DCIS Secure, Assessment and Course Registration System

A project report submitted in partial fulfillment of the requirements for the award of degree of
Master of Computer Applications
By

09MCMC09	Khushboo Gupta
09MCMC12	Lokesh Kumar
09MCMC19	Rajesh Kumar

Department of Computer and Information Sciences
School of Mathematics and Computer/Information Sciences
University of Hyderabad
Hyderabad-500046

CERTIFICATE

This is to certify that the project report entitled "DCIS Secure Assessment and Course Registration System" submitted by Khushboo Gupta (09MCMC09), Lokesh Kumar (09MCMC12) and Rajesh Kumar (09MCMC19) in partial fulfillment of the requirements for the award of Master of Computer Application is a bonafide work carried out by them under our supervision and guidance.

This project has been partially submitted previously in part to this university by Harishchandra Gangavane (08MCMC05) entitled "DCIS Office Automation System". The current project work is the extension of that project.

Dr. S. Durga Bhavani	Dr. T. Shobha Rani	Anupama Potluri
(Project Supervisor)	(Project Supervisor)	(Project Supervisor)
Associate Professor, DCIS	Assistant Professor, DCIS	Assistant Professor, DCIS

Dean	Head
School of MCIS	Department Of CIS
University of Hyderabad	University of Hyderabad

DECLARATION

We Khushboo Gupta, Lokesh Kumar and Rajesh Kumar hereby declare that this project report entitled "DCIS Secure Assessment and Course Registration System" submitted by us under guidance and supervision of *Ms. Anupama Potluri*, *Dr. T. Shobha Rani* and *Dr. S. Durga Bhawani* is a bonafide work carried out by us.

We also declare that it has been partially submitted previously in part to this university by Harishchandra Gangavane (08MCMC05) entitled "DCIS Office Automation System". The current project work is the extension of that project.

Khushboo Gupta	Lokesh Kumar	Rajesh Kumar
09MCMC09	09MCMC12	09MCMC19

ACKNOWLEDGEMENT

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed project supervisors **Ms. Anupama Potluri**, *Assistant Professor, DCIS*, **Dr. T. Shobha Rani**, *Assistant Professor, DCIS* and **Dr. S. Durga Bhawani**, *Associate Professor, DCIS*, for their valuable guidance, encouragement and help for completing this project. Their useful suggestions, fruitful discussions on design and implementation of this project and their co-operative behaviour is sincerely acknowledged.

We are also grateful to the respected faculty members of DCIS for their constant support, valuable feedback and guidance.

Our thanks and appreciations also go to our colleague *K. N. Jagannadharao(10mcmt62)* in developing the project and having valuable discussions with us.

Last but not the least we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

Khushboo Gupta (09MCMC09)
Lokesh Kumar (09MCMC12)
Rajesh Kumar(09MCMC19)

Abstract

Office work in a department of a university involves many activities such as registration of incoming students, keeping track of their attendance and assessment, registration of courses to students, allocation of courses to faculty and so on. A lot of this work is tedious and involves collating information from various sources. This can be best achieved through computerization. The main objective of this project is to automate the office work of Department of Computer and Information Sciences (DCIS). This project titled "DCIS Secure Assessment and Course Registration system (DSACR)" will help department office to keep record of students and faculty. DSACR will provide facility to register new students, search, edit, remove records from database etc. The software is a web based application that will help DCIS with easy access in a secure way.

Security is achieved through secure transport layer and Role-Based Access Control (RBAC) that gives different authorization to different roles. The users of the system are mapped to these roles based on which their access rights are determined. Course Assessment module allows for a flexible and generic way of entering marks by the faculty. It helps the staff to track students' backlog information so that it is easy to determine the students status – promoted and not promoted. It also allows a formula to be entered for the grades so that the grades and CGPA are computed automatically with the flexibility to modify this on demand. The Course Registration module allows allocation of subjects to faculty and eases the task of course registration for students. The system provides the flexibility to specify pre-requisites and limits per stream per subject so that only *limit* number of students who satisfy the pre-requisites are allowed to register for a subject. It also allows for multiple faculty to share a subject.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Office work is a very tough and tedious process. At the start of every semester, there is a lot of activity on the academic front such as storing new students' information, course registration etc.. During the course of the semester, the office and faculty need to maintain information regarding attendance and assessment. At the end of every month/semester, the office needs to prepare a statement giving the total percentage attendance/grades received by a student collating information from various faculty. In this project, we developed a web application that simplifies a lot of this activity by maintaining a database of information that can be updated online. The job of collating the information is now automated through the software. This eases the work load of office staff. In addition, all queries of students and/or faculty/staff can be processed online without resorting to human intervention, once again reducing the work of the office staff. The final aim would be to have a paperless office.

1.2 About The Project

This project titled "DCIS Secure Assessment and Course Registration System (DSACR)" is the extension of the DCIS Office Attendance System (DOAS). DOAS contains core idea of Attendance System. In DSACR we implement this idea and extend the work. Currently DSACR is helping the DCIS office to manage the students' information and all attendance work is done by this web application. Basically DSACR contains four modules.

1. Attendance Module

2. Course Registration Module
3. Course Assessment Module
4. Security Module

1.3 Modules done by us

1. **Course Registration Module:**
   In this module all the course registration has to be done before start of a new semester. By using this module, the faculty/staff in-charge of course registration can allot the courses to the students according to their given preferences and their CGPA or GATE score.

2. **Assessment Module:**
   In this module all the assessment related activity will be done. By using this module staff and faculty can submit and see major and minor exams marks. They can submit using ".csv" files or through online updation.

3. **Security Module:**
   This module takes care of Security aspects and Authentication and Authorization of users of the system. The data is sent encrypted over the network to prevent eavesdropping and all data integrity checking and validation is done. Users are authenticated with the LDAP server and they are navigated to their respective pages on successful login. We implement the Role-Based Access Control (RBAC) system by mapping each user to a role which determines the access that is authorized for this role.

Chapter 2

Principles of Database and UI design

The success of a web application lies in, how good the backend database design is. So, inorder to achieve the best database design we have, followed the principles of database design as mentioned in the below sections.
The user interface of an application will often make or break it. Although the functionality that an application provides to users is important, the way in which it provides that functionality is more important. An application that is difficult to use wont be used. It wont matter how technically superior our software is or what functionality it provides, if your users dont like it they simply wont use it. So we should not underestimate the value of user interface design nor of usability. We considered the principles of user-centered interface design as mentioned in the below sections, in our project.

2.1 Database Design

2.1.1 The purpose of our database

The purpose of our database design is to organize, store, and retrieve the information of DSACR, as efficient and effective as possible. Our purpose is to have a reliable data storage.

2.1.2 The information required

The very first step is to determine what are all the information that is required to store and retrieve from the database. DSACR requires complete information about faculty, subjects, students, courses/streams, assessment and course registration.

2.1.3 Dividing the information into tables

All the information is divided into tables of the database. The database tables are created from converting E–R diagrams into tables of the database. The table names are meaningful and gives an idea of what it contains. we have maintained the tables as mentioned in Design Document chapter.

2.1.4 Turning information items into columns

The information is further represented by columns of a table, the columns basically represent the attributes of the table. The attributes like faculty id, faculty name and course id represent the columns of different tables.

2.1.5 Specifying primary keys

In order to uniquely identify a particular row of the table, the primary keys are needed. The primary keys are determined based on E–R diagrams and based on the need of the application. Primary key ensures referential integrity.

2.1.6 Setting up the table relationships

The relationship among the tables is determined with the help of E–R diagrams. This relationship is maintained with the help of primary keys and foreign keys.

2.1.7 Refining the design

We refined our design based on the needs and based on the testing of the database for our application.

2.1.8 Applying the normalization rules

At last we applied normalization in order to eliminate redundant data and ensure logical dependencies. We have normalized our tables upto 3rd normal form.

2.2 User Interface Design

2.2.1 Consistent look and feel

In user interface design, consistency between pages, functions, and options is vital. Users come to expect certain things as they use the application, and if the interface changes from one page to the next, it is both confusing and frustrating.

We have maintained consistent look and feel throughout the system.

2.2.2 Users are informed about the error

The user needs to feel that their actions have meaning. We didn't left any doubt in users mind,gave them feedback for their actions. We achieved this very nicely, by doing exhaustive testing and by providing error pages and alert boxes at each and every step and hence we inform our users what actually went wrong and what is the actual cause of the problem, when some error occurs, for example:

- the user is informed about the invalid user name and password.
- the user is informed about expiration of session if session is expired.
- the user is informed about javascript if it is disabled and help is provided to enable it.
- the user is informed when some required fields are left empty.

2.2.3 Highlight Information

We are using different color combinations to give users a visual feedback and to attract users eyes on some items. Like we color red, if attendance is less than 75% of a student. Thus a user can visually distinguish between different types of entries.

2.2.4 Efficient Workflow

People have their own predetermined ways, they use particular types of software or application, the design of DSACR interface takes care of the complete workflow of the users.
For example, the faculty (user) after filling the attendance of a particular

subject would like to see the summary on submit, and they would like to take print out of the page.

2.2.5 Hover controls

Putting fewer things on the plate would also make the dish easier to swallow. If the application has too many controls, the users will waste most of their time scanning through all the controls, to find what they are actually looking for. We hide the controls, but then show them when the user is hovering over certain menu item, thus we have simpler interface and no loss in functionality.

2.2.6 Context based controls

The context based approach allows to show fewer controls at any given time, but at the same time, more controls that are relevant to the task at hand. In our system each menu tab displays a set of controls relevant to any given task, like attendance tab has only view, update and summary control.

2.3 Concluding Remarks

A good way to test any interface is by watching people use your system in a real-world scenario. And that too we did it, all users were able to navigate around and achieve their objectives with relative ease and the interface proved intuitive, to both experienced and less experienced computer users.

Chapter 3

Specification Document

3.1 Hardware Specifications

Processor

- 166-megahertz (MHz) or higher processor Memory 128 MB of RAM is needed.
- 256 MB recommended.

HardDisk

- Minimum Requirement: 5 GB
- Server Memory Requirements:
  - Mysql and Database Connector: 70 MB
  - Apache Tomcat 7.0.23: 60 MB
- Tools Requirement:
  - Software Requirements: 10 MB
- Disk Drives CD ROM, CD RDWR ROM
- Communications Internet: 256 Kbps Modem (for supporting internet)

Display

- VGA or Higher Resolution monitor

3.2 Software Specifications

3.2.1 OpenLDAP

OpenLDAP Software is a free, open source implementation of the Lightweight Directory Access Protocol (LDAP) developed by the OpenLDAP Project. We are using OpenLDAP 2.4.31 [1] which in turn uses Version 3 of LDAP.
Why OpenLDAP?

1. LDAP is a lightweight alternative to the X.500 Directory Access Protocol (DAP), it uses TCP/IP stack verses the overly complex OSI stack.
2. LDAP is a platform–independent protocol.
3. Directory itself is a tree–structured, read–optimized database.
4. LDAP also has other simplifications, such as the representing most attribute values and many protocol items as textual strings, that are designed to make clients easier to implement.
5. We are using OpenLDAP to provide a central authentication location for user logins anywhere on the network.
6. LDAP is not in any way related to traditional system usernames or other data. However, part of its functionality in our setup will consist in storing information traditionally found in Unix files /etc/passwd and /etc/group, thus making that data network–accessible at a centralized location.

3.2.2 JAVA

We are using jdk–1.7[2] in the development of the application.
Why Java?

1. Simpler, easier to read programs, more efficient
2. Reuse of code
3. More robust, error-free code
4. Java is Platform Independent.

3.2.3 JSP

We are using JSP 2.0[3] for the development of our application.
Why JSP?

1. Java Server Pages (JSP) technology is the Java platform technology for delivering dynamic content to web clients in a portable, secure and well-defined way.

2. The Java Server Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML, and XML templates and Java code.

3. JSP has been built on top of the Servlet API and utilizes Servlet semantics.

4. JSP technology brings the "Write Once, Run Anywhere" paradigm to interactive Web pages. JSP pages can be moved easily across platforms, and across web servers, without any changes.

3.2.4 APACHE TOMCAT

To process a JSP file, we need a JSP engine that can be accommodated inside a web server. So we are using *Apache Tomcat 7*[4].
Why Tomcat 7.0.23?

1. Implements the Servlet 2.4 and JSP 2.0 specifications.

2. Reduced garbage collection, improved performance and scalability.

3. Faster JSP parsing.

3.2.5 MYSQL

Mysql is a relational database management system, which organizes data in the form of tables. We are using MySQL–community–server–5.1[5].
Why MySQL?

1. Mysql is a database servers based on RDBMS model.

2. It manages a seer of data that attends three specific things i.e. data structures, data integrity and data manipulation.

3. With Mysql co-operative server technology we can realize the benefits of relational systems for all the applications.

4. Mysql makes efficient use of all system resources, on all hardware architecture which reduces the cost.

5. Mysql is Open source and also scalable in nature.

3.2.6 SQL

The Structured Query Language (SQL) is a non-procedural query language. It is a standard language for all DBMSes; used for creating, managing, manipulating and querying the database objects such as tables, views etc. It is categorised into:

1. Data Definition Language-*CREATE, ALTER, DROP*

2. Data Manipulation Language-*INSERT, UPDATE, DELETE*

3. Data Query Language-*SELECT*

4. Data Control Language-*GRANT, REVOKE*

5. Transaction control Language-*COMMIT, ROLLBACK*

3.2.7 JDBC

Java Database Connectivity (JDBC)[6] is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases.

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection.

3.2.8 JNDI

The Java Naming and Directory Interface (JNDI)[7] is a Java API for a directory service that allows Java software clients to discover and look up data and objects via a name. Like all Java APIs that interface with host systems, JNDI is independent of the underlying implementation. Additionally, it specifies a service provider interface (SPI) that allows directory service implementations to be plugged into the framework.
The JNDI API is used by the Java RMI and Java EE APIs to look up objects in a network. The API provides:

- A mechanism to bind an object to a name.
- A directory lookup interface that allows general queries.
- An event interface that allows clients to determine when directory entries have been modified.
- LDAP extensions to support the additional capabilities of an LDAP service.

3.2.9 HTML

The Hyper Text Markup Language (HTML)[8], provides a means to describe the structure of text-based information in a document by denoting certain text as links, headings, paragraphs, lists, and so on. We are using HTML 4 for the development of the application.

It also provides a means to supplement the text with interactive, forms, embedded images and other objects. HTML defines several data types for element content, such as script data and style sheet data, and excess of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

3.3 Client Requirements

- **Operating System** GNU/Linux / WINDOWS 7 / XP Professional / NT
- **Web Browser** Mozilla Firefox / Google Chrome

3.4 Server Requirements

- **Operating System:** GNU/Linux / WINDOWS 7 / XP Professional / NT
- **Servers** Apache Tomcat 7 web server, OpenLDAP server
- **Web Browser** Mozilla Firefox / Google Chrome
- **Database** Mysql
- **Languages** J2SE2.0
- **Scripting** HTML, DHTML

Chapter 4

Product Administration

This chapter gives administrative details of authentication, authorization and security features of the product. The following sections describe the details of these features.

4.1 CA certificate

To enable Transport Layer Security (TLS)[9], we need to have a certificate signed by CA. We created a Root CA to self sign the certificates.

- The root CA having validity of 10 years(3650 days) is created with the help of openssl[10]. The figure 4.1 shows an screenshot of it.
- We created a new Certificate Signing Request(CSR) and get it signed by the Root CA, thus having a self signed certificate. Since the intention is to use it only in our organization/department, it is the best way of signing the certificate. The new certificate signed by the root CA is shown in the figure 4.2.
- We used the certificate signed by RootCA to enable transport layer security, also the *https* container needs this certificate. Thus we can enable TLS in Tomcat server in order to prevent eavesdropping.

4.2 Secure HTTP

We are using *https* instead of *http* to have transport layer security. The data sent over the network is encrypted to minimize the risk of eavesdropping. Since we are using self signed certificate so when a user visits the page for the first time, the web browser shows an alert of untrusted connection. If

```
.inux-n6gw:/etc/ssl/myCA # openssl req -new -x509 -extensions v3_ca -keyout ./private/cakey.pem -out c
Generating a 1024 bit RSA private key
.....................++++++
.....................................................++++++
writing new private key to './private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IN]:IN
State or Province Name (full name) [Some-State]:Andhra Pradesh
Locality Name (eg, city) []:Hyderabad
Organization Name (eg, company) [Internet Widgits Pty Ltd]:dcisonline uohyd ernet in
Organizational Unit Name (eg, section) []:dcis
Common Name (eg, YOUR name) []:lokesh_kumar
Email Address []:lokesh_kumar@dcis.uohyd.ernet.in
.inux-n6gw:/etc/ssl/myCA #
```

Figure 4.1: screenshot of rootCA creation

```
File  Edit  View  Search  Terminal  Tabs  Help
lokesh@linux-g227.site:~   lokesh@linux-g227.site:~   lokesh@linux-g227.site:~
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1048577 (0x100001)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=IN, ST=Andhra Pradesh, L=Hyderabad, O=dcisonline uohyd ernet i
n, OU=dcis, CN=lokesh_kumar/emailAddress=lokesh_kumar@dcis.uohyd.ernet.in
        Validity
            Not Before: Feb  6 13:04:09 2012 GMT
            Not After : Feb  5 13:04:09 2013 GMT
        Subject: C=IN, ST=AP, L=Hyderabad, O=dcisonline, OU=dcis, CN=lokesh
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:b7:c0:c1:2b:1e:4c:f1:54:5e:0d:60:9c:54:88:
                    5e:4c:7d:45:5a:af:33:1d:96:67:20:65:89:8d:88:
                    8f:26:06:b0:ef:fe:e3:41:82:57:c1:b0:e0:ae:f8:
                    5d:50:20:73:0e:1c:05:4c:58:07:6b:94:ea:0f:5f:
                    cc:87:c5:53:b0:21:9b:f4:9b:64:33:c2:f5:01:2e:
                    92:f6:47:22:2b:0a:1e:2c:02:a5:78:f5:91:d2:2e:
                    ae:45:35:91:2e:24:f8:96:61:f9:80:3f:71:05:57:
                    cc:0c:a4:21:1e:4e:c1:e9:27:29:f3:a1:05:e2:8e:
../newcert.pem lines 1-22/62 35%
```

Figure 4.2: Screenshot of a new certificate signed by rootCA

the user adds this site as trusted site and adds permanent security exception, then this warning will not be issued by the browser.

- The figure 4.3 shows the warning given by mozilla firefox browser when a user visits it first time.

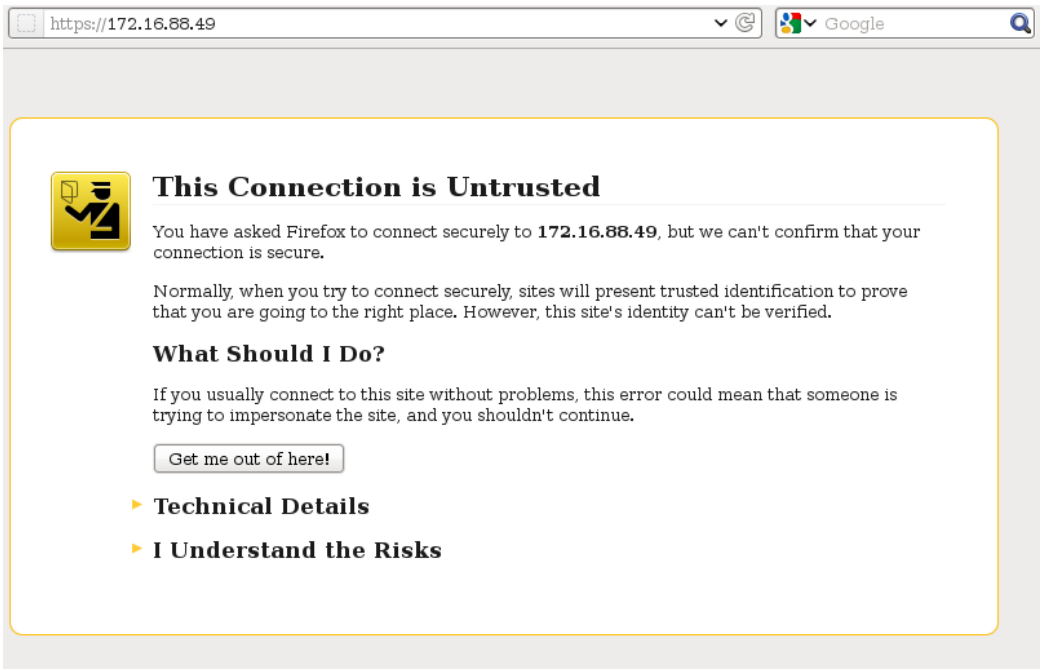

Figure 4.3: Warning untrusted connection

- The technical details can be displayed by clicking on the **Technical Details** link. To add the site as an exception the user can click on **I Understand the Risks** link. The figure 4.4 shows the technical details.
- The user should check permanently add exception (if browser gives this option) and click on confirm security exception in order to avoid the warning on subsequent visits to the site. The figure 4.5 shows an screenshot of adding security exception in firefox.
- To see the site info, user can click on address bar, it shows info as in figure 4.6 and on clicking on more info, the more detailed information can be viewed by the user.
- If the user wants to get the technical details of the site she is visiting, can get as shown in figure 4.7, which shows that your data is encrypted

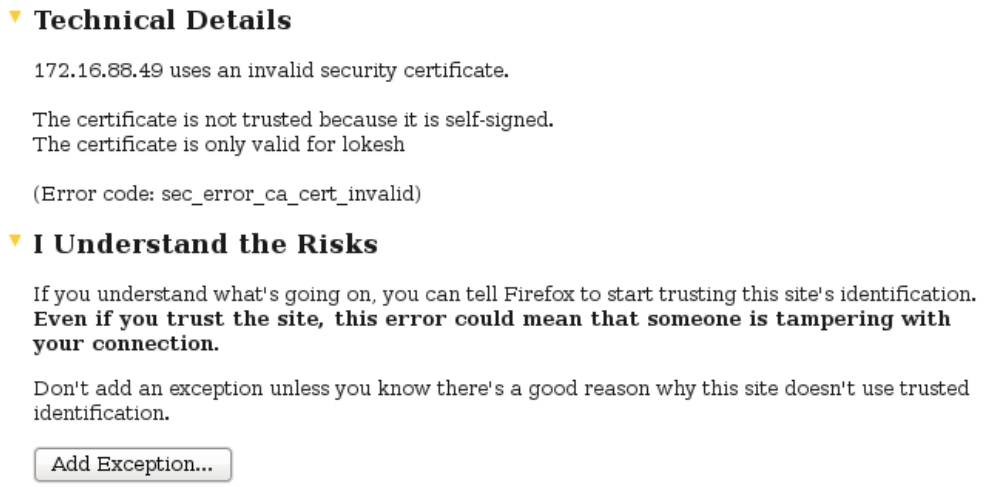

Figure 4.4: Technical details

before being transferred over the network using 128–bit AES encryption scheme.

- The more details about the certificate issued by the server can be viewed by clicking on view certificate in technical details as shown in figure 4.7. The figure 4.8 shows details of the certificate, the issuer, its validity and fingerprint information.

4.3 Centralized authentication

The centralized authentication is done with the help of LDAP server. The user enters his/her username and password on the login page. On clicking login, it contacts the LDAP server with the given credentials for authentication. If the login is successful, then the user's role is retrieved from the LDAP server and the response is redirected to the specific role main page. The figure 4.9 shows a flow chart of login process.

4.4 More on Security features

Apart from HTTPS, our system has the following additional security features:

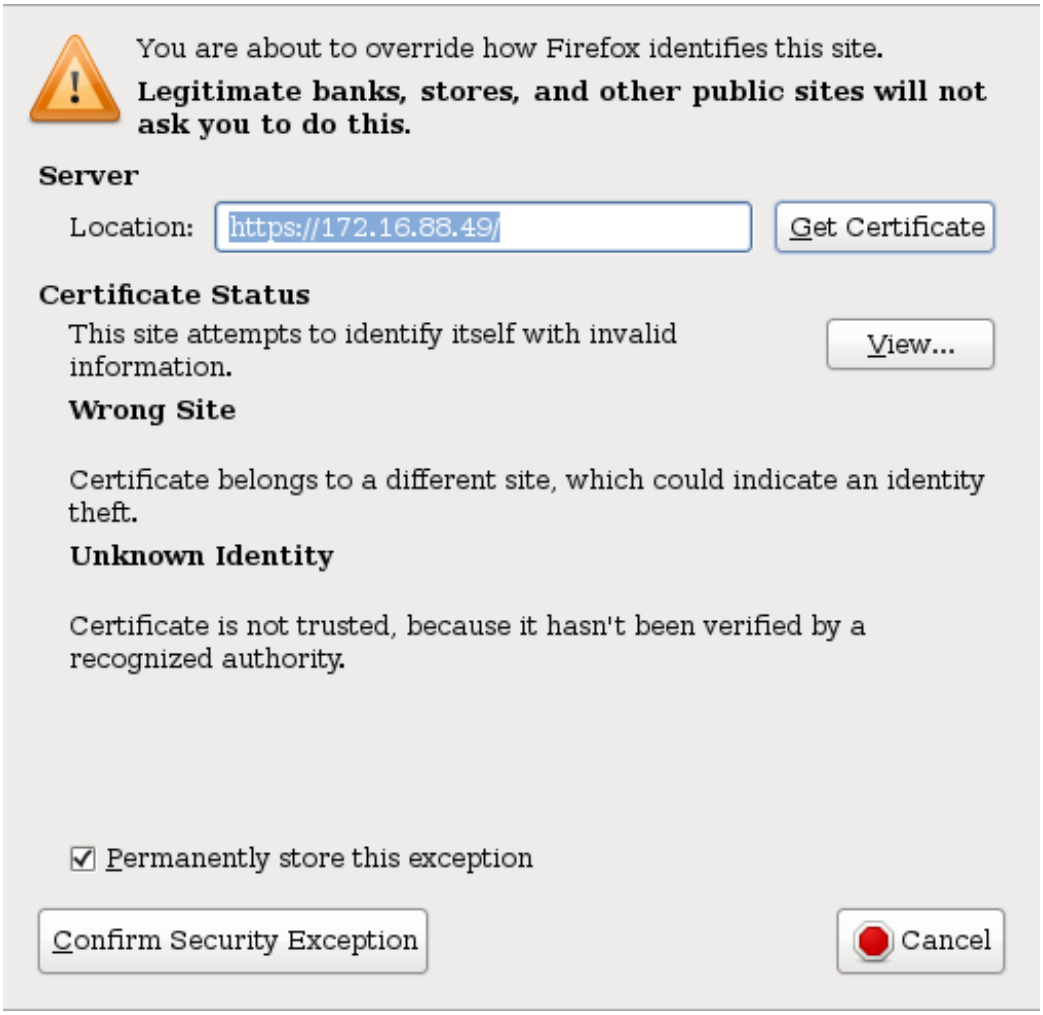

Figure 4.5: Screenshot of add security exception

4.4.1 Role based Access Control

Access to the resources is based on the role of the user as per the RBAC principles. The system currently has student, faculty, staff and admin roles. Based on the role of a user, the page is redirected to his/her main page and the data specific to the user is loaded. The access permissions such as read, write or read and write are determined based on the role of the user.

4.4.2 Secure Against SQL injection

SQL Injection Attack (SQLIA)[11] is considered one of the top 10 web application vulnerabilities of by the Open Web Application Security Project.

Figure 4.6: Encrypted connection

Our system is secure against SQL injection. This is ensured in two ways

1. With the use of prepared statements

   Prepared statement is a feature used to execute the same or similar database statements repeatedly with high efficiency. The prepared statement takes the form of a template into which certain constant values are substituted during each execution.
   The typical workflow of using a prepared statement is as follows:

   (a) **Prepare:** The statement template is created by the application and sent to the DBMS. Certain values are left unspecified, called parameters, placeholders or bind variables (labelled "?" below):
   INSERT INTO PRODUCT (name, price) VALUES (?, ?)

   (b) The DBMS parses, compiles, and performs query optimization on

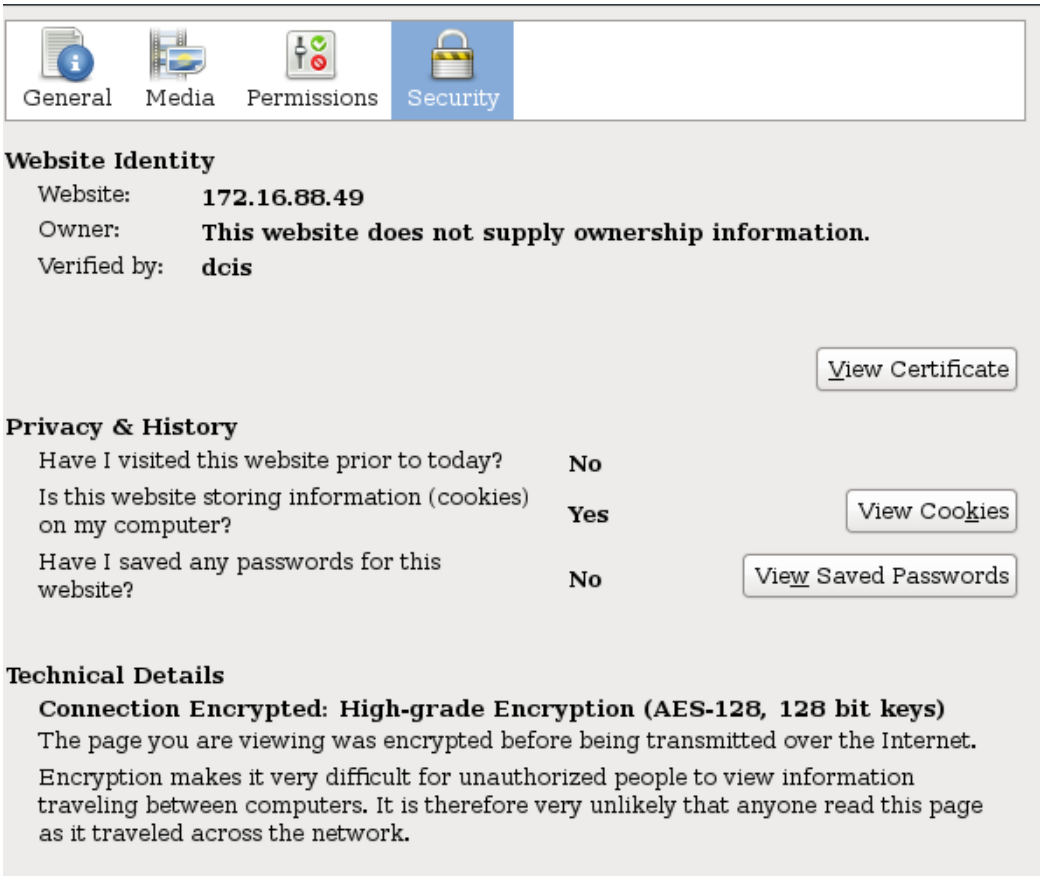

Figure 4.7: Technical details of encrypted connection

the statement template, and stores the result without executing it.

(c) **Execute:** At a later time, the application supplies (or binds) values for the parameters, and the DBMS executes the statement possibly returning a result.

As compared to executing SQL statements directly, prepared statements offer two main advantages:

(a) The overhead of compiling and optimizing the statement is incurred only once, although the statement is executed multiple times.

(b) Prepared statements are resilient against SQL injection, because parameter values, which are transmitted later using a different

Figure 4.8: Screenshot of server certificate

protocol, need not be correctly *escaped*. If the original statement template is not derived from external input, SQL injection cannot occur.

We supply parameters to the prepared statements to be executed, only after validation; thus we are safe from SQL injection attack.

2. No unauthorised access

   A page cannot be accessed without a successful login. If someone tries to access a page directly then system throws an error message stating that you are not logged in, thus minimizing the vulnerability to SQL injection.

4.4.3 Checks for javascript

In order to use our system, javascript must be enabled in the browser. The system checks if it is enabled or not. If it is not, then, it gives an error message and shows help on how to enable javascript.

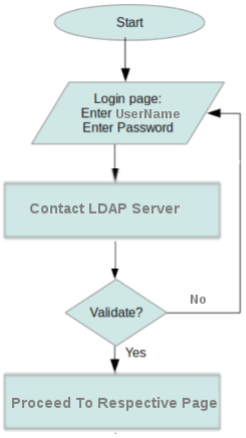

Figure 4.9: Flowchart showing login process

4.4.4 Login activity

On login we display the IP address of the system the user is using. The users can also see their last successful login details by clicking on **login activity** link under the menu item **Welcome User**. This gives more details about the last login activity such as the date and time, IP address and the user agent used. This helps the users to know if their account is compromised and enables them to easily catch any unfamiliar activity.

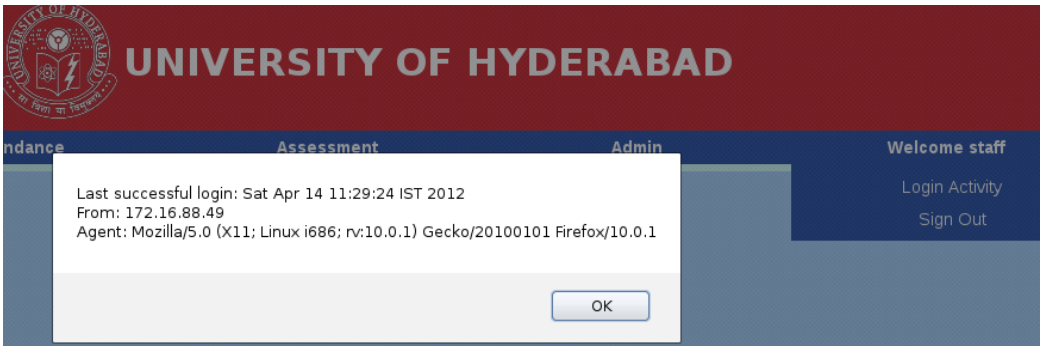

Figure 4.10: Screenshot showing login activity of a user

Chapter 5

DSACR Design Document

This chapter gives an overall picture about the architecture of our project. It specifies the issues that are to be taken care of and guidelines which guides the design of the database.

5.1 Database Design

5.1.1 Requirement

DSACR system should deliver three modules, one is assessment second is course registration and third is security. In order to deliver these functionalities we have to identify the various relationships that exist between different components in these systems. The system can be described as follows:

- All courses and electives offered in a semester for each stream.
- Registered students and the subjects opted for by each student in each semester.
- Faculty details for each subject and all students marks and grade for that subject.
- Information about the recourse and re-admission students.

In order to cater to these requirements DSACR has been normalized till third normal form (3NF). After normalization, DSACR each module has a number of normalized tables. Section 5.1.6 gives the relationships between different entities in the project and Section 5.1.7 gives the translation of these ER diagrams into relational tables which are normalized upto 3NF.

5.1.2 Entities

The entities in DSACR system are:

1. Faculty
2. Subject
3. Stream
4. Student
5. Assessment

The above mentioned entities are identified and used to construct the whole DSACR system.

5.1.3 Relationship between entities

1. Each faculty can teach many subjects as well as each subject can be taught by many faculties.
2. Each stream can have many students but one student cannot belong to many streams.
3. Each stream can have some set of subjects and each subject can belongs to many streams.
4. Each student will have assessment for each subject to which they register and assessment of one subject can be related to many students.
5. Each subject will have one and only one assessment.
6. Each stream has one master table(student table).

5.1.4 Attributes

- Faculty: Faculty Id, Faculty Name.
- Subject: Subject Id, Subject Name, Credits, Type (core subject or elective).
- Stream: Stream Name.
- Master Table: Student Id, Student Name, subjects, grades, CGPA, GATE score.

- Assessment: Reg number, Student Name, Minor 1, Minor 2, Minor 3, Internal marks, Major, Total marks, Grade.
- Elective: Course name, Stream names, Pre-requisites.
- Login Activity: User id, last login, failed attempts, IP address, agent.

5.1.5 Cardinality Mapping

- Faculty to Subject: Many to Many
- Stream to Student: One to Many
- Stream to Subject: Many to Many
- Student to Assessment: Many to Many
- Subject to Assessment: One to One
- Stream to Master table: One to One

5.1.6 Entity Relationship Diagrams

- ER Diagram for Assessment module is shown in the figure 5.1

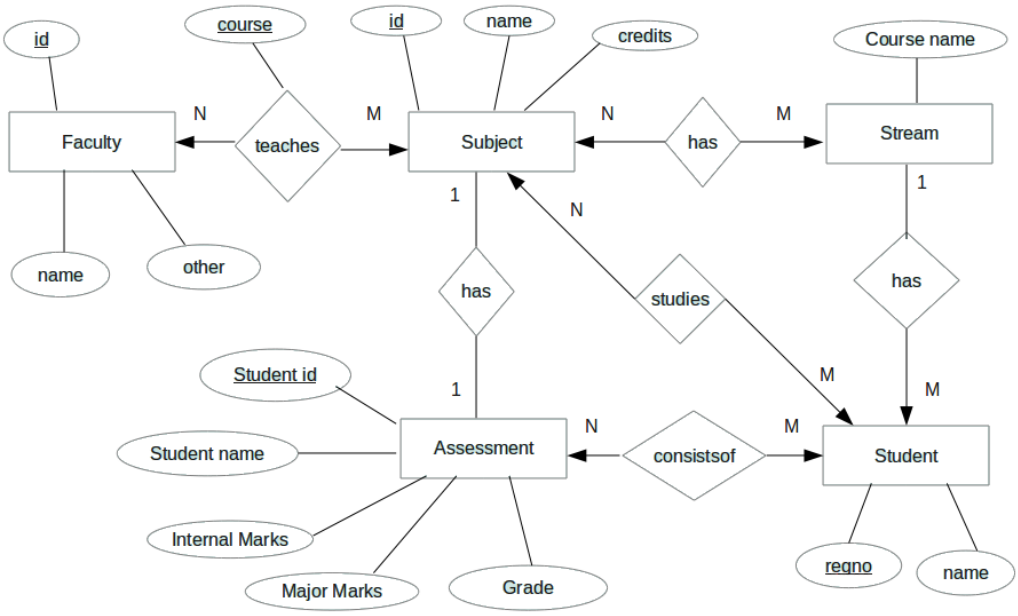

Figure 5.1: ER Diagram for Assessment module.This diagram represents all the entities involved in Assessment module as shown in rectangular boxes. The relationships between entities is shown in diamond boxes. Attributes is shown in ovals. Cardinality is shown as arrows between the entities.

- ER Diagram for Master table is shown in the figure 5.2

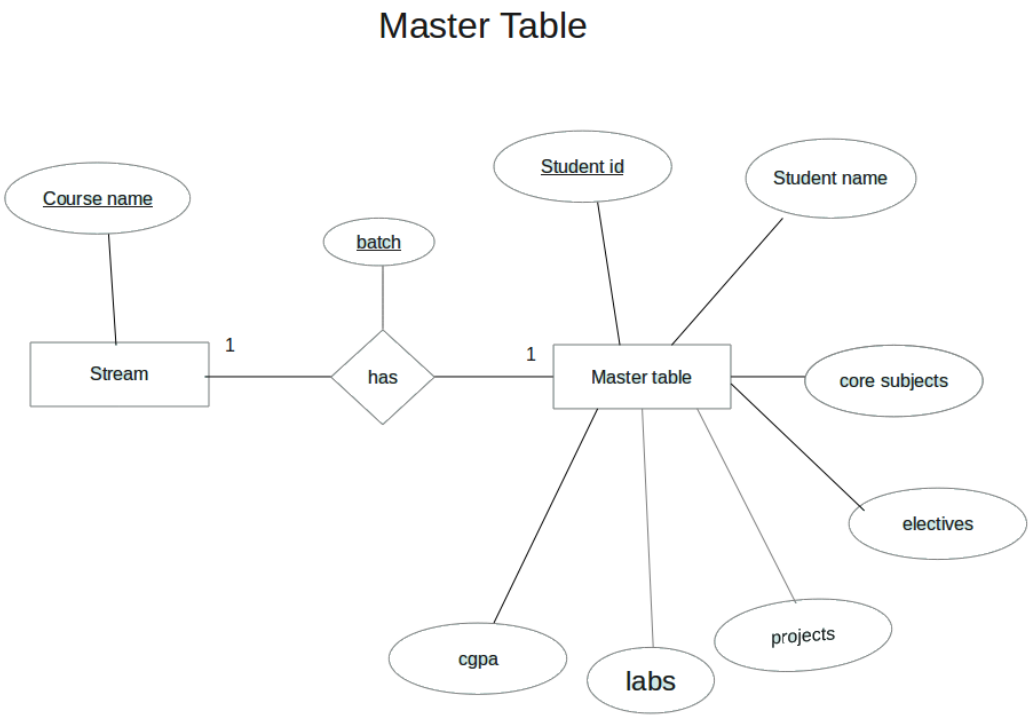

Figure 5.2: ER Diagram for Master table. Master table stores the student details, subject details, grades and cgpa.The combination of Student and Stream entities in ER diag 5.1 will give us Master Table.

5.1.7 DSACR Tables

The ER diagrams are converted into a set of relational schema. Each entity(strong and weak), binary relationships, n-ary relationships are converted into relations. A relational database schema can contain a set of relational schemas. Normalization of the realtional schema is proposed by Codd in order to achieve the desirable properties of i) minimizing redundancy and ii) minimizing the insert, delete and update anomalies. Here we have strove to achieve third normalization for our tables in the database.

Master Table

Master table records the student's whole information, like student's registration number, name, all subjects studied by the student during the complete course and grades obtained, CGPA, it will also contain GATE score for only MTech stream. Figure 5.3 shows Master Table for MCA and the figure 5.4 shows Master Table for MTech stream. Each stream has a set of tables which are created at the time of joining of the students in an academic year and are mainained throughout their period of study.

- Time of creation:
  These tables will be created at the beginning of the session. There will be one master table for each batch. Initially all grades will be tagged as NR(Not Registered) for all respective courses.
- Activities that update table:
  During course registration, all current semester subjects grade will be modified to R(Registered), same in the case of elective registration. At the end of course assessment these R will be modified to respective grade of the subjects.
- Events triggered by this table:
  After updation in this table at the time of course registration activity, subject-attendance tables also gets updated.

StudentId	StudentName	Subj1	Grade1	...	cgpa
10MCMC01	Ajay Singh	CA504	R	NR	NULL

Figure 5.3: Master Table for MCA, here primary key is StudentId

StudentId	StudentName	Subj1	Grade1	...	cgpa	gate
10MCMT02	Allu Shiva	CA527	R	NR	NULL	45

Figure 5.4: Master Table for MTech, here primary key is StudentId

Current Session Table

Current Session Table provides information about the current semsester, its starting date and end date. Figure 5.5 shows current_session table

- Time of creation:
  This table is created at the time of initialization of the application and gets populated at the beginning of each new session.
- Activities that update table:
  For every new semester when Subject-Faculty-Registration Activity happens, this table will get updated with session name, start date and end date.

name	start	end
Winter	2012-01-02	2012-05-02

Figure 5.5: Session Table, here primary key is name

Faculty Data Table

Faculty Data Table provides complete information about faculty. It contains faculty Id and names. Figure 5.6 shows faculty_data table

- Time of creation:
  This table is created once, during initialization process.
- Activities that update table:
  Whenever a new faculty comes, this table gets updated.
- Events triggered by this table:
  No events will be triggered.

ID	Faculty_Name
knmcs	K. Narayana Murthy

Figure 5.6: Faculty Data Table, here primary key is ID

Subject Database Table

Subject Database Table provides complete information about each and every subject offered in the department. Type specifies whether it is a core subject or elective. Figure 5.7 shows subject_database table

- Time of creation:
  This table is created at the time of initialization of the system.

- Activities that update table:
  Whenever a new subject is introduced, this table gets updated.
- Events triggered by this table:
  No events will be triggered.

Code	SubjectName	credit	type
CA504	Theory of Computation	3	C

Figure 5.7: Subject Database Table, here primary key is Code

Login Activity Table

Login Activity Table provides complete information about login activity of the user when they login into the application. Figure 5.8 shows Login Activity_history table.

- Time of creation:
  This table is created at the time of initialization process of the system.
- Activities that update table:
  Whenever a user login into the system, this table gets updated.
- Events triggered by this table:
  No events will be triggered.

userid	lastlogin	failedattempts	ipaddress	agent
alokcs	2012-04-24	2	172.16.88.94	Mozila 3.1

Figure 5.8: Login Activity Table, here primary key is userid

Subject Faculty Table

Subject Faculty Table provides information about which faculty is teaching which subject. Figure 5.9 shows subject_faculty table

- Time of creation:
  This table is created at the time of Subject Faculty Registration Activity. This activity shows in figure 5.34

- Activities that update table:
  Subject Faculty Registration Activity will update this table.
- Events triggered by this table:
  Whenever this table gets updated, the elective subjects are inserted in the elective table, represented by the figure 5.12.

SubjectId	FacultyId
CA504	alokcs

Figure 5.9: Subject faculty table, here primary key is SubjectId and FacultyId, SubjectId is foreign key to Subject Database Table 5.7 and FacultyId is foreign key to Faculty Data Table 5.6

Grade Table

Grade Table uses a grade formula, based on which the grades will be decided. Figure 5.10 shows stream_grade_table.

- Time of creation:
  At the beginning of the session, during initialization process.
- Activities that update table:
  Whenever form 5.33 will be submitted by the staff.
- Events triggered by this table:
  No events will be triggered.

Grade1	Marks1	...	Grade7	Marks7
A+	100	...	F	40

Figure 5.10: Grade Formula Table

Curriculum Table

Curriculum Table provides complete information about the subjects offered to a particular stream, subjects names and credits. This table also gives information about the number of core subjects, labs, projects, and electives offered throughout the course of study for that particular stream, and its semester field specifies which subject is the core subject for that particular semester. Figure 5.11 shows stream_curriculum table.

- Time of creation:
  This table is created and populated at the beginning of the session. There will be one curriculum table per each stream.

- Activities that update table:
  Whenever a new subject is registered for a particular stream.

- Events triggered by this table:
  At the time of course registration for students this table will get updated and after this the course which is registered for a student, that StudentId will be inserted into the corresponding attendance table and assessment table.
  Example: If CA505 is updated in master table for StudentId 10MCMC01. Then 10MCMC01 will be inserted in Winter-2012-CA505.

SubjId	credits	semester
CA504	3	MCA_I

Figure 5.11: Mca curriculum table, here primary key is SubjId

Elective Table

Elective table consists of courses offered as electives for each stream and number of students that can register for that elective from all streams. It also specifies the pre-requisites for each course if there are any. Figure 5.12 shows elective_table.

- Time of creation:
  This table is created at the time of elective registration.

- Activities that update table:
  This table is populated at the time of Elective-Registration-Activity. The activity shows in figure 5.36

- Events triggered by this table:
  After insertion in this table no events will be triggered.

Assessment Tables

Assessment Tables provide complete information about the assessment of each subject. Figure 5.13 shows Assessment_session_currentyear_subjectId table.

Course_name	MCA_I	...	MTech_CS_I	...	Pre_req_1	Pre_req_grade1	...
AA	0	0	20	0	CA504	B+	

Figure 5.12: Elective Table, here primary key is Course_name

- Time of creation:
  These tables will be created at the time of subjects and elective registration. There will be one table per subject.
- Activities that update table:
  This table gets updated, whenever minor marks, or major marks are updated by the faculty, or whenever a new student is registered for that subject, or if some student is deleted from the list.
- Events triggered by this table:
  Whenever major marks are updated, this table will trigger an event to update the grades and cgpa even in the master table.

Reg_num	Minor_1	Minor_2	Minor_3	InternalMarks	Major	TotalMarks	Final
10MCMC01	18	16	18	36	44	80	A

Figure 5.13: Subject assessment table, here primary key is Reg_num

5.2 Architecture Design

Architecture design is an early stage of the system design process. It involves identifying major system components and their communication. The data flow diagrams illustrate the flow of data in the system. They capture the input, processing and output of different modules.

5.2.1 Data Flow Diagrams

Level 0 DFD: The level 0 DFD, shown in Figure 5.14, of the assessment module represents the whole module. The users of the system are Admin, Faculty, Staff and Students. This module accesses the data from the assessment database and files. Using this module Admin, Faculty, Staff and Students can perform several activities such as Student Assessement View, Staff Assessment View, Faculty Assessment Update and View.

Example: Student can perform only Assessment View Activity. Staff and Faculty both can perform Assessment View Activity but Faculty can also do Assessment Update Activity.

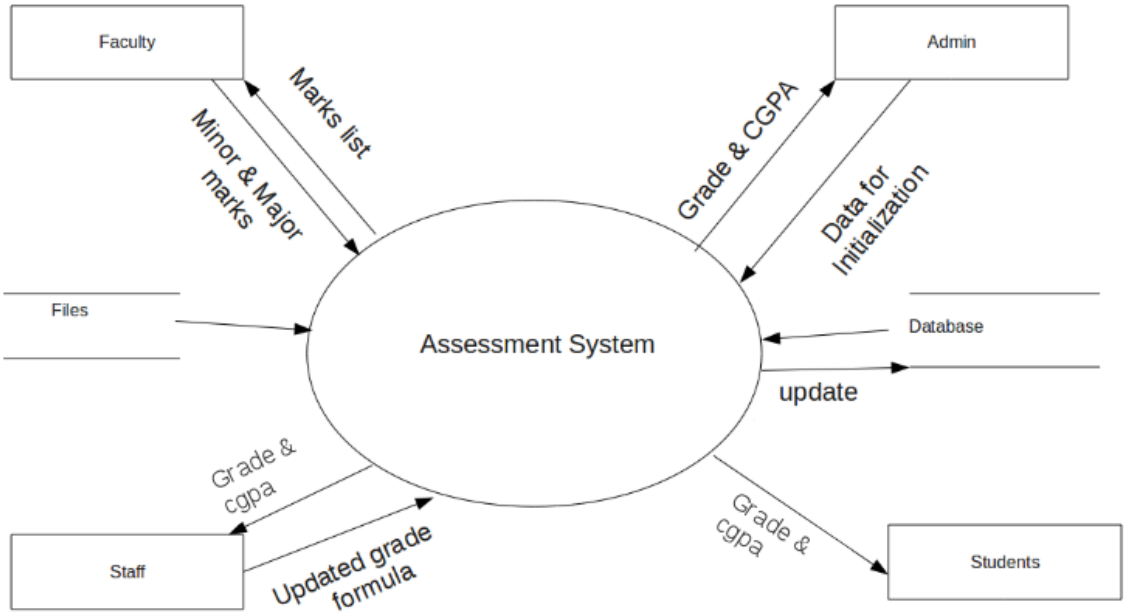

Figure 5.14: Level 0 DFD for Assessment module

Level 1 DFD: The Level 1 DFD is shown in Figure 5.15. This expands the functionality of the assessment module by giving more details. Staff have only permissions to view the assessment details of students, whereas faculty can update the marks. The typical flow of this activity is that faculty/staff have to first log in before they can update/view the marks. On update, the subject assessment tables will be modified. Once all marks are updated, the total and grade are calculated by the software and the master table is updated to reflect the grade in the subject.

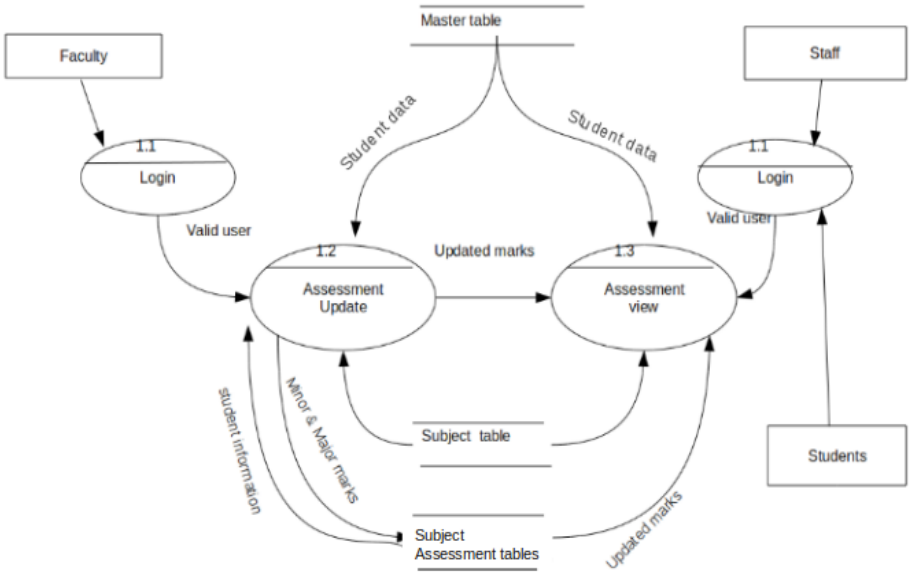

Figure 5.15: Level 1 DFD for update and view marks

Level 2 DFD: Level 2 DFD (Figure 5.16) of the assessment module will represent the one level depth of the level 1 DFD (as shown in figure 5.15). In this Faculty, can update the marks of the students by two ways. One is by using Update Online and the other one is Update Using File. If Faculty choose Update Online then faculty has to fill all the minor or major marks of the students by themselves and click on submit button and the data will gets updated in Course Assessment table of that course. This can be view in figure 5.13.

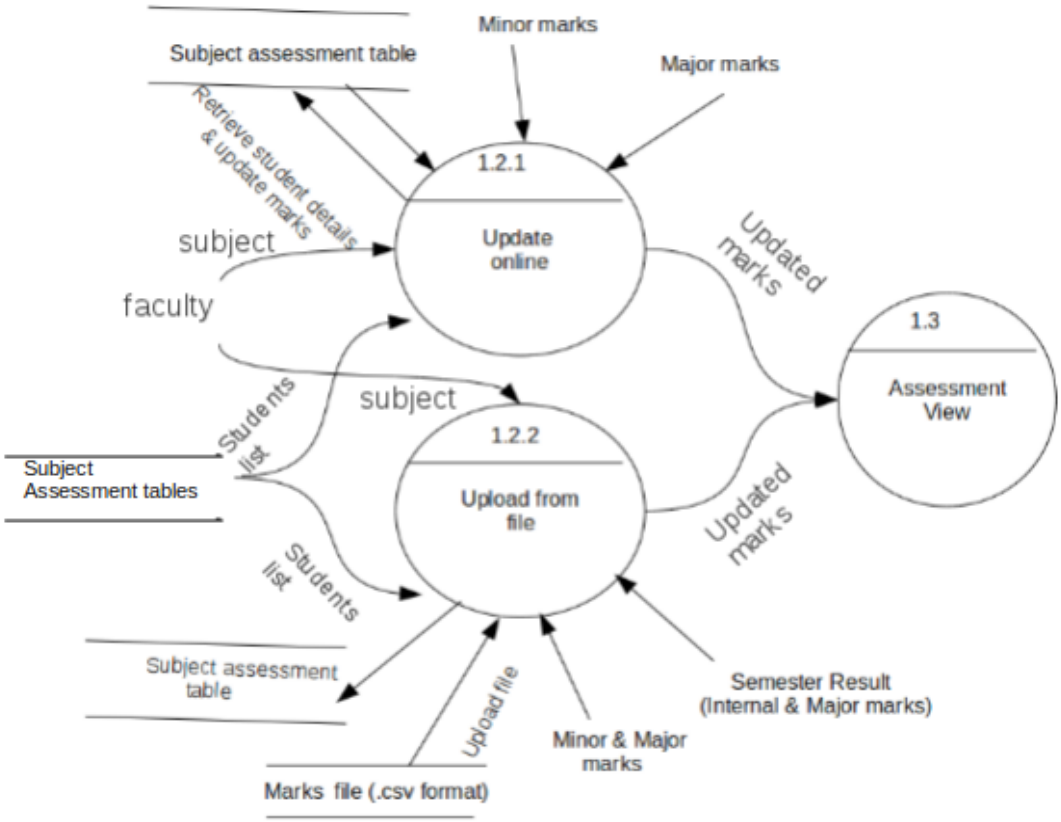

Figure 5.16: Level 2 DFD for update marks

Level 2 DFD (figure 5.17) of the assessment module will represent the one level depth of the level 1 DFD figure 5.15. For this the users are Staff, Faculty, Student, Admin. Staff can view the Assessment information using the student wise view, stream wise view, subject wise view. For student, only student wise view is available. For all the views data comes from course assessment tables of each courses figure 5.13 or from the master tables figure 5.3 and figure 5.4.

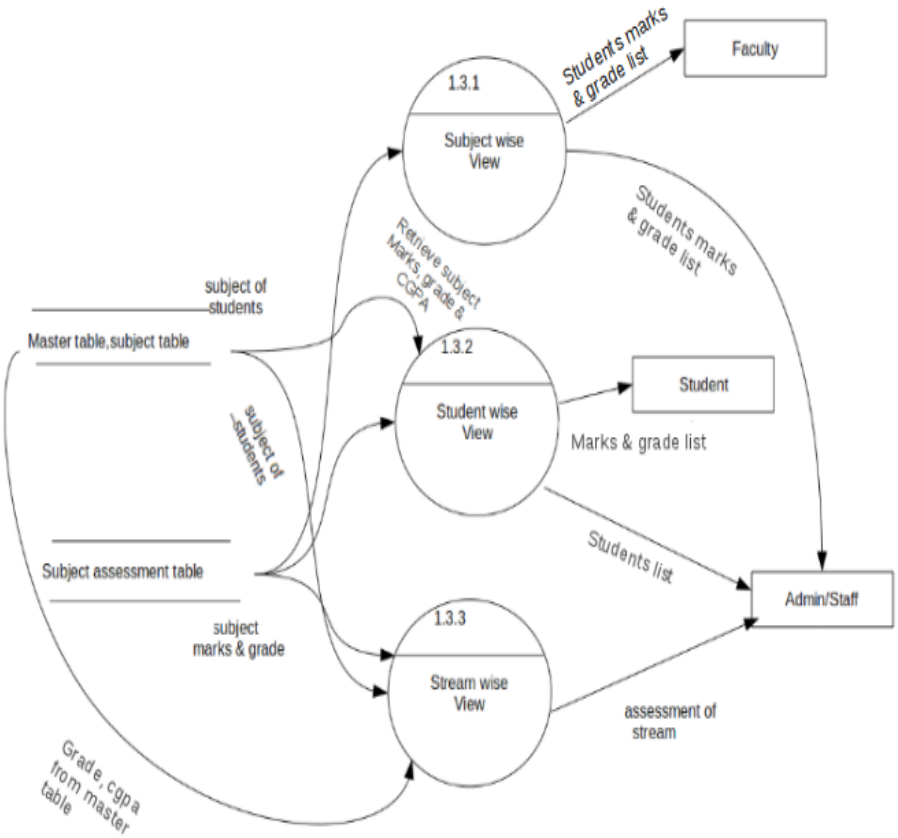

Figure 5.17: Level 2 DFD for staff view

Level 0 DFD shown in figure 5.18. This DFD of the course registration module will represent the whole module. The users of the system are Admin, Elective Registration Coordinator and staff. They can do several activities such as Subject-Faculty Registration Activity, Subject-Stream Registration Activity and Course Registration Activity. These Activities shown in figure 5.34, figure 5.35 and figure 5.36.

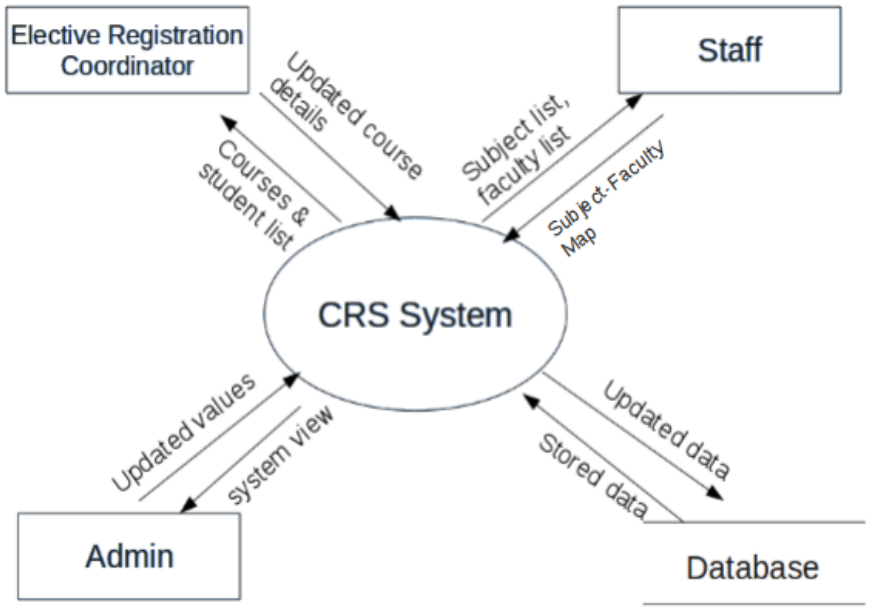

Figure 5.18: Level 0 DFD for course registration module

Level 1 DFD is represented by figure 5.19. This represent the flow of the Staff related activity in the system. Related to Course Registration Module, Staff has to First Login with valid username and password. Then register the Subjects for each faculty using Subject Faculty Registration Activity as shown in the figure 5.34. After submit flow transfer to Subject Stream Registration Activity represented in figure 5.35 . In Subject Stream Registration Activity, staff enters the Max limit of the students and pre-requisite for each course using Subject Stream Registration form shown by figure 5.35. This DFD uses the elective table(figure 5.12) and subject database table(figure 5.7).

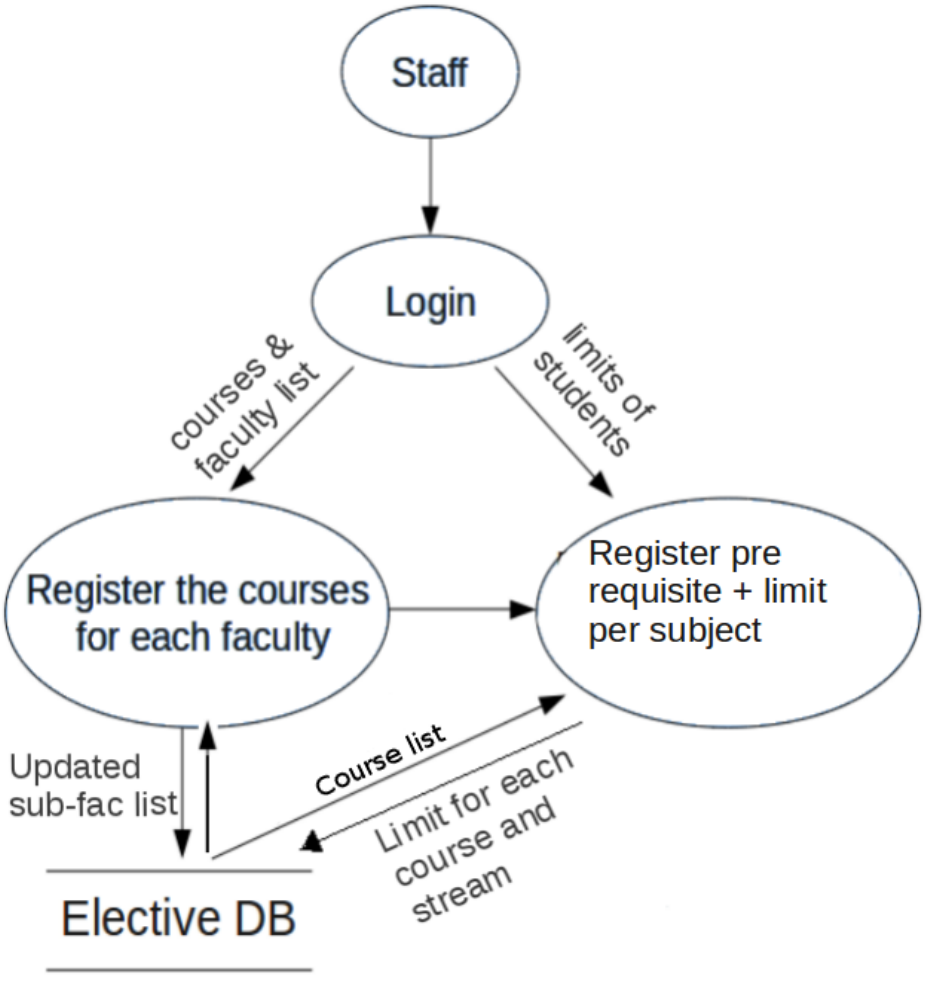

Figure 5.19: Level 1 DFD for staff to register the courses

Level 1 DFD can be viewed by figure 5.20. This represent the flow of the Elective Registration Coordinator related activity in the system. Related to Course Registration Module, Elective Registration Coordinator has to first Login with valid username and password. Register the Courses for each stream's student according to their given preferences. Elective Reg. Coordinator can also update the electives of the students.

This activity can be shown by figure 5.36. From this DFD master table figure 5.3 and coresponding Course Attendance table will get updated.

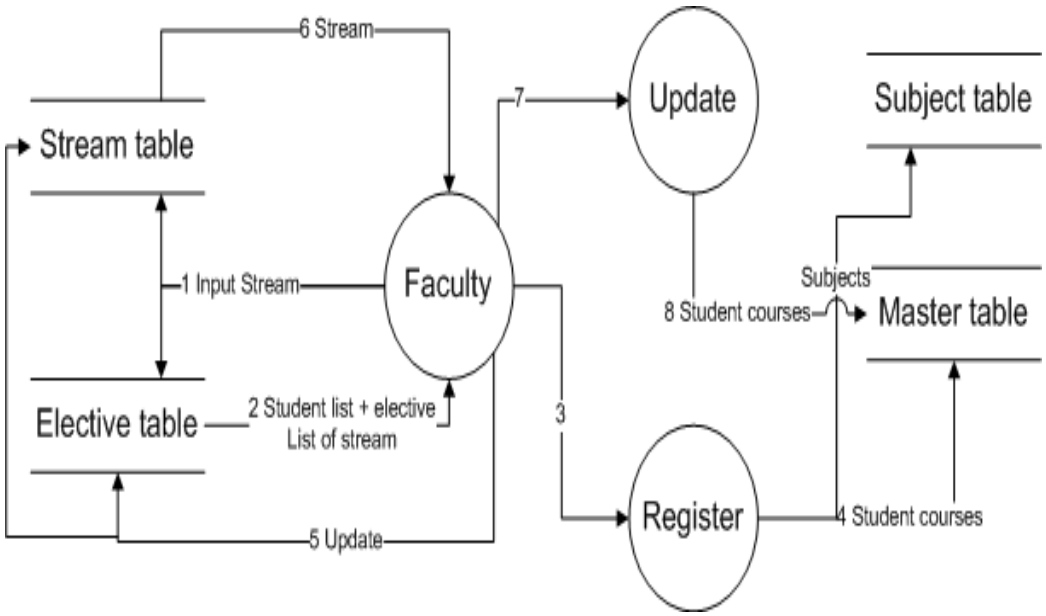

Figure 5.20: Level 1 DFD for course registration of students

5.3 Interface Design

We have designed an effective and user-centered interface for DSACR system. Users make a selection from the list of possibilities presented to them by the system. Typing effort is minnimized and context-dependent help is provided. The user's context is dependent on current menu selection.

5.3.1 Faculty Assessment Activities

There are two main activities for faculty as shown in the DFD 5.15 - updation of marks and viewing the marks. Updation of marks has two components: minor and major marks. The functionality is slightly different for the two of them on the assumption that major marks are entered last. In other words, the functionality is different depending on whether all the internal and major marks have been entered versus only part of the marks have been entered. When all marks have been entered, the total and grade are computed automatically. This triggers updation of the master table for each student with the grade obtained by the student in that subject. On the other hand, if all the marks have not been entered, then, it simply updates the subject table and the job is done. We describe this in more detail in the next sections.

Update Minor Marks

Purpose : To update minor marks of students.

Assumption : Faculty-Subject allocation and Student-Registration is done.

Pre-requisite : Faculty, student and subject information should be there in the database.

Tables used : subject_faculty table, Assessment_session_currentyear_subjectId table.

Updates table : Assessment_session_currentyear_subjectId table.

Procedure :

Login as faculty with valid username and password. This username and password will be matched with the LDAP server. If it does not match, it shows alert message. If it matches then the update page will open.

A pull-down menu containing the subjects being taught by the faculty in the current semester will be displayed in the left frame. This pull-down

menu is populated by querying the subject_faculty table with the faculty ID that has been used for log in. Once the faculty selects a course to update marks for, five radio buttons will be displayed - one each for the three minor marks, one for the major marks and one for the internal marks as shown in the figure 5.21. The latter is used to enter the internal marks directly instead of entering each of the minor marks separately. The faculty can update these marks online or can upload a *.csv* file. If the faculty wishes to update the marks online, the list of students in that subject will be retrieved from the Assessment_session_currentyear_subjectId table and displayed on a different page as shown in the figure 5.22. Once the marks are entered, they will be stored in this table. When the faculty selects a different set of marks to update, all the other fields that have been populated previously will also be shown in read-only text fields as shown in the figure 5.23.

If the faculty has already entered data for this particular minor/major, the previously entered marks along with the student list are retrieved and displayed. The faculty can make modifications to one or more students' marks and re-submit the marks. This will update the database to reflect the new marks. This may trigger re-calculation of the grade for the updated students and updating the master table once again to reflect the new grades.

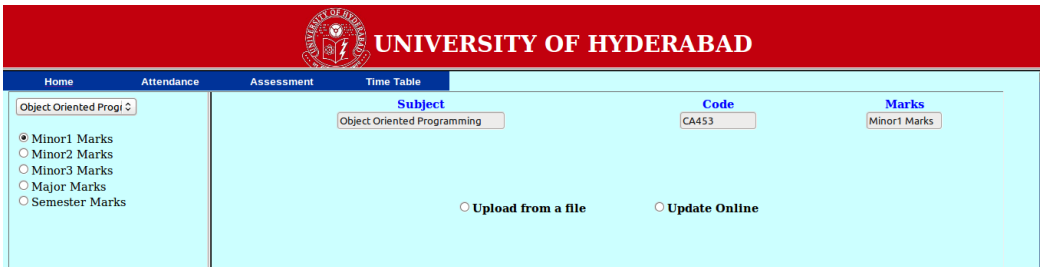

Figure 5.21: Screenshot of the update marks page, where faculty will select the subject, for which they want to update the marks, the type of exam for which they want to update and at last by which way, online or through file

Update Major Marks

Purpose : To update major marks and grade of the students.

Assumption : Faculty-Subject allocation and Student-Registration is done, internal marks are already calculated and saved in the database.

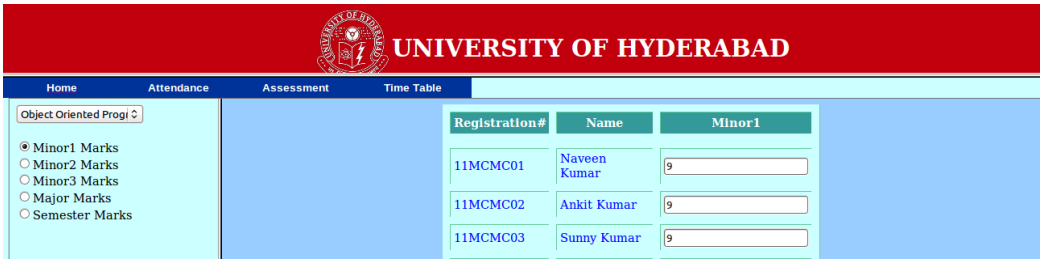

Figure 5.22: Screenshot of the page where Minor_1 marks are updated on-line. List of students is taken from Assessment_session_currentyear_subjectId table, and updations will also be saved in the same table. If marks are updated for the first time, it will show an empty text box, but if marks are already updated once, it will take the values from the database.

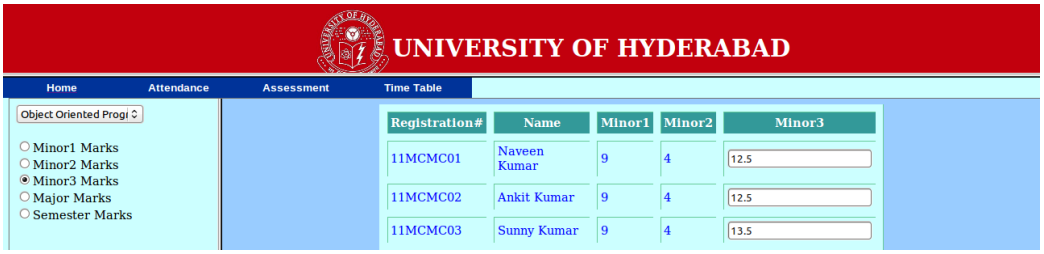

Figure 5.23: Screenshot of the page where Minor-3 marks are updated on-line. After the click on save button, it will automatically select best of the two minor marks, and save it in Internal-Marks field in the table. AB is to be given for absenties.

Pre-requisite : Faculty, student and subject information should be there in the database. Grade field for that subject should be marked as R in the master table.

Tables used : subject_faculty table, Assessment_session_currentyear_subjectId table, Master table, stream_grade_table.

Updates table : Assessment_session_currentyear_subjectId table, Master table.

Procedure :

If already login then do as follows otherwise login first as faculty with valid username and password as explained in previous activity. Now if the faculty selects Major marks to be updated on-line, then Internal Marks will

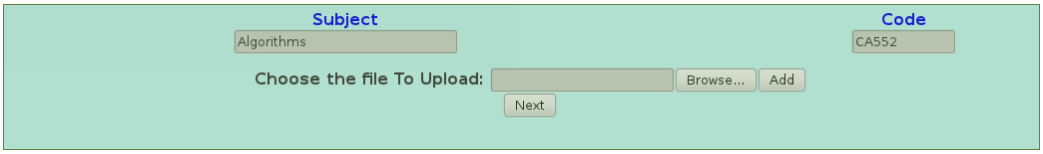

Figure 5.24: Screenshot of the page where marks are updated through file. Browse the marks file, and click on open button, then the path of the file will be displayed in the text field before the Browse button as shown in the next figure.

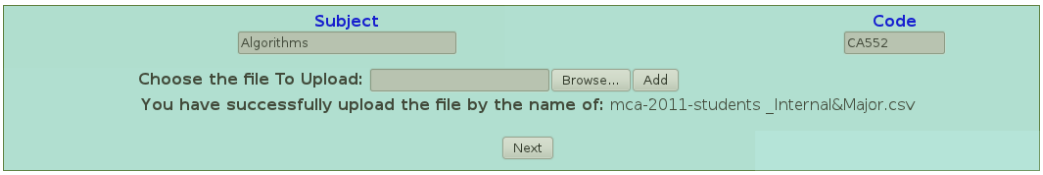

Figure 5.25: Screenshot of the page soon after the click on open button, click on add button, and the file will be uploaded successfully.

be shown in read only text field, and Major marks as input text field, as shown in the figure 5.27. Enter marks for all the students and then click on save button, it will calculate Total Marks, and grade based on grade-formula, taking the values from stream_grade_table, as shown in the figure 5.28. This will go and update grade both in master table and in subject table. This action will aso trigger an event, which updates the students cgpa in master table. Page displayed after updating major marks is shown in the figure 5.28

View Marks

Purpose : To view marks and grade of the students.

Assumption : Faculty-Subject allocation and Student-Registration is done, marks are updated and saved in the database.

Pre-requisite : Faculty, student, subject and assessment information should be there in the database.

Tables used : subject_faculty table, Assessment_session_currentyear_subjectId table.

Updates table : no table gets updated.

Figure 5.26: Screenshot of the page to update Internal and Major marks together

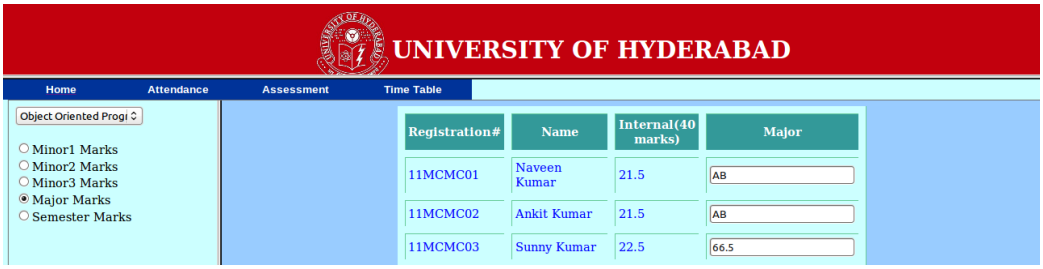

Figure 5.27: Screenshot of the page where Major marks are updated on-line. Grade field for that subject should be marked as R in the master table, before updating major marks. On click of save button, it will calculate Total Marks, and grade based on grade-formula, taking the values from stream_grade_table, as shown in the next figure.

Procedure :

If already login then do as follows otherwise login first as faculty with valid username and password as explained in previous activity. If faculty clicks on View link, then the data will be retrieved from the Assessment_session_currentyear_subjectId tables and the page displayed is shown in the figure 5.29.

5.3.2 Staff Assessment Activities

There are four main activities for staff as shown in the DFD 5.17. Staff can view assessment related activities in three different ways, i.e Streamwise, Subjectwise and Studentwise. We describe this in more detail in the next sections.

Streamwise View

Purpose : To view the marks of all the students in a particular stream.

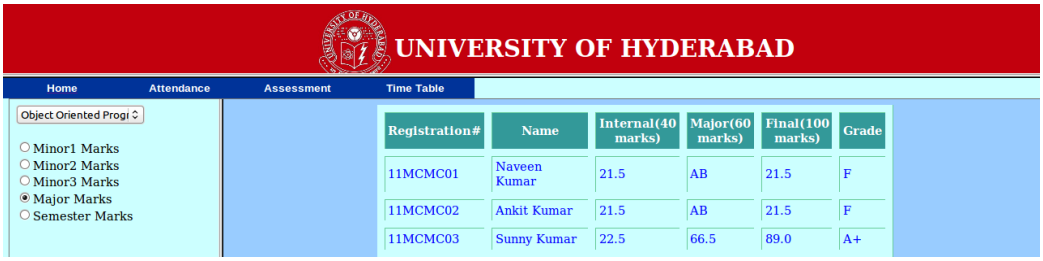

Figure 5.28: Screenshot of the page after updating major marks, grade will also be reflected in master table

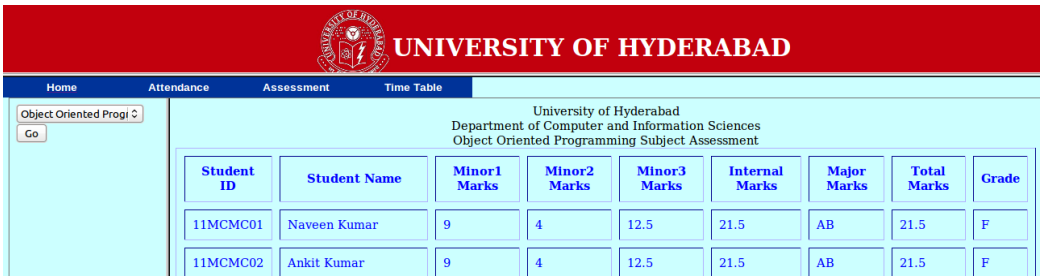

Figure 5.29: Screenshot of the page where marks and grade of all the students are displayed.

Assumption : Assessment is completed for that stream in all the current subjects.

Pre-requisite : Grade and CGPA should be updated in the master table.

Tables used : elective-table, stream-curriculam table.

Updates table : No table gets updated.

Procedure :

Login as staff with valid username and password. This username and password will be matched with the LDAP server, if it doesn't match, it shows alert message.If it matches then staff page will open. From the pull down menu, select Stream option, and click, it will display all the students grade and cgpa belonging in the selected stream, as shown in the figure 5.30.

Studentwise View

Purpose : To view the grade and CGPA of a particular student.

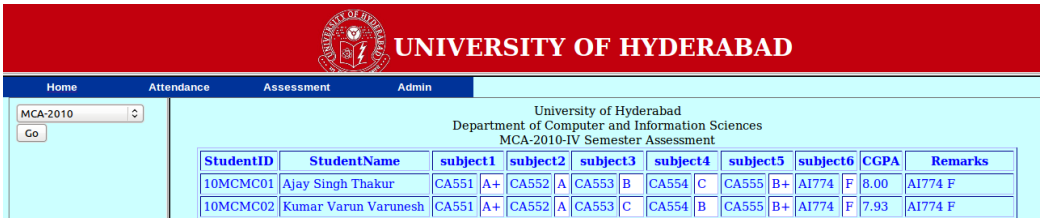

Figure 5.30: Screenshot of the streamwise view page for the staff, where current semester subjects, respective grades and cgpa of all the students in the selected stream will be displayed.

Assumption : Assessment is completed for all the current subjects of the student.

Pre-requisite : Grade and CGPA should be updated in the master table.

Tables used :Master table.

Updates table : No table gets updated.

Procedure :

If already login then do as follows otherwise login first as staff with valid username and password as explained in previous activity. This username and password will be matched with the LDAP server, if it doesn't match, it shows alert message. If it matches then staff page will open. From the pull down menu, select Student option, and click, it will display a form divided into two frames as shown in the figure 5.31.Enter the Student Registration Number in the text field in the left frame, and click on Go button, then the grade and cgpa of the that Student will be visible in the right frame.

Subjectwise View

Purpose : To view marks and grade of all the students in a particular subject.

Pre-requisite : Assessment_session_currentyear_subjectId table is updated

Tables used :Assessment_session_currentyear_subjectId table.

Updates table : No table gets updated.

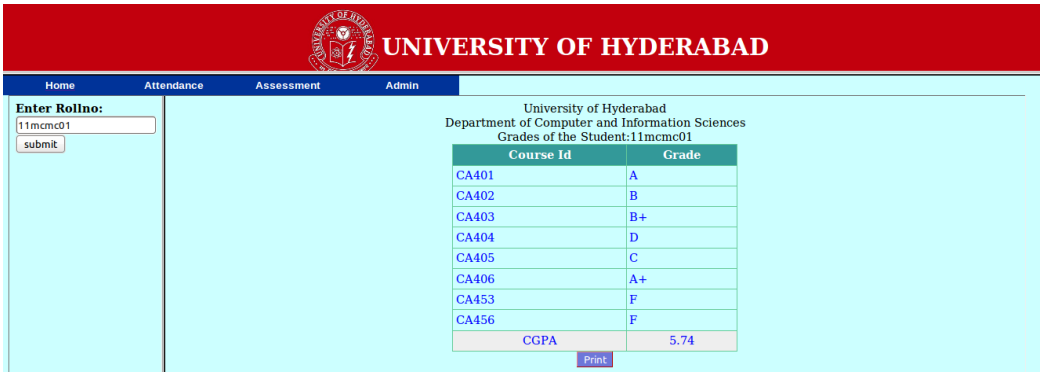

Figure 5.31: Screenshot of the studentwise view page for the staff, enter the Student Registration No in the text field in the left frame, and click on Go button, then the grade and cgpa of the that Student will be visible in the right frame.

Procedure :

If already login then do as follows otherwise login first as staff with valid username and password as explained in previous activity. This username and password will be matched with the LDAP server, if it doesn't match, it shows alert message.If it matches then staff page will open. From the pull down menu, select Subject option, and click, it will display one form divided into two frames as shown in the figure 5.32. Left frame will have a select button containing all the subjects that are running in the current semester, this list of subjects gets poulated from subject-data table. Select a subject and click on Go button, it will go and fetch the data from Assessment-session-currentyear-subjectId table, then all the students marks and grade will be displayed in the right frame.

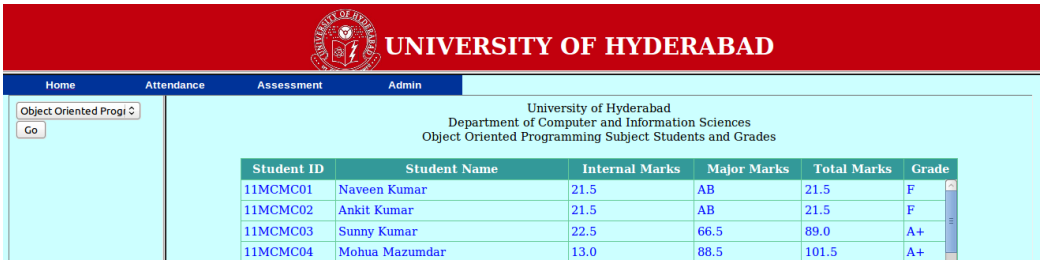

Figure 5.32: Screenshot of the subjectwise view page for the staff, left frame will have a select button containing all the subjects that are running in the current semester and a Go button, all students marks and grade will be displayed in the right frame

Update Grade Formula

Purpose : To update grade formula for a particular course of study.

Assumption : grade table should already exists.

Pre-requisite : Should be done in the starting of the session.

Tables used :stream-grade-table.

Updates table : stream-grade-table.

Procedure :

If already login then do as follows otherwise login first as staff with valid username and password as explained in previous activity. This username and password will be matched with the LDAP server, if it doesn't match, it shows alert message.If it matches then staff page will open. From the pull down menu, select Grade-Formula option, and click, it will display one form as shown in the figure 5.33.In this form all the grades will be dispayed as read only text field, and marks is an input text field, where staff will enter marks, depending on the range within which grade will fall. On the click on save button this page will go and update stream_grade_table. And this is the formula which is used in deciding grades, during major marks updation as shown in the figure 5.28.

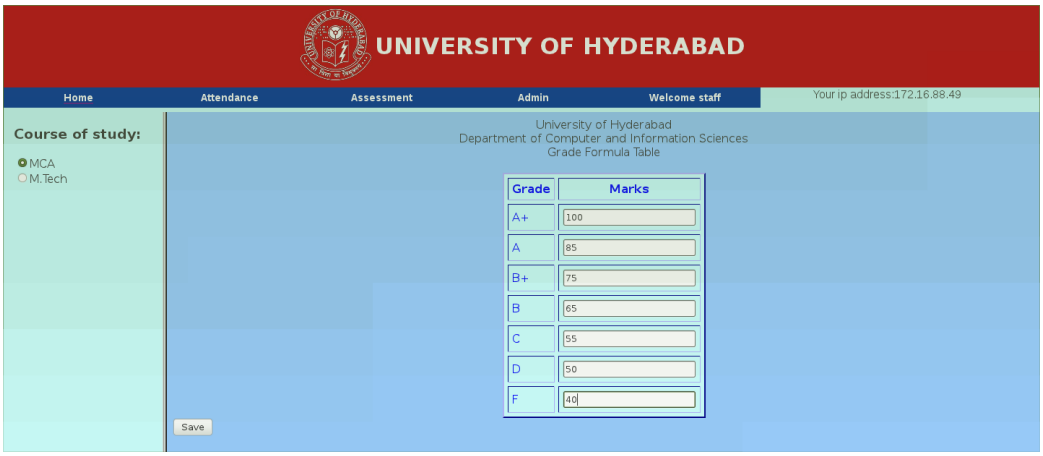

Figure 5.33: Screenshot of the grade formula page, here all the grades will be dispayed as read only, and marks is an input text field, where staff will enter marks, depending on the range within which grade will fall. On the click on save button this page will go and update stream_grade_table

5.3.3 Course Allocation Activities

In this module staff register the subjects for each faculty and each elective course fix the max number of students allowed for each stream and pre requisites of that elective course. These activities can be done by the following activities:

Subject-Faculty Registration Activity

Purpose : To register the subjects for each faculty in every semester.

Assumption : At max only 3 subjects can be register for each faculty in every semester.

Pre-requisite : All the subjects and faculty information should be there in database.

Tables used : faculty_data, subject_data.

Updates table : elective_table, subject_faculty

Procedure :

Login as staff with valid username and password. This username and password match with LDAP server if it not matches it shows alert message. If it matches then the welcome page will open. In that on the top menu, go to Admin option and in that select Subject Faculty Registration sub menu option, it will show one web page with one form like figure 5.34.

This form contains four columns namely Faculty Name, Subject 1, Subject 2, Subject 3. Faculty Names comes from faculty_data table. Subject 1, Subject 2, Subject 3 are in select box type and it contains all the course names from subject_data table as shown in figure 5.7. Along with every Subject column there is one checkbox which tells about elective courses. If the selected subject is elective then that checkbox will be disabled. If it is core for one stream and elective for other stream then click on checkbox. If it is core only then leave as it uncheck. For each faculty, Staff has to select at max 3 subject. After selecting the subjects, Staff has to click on submit button to register the subjects for faculty.

After submit, two tables elective_table (figure 5.12) and subject_faculty (figure 5.9) and for each course one table will be created. These course tables will be used in Attendance Module. The FacultyIds and corresponding selected SubjectIds will be stored in subject_faculty table. And all the elective

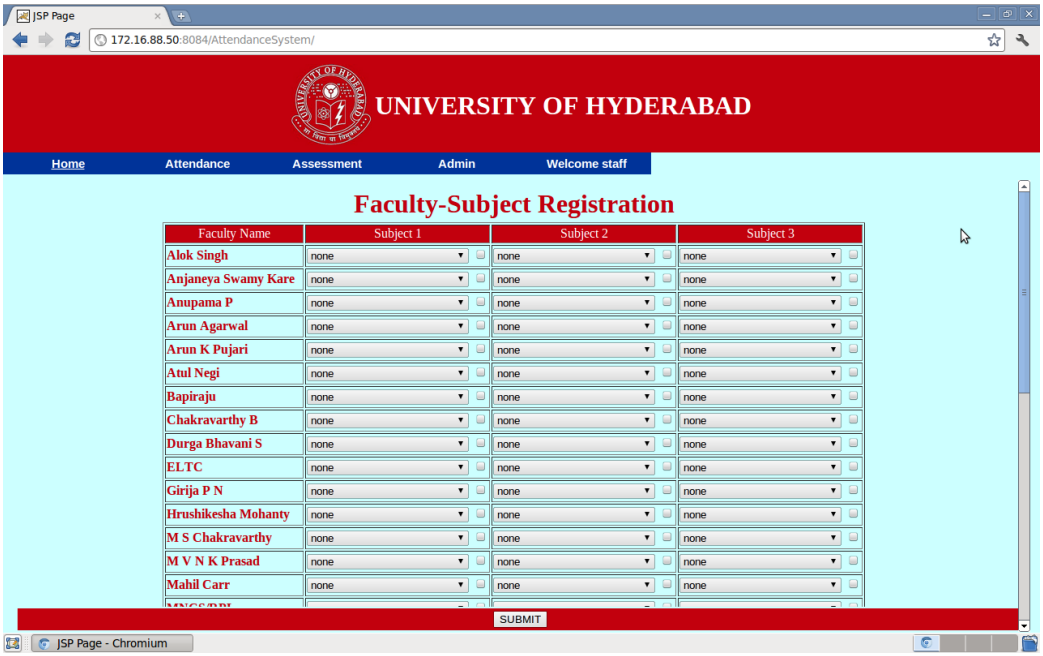

Figure 5.34: Subject Faculty Registraion Form

SubjectIds and elective cum core SubjectIds will be stored in elective_table. After this, flow will navigate to another page represented by figure 5.35. After this Elective-Stream Registration Activity has to be done.

Elective-Stream Registration Activity

Purpose : To register the electives for the streams in every semester.

Assumption : Here the assumption is that only two pre-requisites of subjects and grade for each elective are allowed.

Pre-requisite : All the elective subjects and streams information should be there in database.

Tables used : elective table.

Updates table : elective table.

Procedure :

If already login then do as follows otherwise login first as staff with valid username and password as explained in previous activity. Go to Admin->Elective-Stream Registration menu option and click. It will show one form. That form contains course name, all the streams, pre-req-1, pre-req-grade1, pre-req-2, pre-req-grade2. CouserName comes from elective-table shown in figure 5.12. For each course there is one textbox for every stream in that staff have to give the maximum number of students. Last four columns of pre-req and pre-req-grade are in selectbox type as shown in figure 5.35. For each elective, Staff has to select the pre-requisite for that elective (if any). After doing all this, Staff has to click on submit button. After submit, the max number of students for each stream and pre-requisite for each course will we updated in elective-table shown by figure 5.12.

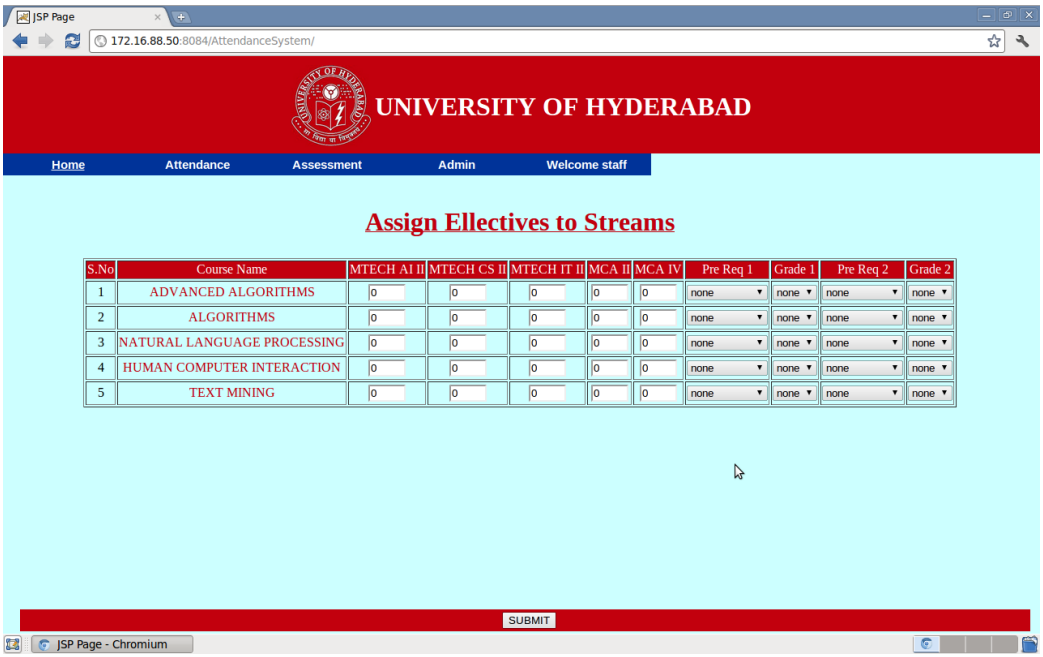

Figure 5.35: Course Stream Allocation Form

5.3.4 Course Registration Activities

Course Registration Activity

Purpose : To register the electives for the students for each stream.

Assumption : Here the assumption is that only elective courses can be registered for the students.

Pre-requisite : Here the pre-requisite is that before going to this step Subject-Faculty Registration and Elective-Stream Registration has already done.

Tables used : master table of each stream, elective table.

Updates table : master table of each stream and all attendance tables for selected electives.

Procedure :

Login as Elective-Registration-Coordinator. It will navigate to new page. In that go to Admin menu option and then select Course Registration menu option. Then this page will navigate to new page.

In this page two frames are there. In the left frame one selectbox is there to choose the stream. Whatever stream, faculty will choose coresponding to that stream table 5.3 and table 5.4 StudentId will be shown in new page. It will show one form. This form contains StudentId of selected stream and checkboxes for each elective (valid for that stream) as shown in figure 5.36. If any elective is already registered in previous semesters than that will be disabled. The elective checkbox will be enabled only if the student will satisfy the pre-requisite conditions. For each student of that stream, faculty has to click on the elective checkboxes to register that elective according to their given preferences. After each click on elective checkboxes it will check that the number of selected students can not exceed the max limit for that elective. After doing this for each student, staff has to click on submit button to register the courses.

After submit, two tables will be updated master table of coresponding stream and attendance course table for selected course for each student.

Example : Let for MCA-IV three electives are valid. Then for 10MCMC01 faculty clicks on the checkbox for CA505 and submit. Then in MCA-2010 master table, in the coresponding column of that elective will be updated with

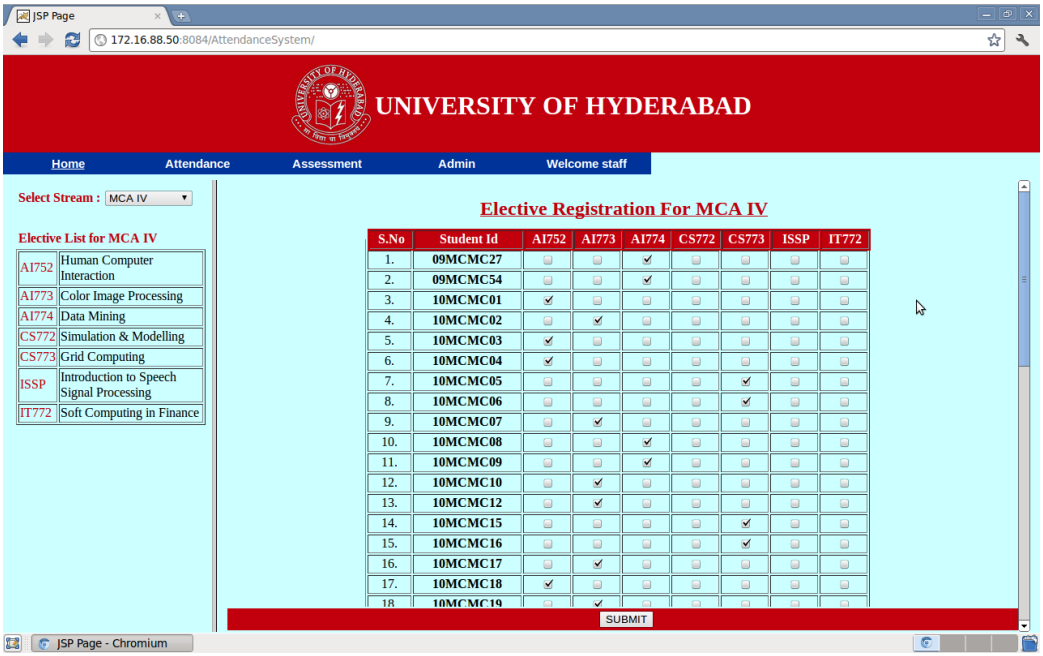

Figure 5.36: Course Registration Form

CA505, and the coresponding grade column will be updated with R(Registered). After this attendance table Winter-2012-CA505 will be updated with one record with that StudentId.

Chapter 6

Future Work

1. Provide GUI way to change user password.
2. Link to reset the forget password and policies to change password.
3. Strong password policies.
4. Student can give their elective preferences online.
5. Provide grade formula form for faculty in order to achieve relative grading.
6. Provide link for the student to appear for improvement and supplementary exams.
7. Same look and feel for each module.
8. Project and Ph.D. Students registration and assessment.
9. Database backup, archival and recovery

User Manual

1 Software Installation

1.1 Java installation

To run Tomcat as well as to run the DOAS software, you need Java Standard Edition (Java SE), also known as the JDK, that can be downloaded form here.

- Download the binary (say to Downloads directory) and install it as follows as root:

  su - (for OpenSuse/fedora/RHEL) or sudo -s (for Ubuntu/Linux Mint) or su root (for Debian).

- Now do the following:-

  # mkdir -p /usr/local

  # cd /usr/local

  # chmod 700 /yourhomedir/Downloads/jdk-version-architecture.bin

  # /yourhomedir/Downloads/jdk-version-architecture.bin

- export PATH variable as:-

  # export $JAVA\_HOME = /usr/java/jdk1.7.0_01$

  # export $PATH =\$JAVA\_HOME/bin$:\$PATH

- **Check the version**

  # which java

  $/usr/local/jdk1.7.0\_01/bin/java$

  # java –version (should give something like following)

  java version "1.7.0_01"

  Java(TM) SE Runtime Environment ($build1.7.0\_01 - b08$)

  Java HotSpot(TM) Server VM (build 21.1-b02, mixed mode)

1.2 Apache Tomcat server installation

To install Apache Tomcat Server Download the Tomcat server binary.
Installing Tomcat from a binary release (tar file) requires manual creation of the Tomcat user account. This is not necessary if you install the Tomcat RPM package on a Linux system that supports RPMs or You install with Netbeans IDE.

- let us install it manually from downloaded tar file.

- first create tomcat user as follows (with nologin):

  # groupadd tomcat

  # useradd -g tomcat -s /usr/sbin/nologin tomcat

- Extract the tar file to /srv and changed the ownership of all files and directories to tomcat:

  # cd /srv

  # tar zxvf /yourhomedir/Downloads/apache-tomcat-7.0.14.tar.gz

  # chown -R tomcat.tomcat /srv/apache-tomcat-7.0.14

- To get the Tomcat version of the newly installed Tomcat, run:

  # /srv/apache-tomcat-7.0.14/bin/version.sh

- To start tomcat, run:

  # /srv/apache-tomcat-7.0.14/bin/startup.sh

- To stop tomcat, run:

  # /srv/apache-tomcat-7.0.14/bin/shutdown.sh

1.3 MySQL community server installation

- For Debian type system(like Debian, Ubuntu, LinuxMint etc)

  Download DEB package of MySQL-community-server.

  $ cd /yourhomedir/Downloads

  $ chmod +x mysql-community-server-5.x.deb

  $ sudo dpkg -i mysql-community-server-5.x.deb

- For RPM supporting systems (like OpenSuse, Fedora etc)

  Download RPM package of MySQL-community-server.

  $ cd /yourhomedir/Downloads

  $ chmod +x mysql-community-server-5.x.rpm

  su -

  # rpm -ivh mysql-community-server-5.x.rpm

- For OpenSuse (gui based)

  go to yast2-> software management.

  search mysql-community-server and mark for installation.

  apply the changes, mysql-community server will be installed on your system.

1.4 LDAP server installation

- Prerequisite database software is Berkeley DB database that can be downloaded from here.
- Download OpenLDAP here.
- Unpack the distribution and change directory:

  $ tar xvfz openldap-VERSION.tgz

  $ cd openldap-VERSION
- Configure the build system:

  $ ./configure

  If all goes well, the configure script will automatically detect the appropriate settings. If the configure script fails, you should read the config.log file that it generated to see what it was trying to do and exactly what failed. You may need to specify additional options and/or variables to obtain desired results, depending on your operating system.
  The Platform Hints section of the FAQ provides help for operating system related problems.
- Build dependencies:

  $ make depend
- Build the system:

  $ make

  If all goes well, the system will build as configured. If not, return to Configure step after reviewing the configuration settings. You may want to consult the Platform Hints subsection of the FAQ if you have not done so already.

- Test the standalone system:

  This step requires the standalone LDAP server, slapd(8), with HDB and/or BDB support.

  $ make test

  If all goes well, the system has been built as configured.
  If not, return to configure step after reviewing your configuration settings. You may want to consult the Installation section of the FAQ if you have not done so already.

- Install the software. You may need to become the super-user (e.g. root) to do this :

  $ su root -c 'make install'

- That's it. Enjoy!

- Special Case: **For OpenSuse users(gui based method)**:-
  To install OpenLDAP on OpenSuse do as follows

- go to yast2->software management

- search yast2-ldap-server and mark for installation.

- That's it.

- For more on ldap configuration and user management please consult my another manual:
  Manual For Administrators

2 Users

2.1 Admin

2.2 Staff

2.3 Faculty

2.4 Student

3 Admin

As you login as the Admin, you can see the following menu items

- Attendance
- Assessment
- Initialization
- Yearly Activities

Attendance and Assessment will be explained in the Staff Module.

3.1 Initialization

There are three sub menus under Initialization menu.We can see them on moving the mouse pointer on Initialization menu.

- Faculty Database
- Load Curriculum File
- Elective Subject List

3.1.1 Faculty Database

1. On mouse Click on the *Faculty List* link, you will see a new form below the menu bar containing a Browse button and Upload button.
2. Click on Browse button
3. Choose the Faculty List file(file.csv) and click on open
    - Note: The file should be with .csv extension for ex. filename.csv etc.

The file should contain only two columns(Faculty User Id, Faculty Name) and columns should be separated by comma(,).

An example of the file is shown below.

Faculty User Id,Faculty Name
apcs,Anupama P //first row
sdbcs,S Durga Bhavani
tsrcs,T Sobha Rani
————————

4. Click on Upload button

5. If the file uploaded successfully then it will show a Success page else Error page.

3.1.2 Load Curriculum File

1. On mouse Click on the *Faculty List* link, you will see a new form below the menu bar containing a Select button,a Browse button and Upload button.

2. Select a stream in the select button

3. Browse the Curriculum file of the stream that you have selected in the select button by using Browse button.

   - Note: The file should be with .csv extension for eg. file.csv,file1.csv etc.
   - Curriculum file should contain all the core subject list of the selected stream.
   - First Row of the file should contain #c,#l,#p,#e in the order
     - #c-number of core subjects
     - #l-number of labs
     - #p-number of projects
     - #e-number of electives
   - Second Row onwards, #c rows should be filled with cores subjects, next #l rows with lab subjects, #p rows with project information.
   - The columns should be in below given order

The file should contain only four columns(Subject Id, Subject Name, Credits, Semester) and columns should be separated by comma(,) and columns optionally enclosed by double quotes(" ").

An example of the file is shown below.

Subject Id,Subject Name,Credits,Semester
4,3,1,6 //first row
"CS703","Operating Systems",4,"I"
"CS704","Algorithms",4,"I"
"CS705","Computer Architecture",3,"I"

4. Click on Upload button

5. If the file uploaded successfully then it will show a Success page else Error page.

3.1.3 Elective Subject List

1. On mouse Click on the *Elective Subject List* link, you will see a new form below menu bar containing three columns Subject Id, Subject Name, Credits

2. It has five rows, one row for each subject information. Atleast one and atmost five can be updated at a time.

   - Note:Only elective subjects need to be updated through this form

3. Once you fill the data in the form, click on submit button

4. If the data updated successfully then it will show a Success page else Error page.

3.1.4 Add Faculty

1. On mouse Click on the *Add Faculty* link, you will see a form below containing columns Faculty User Id,Faculty Name

2. It has five rows, one row for each Faculty information.Atleast one and atmost five can be updated at a time

3. Once you fill the data in the form,click on submit button

4. If the data updated successfully then it will show a Success page else Error page.

3.2 Yearly Activities

3.2.1 New Batch

1. On mouse Click on the *New Batch* link, you will see a form below containing fields, one select button for selecting the stream, another select button for selecting the year and a browse and Upload button to upload the Student List of selected stream and batch.

2. Select Stream and Year

3. Browse the Curriculum file of the stream that you have selected in the select button by using Browse button.

   - Note: The file should be with .csv extension for eg. stream.csv, stream1.csv etc.

   The file should contain only two columns(Student Registration No. , Student Name) and columns should be separated by comma(,) and columns optionally enclosed by double quotes(“”).

   An example of the file is shown below.

Registration No. , Student Name
”11mcmt01”,”student1”
”11mcmt02”,”student2”
”11mcmt03”,”student3”
————————

4. Click on Upload button

5. If the file uploaded successfully then it will show a Success page else Error page.

3.2.2 Re-Admission

3.3 Semester Activities

3.3.1 Faculty Subject Allocation

1. For this, staff have to login and select Admin->Online Faculty-Subject Registration menu option. It will show a form. where:

   (a) the staff can choose for each faculty a maximum of 3 subjects using **pull-down menu**.

(b) For each subject there is a **check box** which allows one to specify whether the selected subject is an elective subject or not.

- If the subject is an elective, then this **check box** is automatically selected.
- If the subject is a core but being opened up as an elective then this **check box** is selected to indicate the same.
- If the subject is only a core, then, this **check box** should not be selected.

2. After filling this for each faculty, submit that form using **submit button**.

3.3.2 Course Details

The course details page is automatically displayed after *Faculty-Subject Registration* in previous section is successfully done. If this is not immediately done, then, the staff can login and navigate to this page using the Admin->Course Details menu option. This will open a form in which we need to fill the following:

- The **maximum number of students per stream** allowed in the course.
- If there are any **pre-requisites** in terms of courses done and grade obtained in a particular course, these details have to be filled in.

Once both these details are filled for each elective, submit that form using **submit button**.

3.3.3 Course Registration

1. For this, staff have to login and select Admin->Course Registration menu option. Then choose the stream for which course registration is being done.
2. This shows all the students corresponding to that stream as rows and possible electives for that stream as columns. Each subject column has a **check box**.
3. Select the **check boxes** according to students preferences and click on **submit button** and it shows the list of the courses chosen by each student.

3.4 Attendance

3.5 Assessment

4 Staff

4.1 Attendance

4.1.1 Update

1. On mouse Click on the *Update* link, you will see left and right frames below the menu bar.

2. Left Frame will have a select button containing all the subjects that are running in the current semester and a Go button

3. Right Frame consists of many form fields explained each one below.

   (a)

Attribute	Description	Type
Year	Displays current year	read only
From	Start date of Attendance of current month	Select Button
To	End date of Attendance of current month	Select Button
Semester	Current semester	read only
Subject Name	Name of Selected Subject in left frame	read only
Code	Subject code of Selected Subject in the frame	read only
Previous Total	Previous Months Cumulative Attendance	read only
Current Total	Current Month Total Attendance	text field

   (b) Two Radio Buttons,one for upload Attendance from a file and other for update Attendance online

4. Use the select buttons from and to for selecting the duration of the Month

5. Use the Current Total Text Field to enter total attendance of the selected Month.

6. Click on one Radio Button to enter the Attendance.

7. If you click on radio button, Upload from a file then

   (a) It will display the page containing all the above form fields(read only) and one more form field Total(readonly).

   (b) The page also contains the Browse button and Add Button.

(c) On clicking on the browse button, you select the Attendance File and click on Open button. Then the path of the file will be displayed in the text field before the Browse button.

(d) Click on the Add button.

(e) It will display a message "File uploaded successfully" and a View Data button, if the file is uploaded successfully else no message will be displayed(again you need to upload the file)

- Note: The file should be with .csv extension for eg. attendancelist.csv,attendancelist1.csv etc.

The file should contain only three columns(Student Registration No. , Student Name, Attendance of the Current Month) and columns should be separated by comma(,) and columns optionally enclosed by double quotes("").

An example of the file is shown below.

Registration No. , Student Name, Attendance
"11mcmt01","student1",12
"11mcmt02","student2",11
"11mcmt03","student3",10
————————

(f) On click on View data button, you can view the Updated Attendance and Percentage.

8. If you click on the Radio button, Update Online then

(a) It will display the page containing all the above form fields(read only) and one more form field Total(readonly).

(b) The page also contains the Registration No., Student Name, Current Attendance of the subject that you selected in the left frame.

(c) Registration No., Student Name are readonly fields, and Current Attendance is editable text field where you need to give the respective student attendance of that month.

- Note: Current Attendance field of all students is already filled Current Total Attendance. You can simply change the fields where there is less Attendance for the student.

(d) Once you fill the data, click on Update Database button.

(e) After clicking on Update Database button, you can view the Updated Attendance and Percentage.

4.1.2 Student

1. On mouse Click on the *Student* link, you will see left and right frames below the menu bar.
2. In the left frame you can see an editable text field and a Go button.
3. Enter the Student Registration No. in the text field and ENTER.
4. Then the Attendance of the that Student can be visible in the right frame.

4.1.3 Subject

1. On mouse Click on the *Subject* link, you will see left and right frames below the menu bar.
2. Left Frame will have a select button containing all the subjects that are running in the current semester and a Go button
3. Select a subject from select menu and click on Go button.
4. Then the Attendance of all the Students studying in that subject can be visible in the right frame.

4.1.4 Stream

1. On mouse Click on the *Stream* link, you will see left and right frames below the menu bar.
2. Left Frame will have a select button containing all the steams that are running in the current semester and a Go button
3. Select a Stream from select menu and click on Go button.
4. Then you can view the Cumulative Attendance of every subject of every student who are in that stream, overall percentage and remarks in the right frame.

4.1.5 Summary View

1. On mouse Click on the *Summary* link, you will see left and right frames below the menu bar.
2. Left Frame will have a select button containing all the steams that are running in the current semester and a Go button

3. Select a Stream from select menu and click on Go button.

4. Then you can view the Overall Attendance of every Student who are in that stream and remarks in the right frame

4.2 Assessment

4.2.1 Student

1. On mouse Click on the *Student* link, you will see left and right frames below the menu bar.

2. In the left frame you can see an editable text field and a Go button.

3. Enter the Student Registration No. in the text field and ENTER.

4. Then the grade and cgpa of the that Student can be visible in the right frame.

4.2.2 Subject

1. On mouse Click on the *Subject* link, you will see left and right frames below the menu bar.

2. Left Frame will have a select button containing all the subjects that are running in the current semester and a Go button

3. Select a subject from select menu and click on Go button.

4. Then the grade of all the Students studying in that subject can be visible in the right frame.

4.2.3 Stream

1. On mouse Click on the *Stream* link, you will see left and right frames below the menu bar.

2. Left Frame will have a select button containing all the steams that are running in the current semester and a Go button

3. Select a Stream from select menu and click on Go button.

4. Then you can view the grade of every subject of every student who are in that stream, cgpa and remarks in the right frame.

4.2.4 Grade-Formula

1. On mouse click of the *Grade-Formula* link, you will see left and right frames below the menu bar.

2. Left Frame will have radio buttons containing all course of study, running in the department.

3. Select a course of study.

4. Then you can view a table in the right frame, which display all the grades, and you have to fill the respective marks, within which the grade will fall.

5. Then finally click on save button.

5 Faculty

5.1 Attendance

5.1.1 Update

1. On mouse Click on the *Update* link, you will see left and right frames below the menu bar.

2. Left Frame will have a select button containing the subjects, which has been taught by the faculty, that are running in the current semester and a Go button

3. Right Frame consists of many form fields explained each one below.

(a)

Attribute	Description	Type
Year	Displays current year	read only
From	Start date of Attendance of current month	Select Button
To	End date of Attendance of current month	Select Button
Semester	Current semester	read only
Subject Name	Name of Selected Subject in left frame	read only
Code	Subject code of Selected Subject in left frame	read only
Previous Total	Prev Months Cumulative Attendance	read only
Current Total	Current Month Total Attendance	text field

(b) Two Radio Buttons,one for upload Attendance from a file and other for update Attendance online

4. Use the select buttons from and to for selecting the duration of the Month

5. Use the Current Total Text Field to enter total attendance of the selected Month.

6. Click on one Radio Button to enter the Attendance.

7. If you click on radio button, Upload from a file then

   (a) It will display the page containing all the above form fields(read only) and one more form field Total(readonly).

   (b) The page also contains the Browse button and Add Button.

   (c) On clicking on the browse button, you select the Attendance File and click on Open button.Then the path of the file will be displayed in the text field before the Browse button.

   (d) Click on the Add button.

   (e) It will display a message "File uploaded successfully" and a View Data button, if the file is uploaded successfully else no message will be displayed(again you need to upload the file)

   - Note: The file should be with .csv extension for eg. attendancelist.csv,attendancelist1.csv etc.

   The file should contain only three columns(Student Registration No. , Student Name, Attendance of the Current Month) and columns should be separated by comma(,) and columns optionally enclosed by double quotes(“”).

   An example of the file is shown below.

Registration No. , Student Name, Attendance
”11mcmt01”,”student1”,12
”11mcmt02”,”student2”,11
”11mcmt03”,”student3”,10
———————

   (f) On click on View data button, you can view the Updated Attendance and Percentage.

8. If you click on the Radio button, Update Online then

   (a) It will display the page containing all the above form fields(read only) and one more form field Total(readonly).

(b) The page also contains the Registration No., Student Name, Current Attendance. of the subject that you selected in the left frame.

(c) Registration No., Student Name are readonly fields and Current Attendance is editable text field where you need to give the respective student attendance of that month.

- Note: Current Attendance field of all students is already filled with Current Total Attendance. You can simply change the fields where there is less Attendance for the student.

(d) Once you fill the data, click on Update Database button.

(e) After clicking on Update Database button, you can view the Updated Attendance and Percentage.

5.1.2 View

1. On mouse Click on the *Subject* link, you will see left and right frames below the menu bar.

2. Left Frame will have a select button containing the subjects, which has been taught by the faculty, that are running in the current semester and a Go button

3. Select a subject from select menu and click on Go button.

4. Then the Attendance of all the Students studying in that subject can be visible in the right frame.

5.1.3 Summary View

1. On mouse Click on the *Subject* link, you will see left and right frames below the menu bar.

2. Left Frame will have a select button containing the subjects, which has been taught by the faculty, that are running in the current semester and a Go button

3. Select a subject from select menu and click on Go button.

4. It will display Student Id, Student Name, Current Month Attendance, Overall percentage of that subject in the right frame.

5.2 Assessment

5.2.1 Update Marks

1. On mouse Click on the *Update Marks* link, you will see left and right frames below the menu bar.

2. Left Frame will have a select button containing all the subjects,which has been taught by the faculty, that are running in the current semester and five radio buttons to enter internal marks or major marks.

3. Right Frame consists of many form fields explained each one below.

   (a)

Attribute	Description	Type
Subject	Name of selected subject in left frame	read only
Code	Subject code of selected subject in left frame	read only
Marks	Type of marks selected by radio button in left frame	read only

   (b) Two Radio Buttons, will be displayed, in case you selected Minor1, Minor2, Minor3 or Major marks, one for upload marks from a file and other for update marks online, and one radio button upload from a file will be displayed, in case you selected Semester marks, where faculty can directly upload internal and major marks together.

4. Click on one Radio Button to enter the marks.

5. If you click on radio button, Upload from a file then

   (a) It will display the page containing all the above form fields(read only).

   (b) The page also contains the Browse button and Add Button.

   (c) On clicking on the browse button, you select the marks file and click on Open button. Then the path of the file will be displayed in the text field before the Browse button.

   (d) Click on the Add button.

   (e) It will show the updated data, if the file is uploaded successfully, else will show an error message(again you need to upload the file)

   - Note: The file should be with .csv extension for eg. minor1.csv,major.csv etc.

If Minor1, Minor2, Minor3 or Major marks radio button is selected, then the file should contain only three columns(Student Registration No. , Student Name, Marks) but if, Semester marks is selected, then the file should contain four columns(Student Registration No. , Student Name, Internal marks, Major marks) and columns should be separated by comma(,) and columns optionally enclosed by double quotes(“”).

Example of the files are shown below.

Registration No. , Student Name, marks
”11mcmc01”,”student1”,12
”11mcmc02”,”student2”,11
”11mcmc03”,”student3”,10
————————

Registration No. , Student Name, Internal marks, Major marks
”11mcmc01”,”student1”,12,45
”11mcmc02”,”student2”,11,55
”11mcmc03”,”student3”,10,40
————————

(f) On click on View data button, you can view the Updated Marks and grade.

6. If you click on the Radio button, Update Online then:

   (a) It will display the page containing all the above form fields(read only).

   (b) The page also contains the Registration No., Student Name, Current marks of the subject that you selected in the left frame.

   (c) Registration No., Student Name are readonly fields, and Current marks is editable text field where you need to give the respective student marks of that exam.

   - Note: For the first time marks field for all the students will be empty, you can fill the respective marks of every student, but if you click on the same radio button again, you will see field of all students is already filled with the marks you updated last time. You can simply change the fields where there is any modification in the marks of the student.

   (d) Once you fill the data, click on save button.

   (e) After clicking on save button, you can view the Updated marks.

6 Student

6.1 Attendance

6.1.1 View

1. On mouse Click on the *View* link, Attendance of all the subjects of that Student can be visible below the menu bar.

6.2 Assessment

6.2.1 View

1. On mouse Click on the *View* link, grade of all the subjects of the current semester and cgpa, will be visible for that Student below the menu bar.

6.2.2 Detail View

1. On mouse Click on the *Detail View* link, grade of all the subjects upto that semester and cgpa, will be visible for that Student below the menu bar.

Manual for Administrators

1 OpenLDAP Configuration

Once the OpenLDAP software has been built and installed, we are ready to configure it. The configuration is primarily accomplished through the slapd.conf file, normally found in the /usr/local/etc/openldap or /etc/openldap directory.

- The slapd.conf file consists of three types of configuration information: global, backend specific and database specific. Global information is specified first, followed by information associated with a particular backend type, which is then followed by information associated with a particular database instance. Global directives can be overridden in backend and/or database directives, and backend directives can be overridden by database directives.
- Blank lines and comment lines beginning with a # character are ignored.
- *Note:* If a line begins with white space, it is considered a continuation of the previous line (even if the previous line is a comment).
- Please take care of extra spaces and newlines. Extra white spaces or new line may lend you in trouble.

1. Global directives:

   1.1 access to <what> [by <who> <accesslevel> <control>]+
   This directive grants access (specified by <accesslevel>) to a set of entries and/or attributes (specified by <what>) by one or more requesters (specified by <who>).
   A snapshot of slapd.conf:

```
#               Allow self write access to user password
#               Allow anonymous users to authenticate
#               Allow read access to everything else
#       Directives needed to implement policy:
access to dn.base=""
        by * read
access to dn.base="cn=Subschema"
        by * read
access to attrs=userPassword,userPKCS12
        by self write
        by * auth
access to attrs=shadowLastChange
        by self write
        by * read
access to *
        by * read
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn.  (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!
#######################################################################
```

1.2 attributetype <Attribute Type Description>
This directive defines an attribute type. We can specify new attributes also in schema file.

1.3 idletimeout <integer>
Specify the number of seconds to wait before forcibly closing an idle client connection. An idletimeout of 0, the default, disables this feature.

1.4 include <filename>
This directive specifies that slapd should read additional configuration information from the given file before continuing with the next line of the current file.
A snapshot of slapd.conf:

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/rfc2307bis.schema
include         /etc/openldap/schema/yast.schema
#include                /etc/openldap/schema/nis.schema
# Define global ACLs to disable default read access.
# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org
pidfile         /var/run/slapd/slapd.pid
argsfile        /var/run/slapd/slapd.args
# Load dynamic backend modules:
# modulepath    /usr/lib/openldap/modules
# moduleload    back_bdb.la
# moduleload    back_hdb.la
# moduleload    back_ldap.la
# Sample security restrictions
```

1.5 loglevel <integer>
This directive specifies the level at which debugging statements and operation statistics should be syslogged.

2. Backend Directives:

2.1 backend <type>
This directive marks the beginning of a backend declaration. <type> should be one of the supported backend types.

3. Database Directives:

3.1 database <type>
This directive marks the beginning of a database instance declaration. <type> should be one of the supported backend types.

3.2 rootdn <DN>
This directive specifies the DN that is not subject to access control or administrative limit restrictions for operations on

this database. The DN need not refer to an entry in this database or even in the directory.

3.3 rootpw <password>
This directive can be used to specifies a password for the DN for the rootdn.
Note: Use slappasswd to generate password.

3.4 suffix <dn suffix>
This directive specifies the DN suffix of queries that will be passed to this backend database. Multiple suffix lines can be given, and at least one is required for each database definition. A snapshot of slapd.conf:

```
#######################################################################
# BDB database definitions
#######################################################################
database        bdb
suffix          "dc=dcisonline,dc=uohyd,dc=ernet,dc=in"
checkpoint      1024    5
cachesize       10000
rootdn          "cn=root,dc=dcisonline,dc=uohyd,dc=ernet,dc=in"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          {SSHA}[redacted]
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory       /var/lib/ldap
# Indices to maintain
index   objectClass     eq
```

That's it! We are done with the LDAP configuration. for more info see the *man* pages.

2 LDAP user account creation

1. Initializing the base system

   To initialize the system we add initial.ldif file with ldapadd command, the input file format is LDIF(LDAP Data Interchange Format), which is of this form:
   # comment
   dn: <distinguished name>
   <attrdesc>: <attrvalue>
   <attrdesc>: <attrvalue>
   ...

   1.1 First of all we add initial.ldif to ldap server, an example snapshot of initial ldif is given below

```
dn: dc=dcisonline,dc=uohyd,dc=ernet,dc=in
objectclass: dcObject
objectclass: organization

dn: ou=People,dc=dcisonline,dc=uohyd,dc=ernet,dc=in
objectClass: organizationalUnit
objectClass: top
ou: People

dn: ou=Groups,dc=dcisonline,dc=uohyd,dc=ernet,dc=in
objectClass: organizationalUnit
objectClass: top
ou: Groups

dn: ou=Roles,dc=dcisonline,dc=uohyd,dc=ernet,dc=in
objectClass: organizationalRole
ou: Roles
```

Then run ldapadd to import the file.

```
ldapadd -x -D "cn=root,dc=dcisonline,dc=uohyd,dc=ernet,dc=in"
-f initial.ldif
```

1.2 Adding users to the system

Create users.ldif and add it to ldap server with ldapadd

```
ldapadd -x -D "cn=root,dc=dcisonline,dc=uohyd,dc=ernet,dc=in"
-f users.ldif
```

1.3 Migrating passwd or nis users

To do this we need some scripts, we use the migration perl scripts to import all users to ldif format and then add allusers.ldif to ldap server with ldapadd.

```
perl migrate_ passwd.pl -i /etc/passwd > allusers.ldif
```

and then

```
ldapadd -x -D "cn=root,dc=dcisonline,dc=uohyd,dc=ernet,dc=in"
-f allusers.ldif
```

3 Using LDAP server for authentication

Prerequisite:
We need libnss_ldap and libpam_ldap installed on clients to use ldap for authentication.

3.1 Configuring Clients to use ldap for authentication

If we want to use LDAP as central authentication server like nis server, We must configure OpenLDAP on each of the client systems, This also includes

the server as it will likely be a client unto itself (i.e. it will access the LDAP server via localhost to obtain authentication information). To do this, you must edit the /etc/ldap.conf file. The entries we are most interested in are the following:

host 172.16.88.49 <— The address of ldap server
base dc=dcisonline,dc=uohyd,dc=ernet,dc=in rootbinddn cn=root,dc=dcisonline,dc=uohyd,dc=ernet,dc=in scope one pam_filter objectclass=posixaccount pam_login_attribute uid pam_member_attribute gid
pam_password crypt <—in palce of crypt you can use md5,ssha or whatever encryption your system is using, make sure no incompatible form, otherwise it may lend you in trouble!
nss_base_passwd ou=People,dc=dcisonline,dc=uohyd,dc=ernet,dc=in?one nss_base_shadow ou=People,dc=dcisonline,dc=uohyd,dc=ernet,dc=in?one nss_base_group ou=Group,dc=dcisonline,dc=uohyd,dc=ernet,dc=in?one

Then edit /etc/nsswitch.conf

passwd: files ldap or compat ldap group: files ldap shadow: files ldap

3.2 GUI way of client configuration

Install auth-config or ldap-auth-config gui (on most of the linux versions) and set the host name and base as above.
For OpenSuse users:

1. Go to yast2
2. under Network services choose LDAP client
3. On new window (LDAP client cofiguration) choose radio button use ldap
4. Enter address of LDAP server and base DN
5. click OK.

That's it!

4 SSL certificate and TLS 1.0

SSL, or Secure Socket Layer, is a technology which allows web browsers and web servers to communicate over a secured connection. This means that the

data being sent is encrypted by one side, transmitted, then decrypted by the other side before processing. This is a two-way process, meaning that both the server AND the browser encrypt all traffic before sending out data.
There are two ways of enabling TLS in tomcat, 1- with openssl and 2- with keytool.
the tomcat server supports only JKS, PKCS11 or PKCS12 format keystores. The JKS format is Java's standard "Java KeyStore" format, and is the format created by the keytool command-line utility. This tool is included in the JDK. The PKCS12 format is an internet standard, and can be manipulated via OpenSSL.

- using keytool:

  To create a new keystore from scratch, containing a single self-signed Certificate we use following command:
  $ JAVA_HOME/bin/keytool –genkey –alias tomcat –keyalg RSA
  and specify a password value of 'changeit'.(The tomcat uses the default password as *changeit*, we can specify our own also but that we need to specify in configuration of tomcat)
  This command will create a new hidden file, as /userhomedir/.keystore.

- using OpenSSL:

  1. Create root ca
     To create a rootCA i have written a script createRootCA.sh, simply run the script, enter the self explanatory answers to questions and specify your private key(password) for RootCA, your root CA will be created.
  2. Create a CSR(Certificate Signing Request)
     openssl req –config openssl.cnf –new –nodes –keyout private/server.key –out server.csr –days 365
     Enter the answers to the self explanatory questions and your key for server certificate, on completion, two files - server.csr and private/server.key are created.
  3. Signing the csr:
     openssl ca –config openssl.cnf -policy policy_anything –out certs/server.crt –infiles server.csr
     Enter your root CA private when prompted.
  4. Import certificate to PKCS12:
     openssl pkcs12 –export –in server.crt –inkey server.key –out mycert.p12 –name tomcat –CAfile rootCA.crt –caname root –chain

5 Tomcat server configuration

The two most important configuration files to get Tomcat up and running are called server.xml and web.xml. By default, these files are located at TOMCAT-HOME/conf/server.xml and TOMCAT-HOME/conf/web.xml, respectively.

- Web.xml
  The web.xml file is derived from the servlet specification, and contains information used to deploy and configure the components of the web application. It is used to define default values for all contexts. If this method is utilized, Tomcat will use TOMCAT-HOME/conf/web.xml as a base configuration, which can be overwritten by application-specific WEB-INF/web.xml files.
  - In this file we can specify default welcome pages (e.g. index.jsp) to whatever we want like welcome.jsp or any default page.
  - we can change default time for session expire in web.xml default time is 30 min, we can have our own time set like 10 mins.
    <session-config>
    <session-timeout>10</session-timeout>
    </session-config>

- server.xml
  The server.xml file is Tomcat's main configuration file, and is responsible for specifying Tomcat's initial configuration on startup as well as defining the way and order in which Tomcat boots and builds. The elements of the server.xml file belong to five basic categories - Top Level Elements, Connectors, Containers, Nested Components, and Global Settings. All of the elements within these categories have many attributes that can be used to fine-tune their functionality. Most often, if you need to make any major changes to your Tomcat installation, such as specifying application port numbers, server.xml is the file to edit.

5.1 Enabling HTTPS

Tomcat can use two different implementations of SSL:

1. The JSSE implementation provided as part of the JRE

- Define the connector protocol in server.xml

```
<-- Define a blocking Java SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector protocol="org.apache.coyote.http11.Http11Protocol"
           port="8443" .../>

<-- Define a non-blocking Java SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector protocol="org.apache.coyote.http11.Http11NioProtocol"
           port="8443" .../>
```

- Uncomment the connector in server.xml

```
<-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<!--
<Connector
           port="8443" maxThreads="200"
           scheme="https" secure="true" SSLEnabled="true"
           keystoreFile="${user.home}/.keystore" keystorePass="changeit"
           clientAuth="false" sslProtocol="TLS"/>
-->
```

2. The APR implementation, which uses the OpenSSL engine by default.
   - Specify APR connector protocol in server.xml

```
<-- Define a APR SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector protocol="org.apache.coyote.http11.Http11AprProtocol"
           port="8443" .../>
```

   - Specify engine in listener tag in server.xml

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
          SSLEngine="on" SSLRandomSeed="builtin" />
```

   - Uncomment the connector in server.xml

```
<-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<!--
<Connector
           port="8443" maxThreads="200"
           scheme="https" secure="true" SSLEnabled="true"
           SSLCertificateFile="/usr/local/ssl/server.crt"
           SSLCertificateKeyFile="/usr/local/ssl/server.pem"
           clientAuth="optional" SSLProtocol="TLSv1"/>
-->
```

Note: Here we can specify the default port number(443) for https instead of 8443

6 Tomcat and LDAP

To use LDAP for authentication with Tomcat, we need to change in server.xml file of Tomcat the following:

```
<Realm className="org.apache.catalina.realm.JNDIRealm" debug="99"
connectionURL="ldap://localhost:389"
alternateURL="ldap://localhost:389"
userRoleName="member"
userBase="ou=People,dc=dcisonline,dc=uohyd=dc=ernet=dc=in"
userPattern="cn=0,cn=Users,ou=Roles,dc=dcisonline,dc=uohyd=dc=ernet=dc=in"
roleBase="cn=Users,ou=Roles,dc=dcisonline,dc=uohyd=dc=ernet=dc=in"
roleName="cn"
roleSearch="(member=0)"
roleSubtree="false"
userSubtree="true"
/>
```

Now define the role in the tomcat-users.xml and the web.xml.
Edit web.xml as follows:

```
<security-constraint>
<display-name>Security Constraint</display-name>
<web-resource-collection>
<web-resource-name>Protected Area</web-resource-name>
<!– Define the context-relative URL(s) to be protected –>
<url-pattern>/*</url-pattern>

<!– If you list http methods, only those methods are protected –>
</web-resource-collection>
<auth-constraint>
<!– Anyone with one of the listed roles may access this area –>
<role-name>faculty</role-name>
<role-name>student</role-name>
<role-name>staff</role-name>
<role-name>admin</role-name>
</auth-constraint>
</security-constraint>

<!– Default login configuration uses form-based authentication –>
<login-config>
<auth-method>FORM</auth-method>
<realm-name>Form-Based Authentication Area</realm-name>
<form-login-config>
<form-login-page>index.jsp</form-login-page>
<form-error-page>error.jsp</form-error-page>
</form-login-config>
</login-config>

<!– Security roles referenced by this web application –>
<security-role>
<role-name>faculty</role-name>
<role-name>student</role-name>
<role-name>staff</role-name>
<role-name>admin</role-name>
</security-role>
```

References

[1] OpenLDAP, http://www.openldap.org/

[2] JAVA, http://www.oracle.com/technetwork/java/index.html

[3] JSP W3School, http://www.w3schools.com/js/default.asp

[4] Apache Tomcat Web Server, http://tomcat.apache.org

[5] MySQL community Server, http://dev.mysql.com/downloads/mysql/5.1.html

[6] JDBC tutorial, http://www.roseindia.net/jdbc/jdbc.shtml

[7] JNDI, http://www.oracle.com/technetwork/java/jndi/index.html

[8] HTML W3School, http://www.w3schools.com/html/default.asp

[9] TLS wikipedia, http://en.wikipedia.org/wiki/Transport_Layer_Security

[10] OpenSSL, http://www.openssl.org/

[11] SQL Injection Attack, http://en.wikipedia.org/wiki/SQL_injection