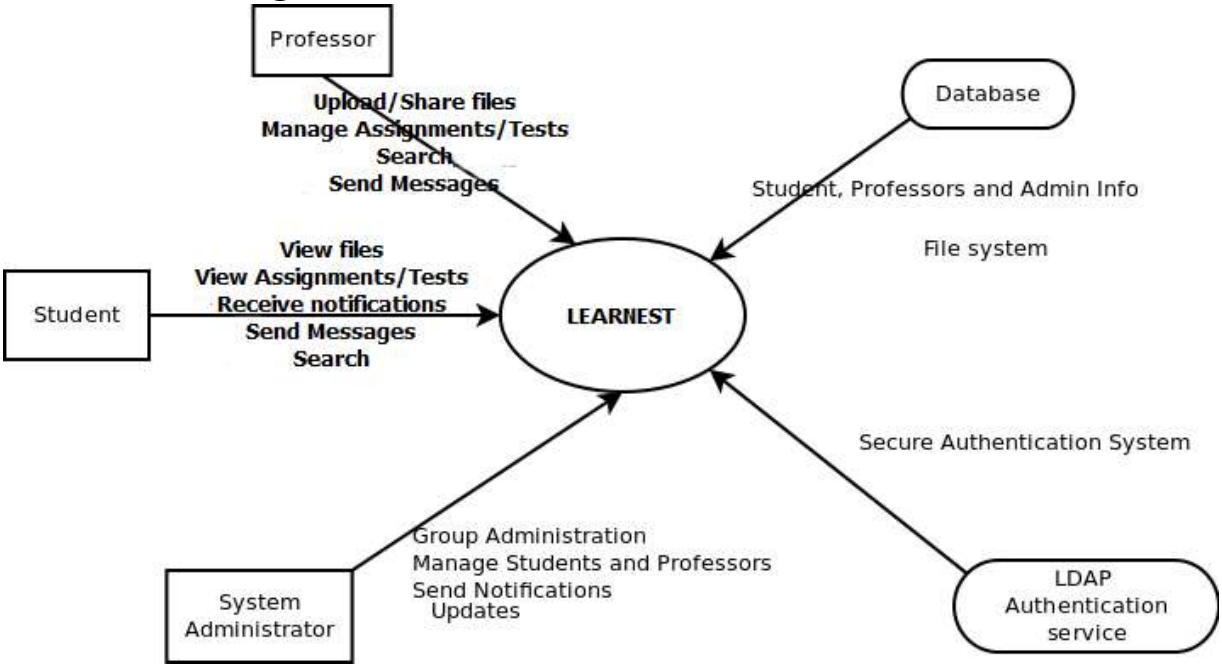
2.1 Product Perspective

Online Education Portal is a standalone system. It aims to create a new website by providing all features like to search course material and download it, watch videos related to various topics in a course which will be uploaded by the faculty, and an easy means to communicate with the faculty. Thus, our project is a self-contained database project. All the students will benefit from the services rendered by the "Portal" shall achieve their learning goals. The proposed solution is built from scratch by customizing Open Source web technologies and frameworks such as Java, PHP, and MySQL etc.
This product is a web based application with a network of users to access course material uploaded by the university's faculties. It requires a web browser with an internet connection and a server supporting PHP and a MySQL database.



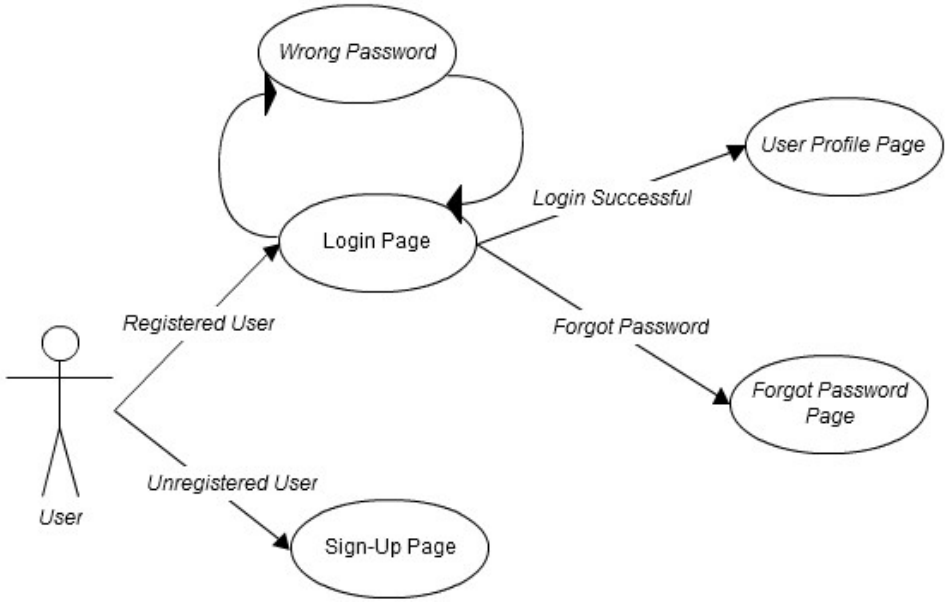
2.2 Product Functions

2.2.1 Context Diagram

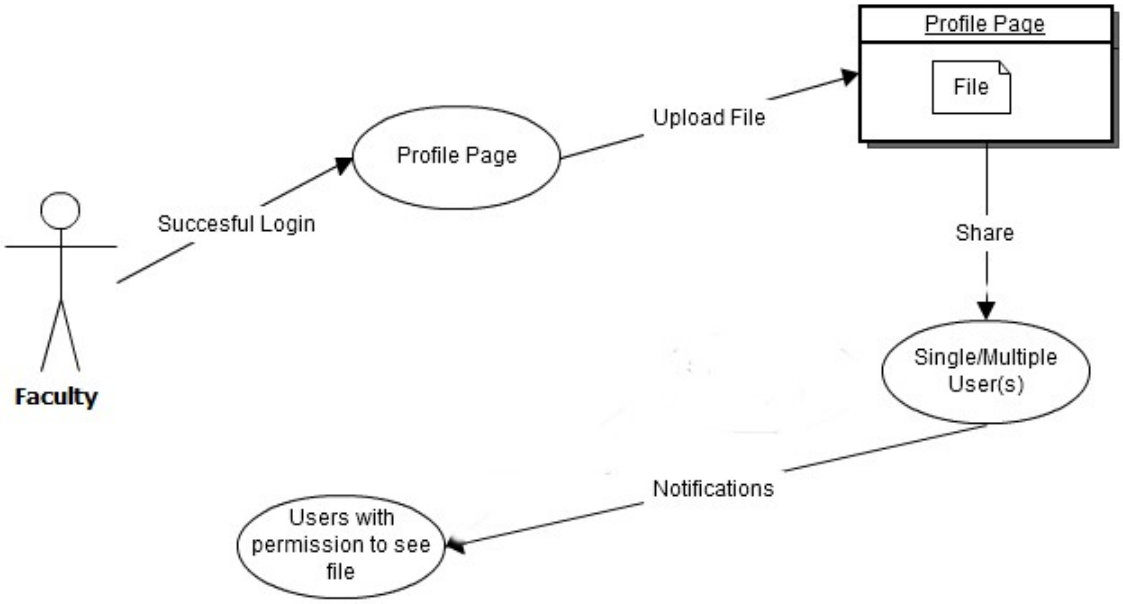


2.2.2 Use Case Diagrams

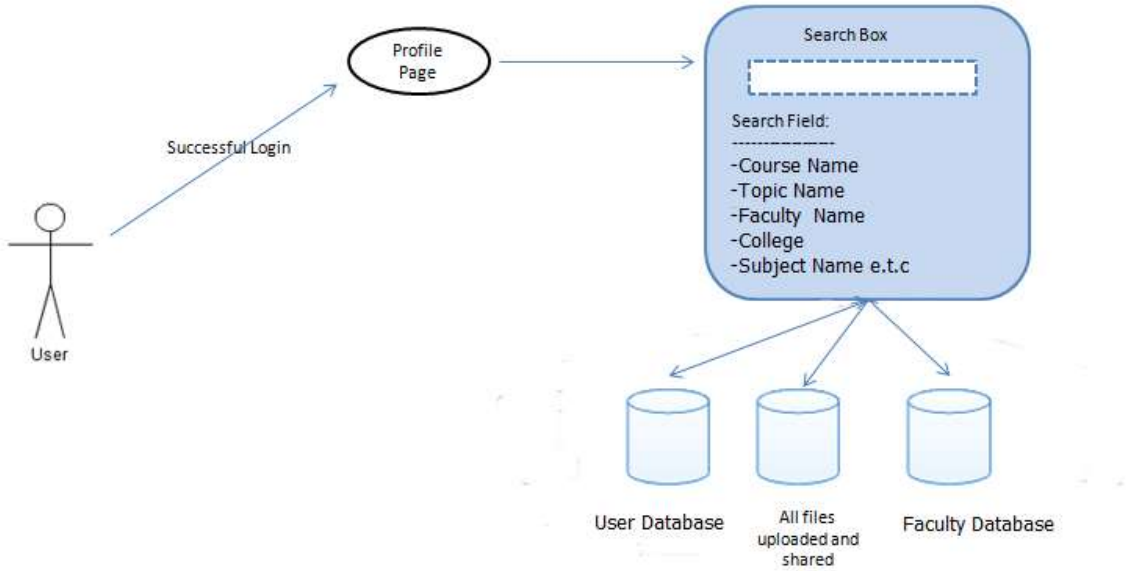
2.2.2.1 User Login



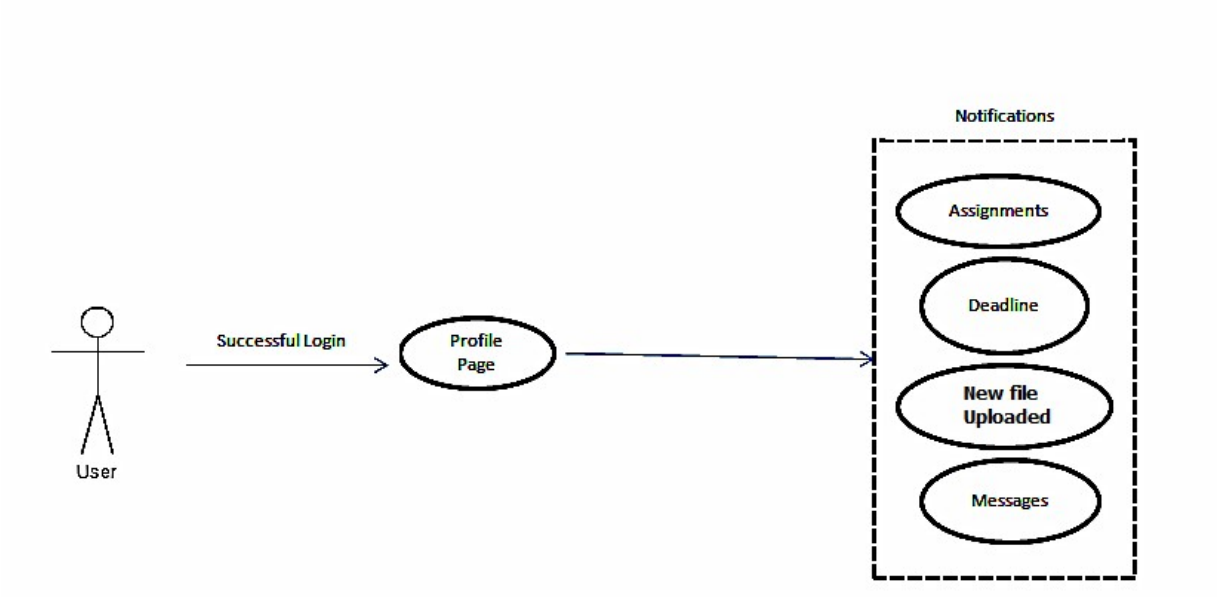
2.2.2.2 Content Sharing (File Upload)



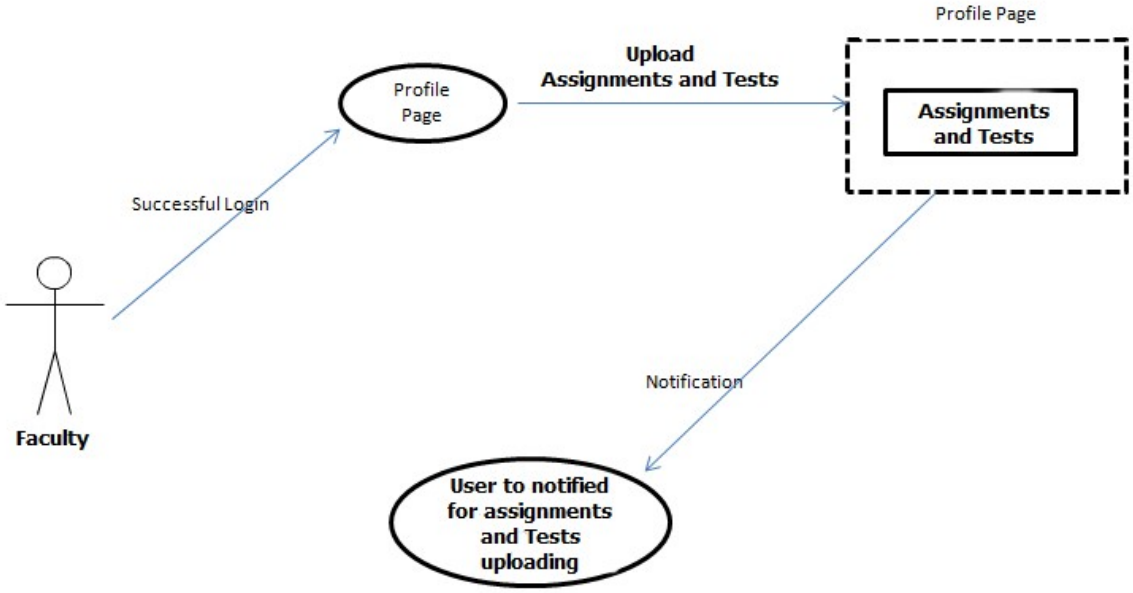
2.2.2.3 Search



2.2.2.4 Notifications



2.2.2.5 Assignments and Tests



2.2.3 Use case descriptions /Introductions

2.2.3.1 Groups

The system shall provide the administrator ability to create and manage new groups of different kinds like a course group or a department group. Here a user can be part of multiple groups.

Course Group

Students belonging to same course (e.g. Database Course) along with the professors and teaching assistant of the course will be part of the particular course group. Here the professors along with the teaching assistants (*conditional) will be the course administrators with the students being the users of the group.

Department Group

Students belonging to same department (e.g. Computer Science, Electricals) along with the professors of the department will be part of the particular department group. Here an assigned professor will be the course administrator with the students and the other professors being the users of the group.

2.2.3.2 Messaging

The system shall provide the functionality to send a personal message to any of the faculty(s).

2.2.3.3 Content Sharing

The System shall provide the ability to the faculty to share content on a group level i.e. one can share any data (books/pdfs) with the entire group.

Also they will be able to upload various types of files (PowerPoint presentation, Portable document format, AV Clips, Spread sheets, Documents, Programs etc.) for students to download/view.

Group Content Sharing

The faculty shall be provided with the functionality to share a particular piece of data with any subset of users belonging to the same group.

2.2.3.4 Tests

The system shall provide Test upload facility i.e. Tests can be uploaded in a fixed format by the faculty i,e MCQ. - The tests page would have an active timer to define the time limit for students.The Tests page shall also provide the administrator with the option to automatically close the Test link once past the deadline. A user can give a test only once.

2.2.3.5 Search

The system must be able to use search functionality as a way to navigate pages instead of using hierarchical links. Academic Portals can be difficult to navigate and requires too many clicks to be efficiently used. Too many steps to complete basic actions. A search utility enables students to find what they are looking for quickly in addition to having a hierarchical approach to finding functions of the academic portal.

1. A user must be able to search through pages in a course.

Users should be able to search for pages of the academic portal, since most students will be searching for material given in courses. The search feature should be present in the following way.

It should be present in the form of a search box and a "Search" button.. When a user clicks a search result link, they should be taken to the page corresponding to the search result.

2. The system must display a search box on every page after a user has logged in. Users should be able to search from any page.

2.2.3.6 FAQ

This utility enables students to have a look at the FAQ to get the common questions answered

2.2.3.7 Notifications

1. The system must provide notifications for new updates(such as assignments or announcements).

Whenever a file is uploaded by the teacher the students who have subscribed that faculty will get a notification that a new file is uploaded.

2.3 User Characteristics

2.3.1 Students

Students are the primary consumers of an academic portal. They are accessing information posted by professors.

2.3.2 Professors

Professors are the primary content administrators of an academic portal. They are uploading files, links, and multimedia

2.3.3 System Administrators

System administrators are primarily responsible for maintaining the academic portal. They contribute minimally to the courses themselves, but spend more time modifying the system's configuration and making appropriate updates.

2.4 Operating Environment

Software:-

1. Browser(IE9 and above)

2. Pdf reader(Adobe Reader)

3. Document Viewer(MS Office)

4. Media Player(VLC)

5. Adobe Flash Player

Hardware:-

1. P-4

2. RAM -256 MB

2.5 Constraints

2.5.1 User Interface Constraints

Using this system is fairly simple and intuitive. A user familiar with basic browser navigation skills should be able to understand all functionality provided by the system.

2.5.2 Hardware Constraints

The system should work on most home desktop and laptop computers which support JavaScript and HTML5.

2.5.3 Software Constraints

The system will be intended to run on Firefox 4 and above, Google Chrome 10 and above and Internet Explorer 8 and above.

2.5.4 Data Management Constraints

System shall be able to interface with other components according to their specifications.

2.5.5 Operational Constraints

The system is limited by its operating server in terms of the maximum number of users it can support at a given time.

2.5.6 Site Adaptation Constraints

The component will be adapted to the overarching system at the conclusion of the system creation.

2.5.7 Design Standards Compliance

The system shall be implemented in PHP.

2.6 Assumptions and dependencies

Most of the academic portals have a lot of redundant features which are rarely used in an academic sessions. Our new system focuses on the features which are most important to the users of an academic institute along with introduction of some new features which other portals lacks.

3. External Interface Requirements

3.1. User Interfaces

The "Portal" should work and be tested against IE 8.0 or later and Firefox3.6 or later.

3.2 Hardware Interfaces

There are no special hardware interface requirements for this "Portal"

3.3 Software Interfaces

3.3.1 Web Server

- Apache will be used as web server
- The user inputs data via the web server using HTML forms
- The web server executes the PHP as a module and PHP script retrieves the post data if available.
- The web server receives information back from the PHP script.
- The web server displays a HTML page as result to the end-user.

3.3.2 PHP Application

The actual program that will perform the operations is written in PHP. All data will be stored in a database.

3.3.3 MySQL Database

It's an open source SQL database to store all data which communicates with the application on the server.

3.3.4 Web Browser

Web browser should give support to angular JS.

3.4 Communications Interfaces

There are no special communication interface requirements

4. System Features

4.1User Login

- **Purpose:** User logs in to system using existing profile.
- **User:** A user with an existing profile.
- **Input Data:** Profile username and password.
- **Output Data:** Corresponding page data.
- **Invariants:** Profile table data and user information.
- **Pre-conditions:** User is not logged in to a profile, input profile exists in data base, user password matches profile.
- **Post-conditions:** User's computer has been supplied with appropriate cookie, page data is appropriate for selected profile.
- **Basic Flow:** Webpage looks up profile data and returns the matching cookie. Webpage is updated to match new user data.
- **Alternative Flow(s):** Invalid password, invalid username, or mismatched username and password redirect to error message and previous page. Business Rules: This allows users to log in to their profile from anywhere.

4.2 Content Sharing (Upload Files)

A faculty logs into the system and is on any page and wants to share some content on a group level.

- **Purpose:** A user wants to share some data (pdf, ppt etc).
- **User:** A legitimate faculty logged into the system.
- **Input Data:** The file to be shared.
- **Output Data**: File ready to download by other users.
- **Invariants**: The file.
- **Pre-conditions**: User is logged in; file exists on user's computer.
- **Post-conditions**: Any other person to whom the content was made available is able to download it.
- **Basic Flow**: The user uploads a file to be shared using the upload box and selects a subset of other users of the group with whom the user wants to share the file. The file then gets uploaded to the server and desired users are able to download it after logging in.

4.3 Download Files

- **Purpose:** A user wants download a particular file.
- **User:** Any user of the academic portal.
- **Input Data:** Request to download a particular file.
- **Output Data:** File is downloaded on user’s computer.
- **Invariants:** The user and the file.

- **Pre-conditions:** User is logged in and the file must be shared with him or with the group which he is a member of.
- **Post-conditions:** The user has downloaded the file successfully.
- **Basic Flow:** User logs in, selects the file which he wants to download. The file is then transferred from the server to the user's computer.

4.4 Search Result

A user wants to search for a topic using a keyword.

- **Purpose:** A user wants to search for a particular keyword.
- **User:** Any user of the academic portal.
- **Input Data:** The keyword.
- **Output Data:** Search Results.
- **Invariants:** The user and the portal itself.
- **Pre-conditions:** User is logged in.
- **Post-conditions:** Search results
- **Basic Flow:** User logs in, enters the keyword in the search box, clicks the search button and gets the search results.

4.5 Notifications

A user wants to search for a topic using a keyword.

- **Purpose:** To get notifications of assignments, deadlines and other important messages.
- **User:** Any user of the academic portal.
- **Input Data**: User enables the notifications from settings.
- **Output Data**: Notifications.
- **Invariants:** The user and the portal itself.
- **Pre-conditions:** User has chosen to receive notifications.
- **Post-conditions**: User receives the desired notifications.
- **Basic Flow:** User enables the desired notifications from settings and receives notifications.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system should support at least 200 concurrent users. This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of users be able to access the system at the same time, since an academic portal is important to the courses that employ it. The times when the system will be under the most stress are likely during assignment submissions. Therefore, it must be able to handle at least 200 concurrent users

5.2 Safety Requirements

The responsibility of the material to be shared lies with the faculties who upload the documents and he will be responsible for the material which he uploads.

5.3 Security Requirements

1. Passwords will be saved encrypted in the database in order to ensure the user's privacy.

2. The user's IP will be logged.

3. The system will be protected against vulnerabilities such as SQL injection attacks

5.4 Software Quality Attributes

Web design conventions should be consistent with unified user interface between all the components of the “Portal”

5.5 Business Rules

The basic fact here is the material for upload and download is totally dependent on the user’s discretion and network has no responsibility for that matter. It only provides a medium of transport. It’s up user’s sense of morality in which manner he exploits the system

6. Other Requirements

6.1 Logical Database Requirements

All data will be saved in the database: user accounts and profiles, discussion data, messages etc. (except files which are stored on the disk.) The database allows concurrent access and will be kept consistent at all times, requiring a good database design.

6.2 Design Constraints

1. The communication between the portal software and the database will be in SQL.

2. The portal layout will be produced with HTML/CSS.

3. The product will be written in PHP.

4. Angular JS and Bootstrap will be used as Frameworks.

5. The source code must follow the coding conventions of PHP.

6. System administrators must have access to comprehensive documentation