# *Faculty Information System*

## Software Design Specification

**22/04/03**

**V G**
**PV**
**P J**
**R Q**

# 1. Introduction

## 1.1 Goals and Objectives

The purpose of the Faculty Information System application is to have a common repository for all faculty information. This project will also incorporate all the software engineering principles involved in the design and implementation of a complete working system. This system will help faculty members store and retrieve all their information. Students can also access specific areas of faculty information depending on the access provided. Also the information is safe by enabling SSL on the web directories.

## 1.2 System Statement of Scope

Faculty Information System is a web-based database application wherein all the information regarding the Faculty can be stored and retrieved. A user with the knowledge of how to use a browser can access and use the site. The administrator will have access to all the information present in the database, whereas the user will have privileges to view his/her information. Also users can be put into groups having access to certain areas of the website. The user interface will have the necessary input options to guide them through the process of inserting and updating their information.

Data will be inputted using data entry screens (user friendly interfaces), which will have all the steps for proper insertion. The administrator will have all the privileges to add or update any data in the system. Output will be in the form of an information repository showing user information pages.

Faculty Information System will hold the following information:

- Faculty Personal Information
- Work History for each faculty member.
- Publication archiving and retrieval for each faculty.
- Degrees earned by the faculty members.
- Grants/Awards given to faculty members.
- Courses taught by the Faculty members.

## 1.3 System Context

Faculty Information is presented as a website, which can be used by the University to hold their Faculty Information and is posted also on the web. It is portrayed as an easy-to-use tool People with minimal web browser experience can use the system to their fullest. Any student from any school or faculty member can take advantage of this tool, to view information about a professor or from the faculties point of view their own information, CSIS faculty members information can be viewed. Any student wishing to see the credentials of the professor may do so. This website allows him to see how well the professor is by taking a look at the publications that he has made and his experience in teaching. So this website is available to any student who would like to take school at Pace University.

**Potential Users**

Potential users include faculties, students, university officials such as Dean, chairs, etc and general users.

***Faculty members:*** The system must allow each faculty member to manipulate his or her own information like the basic information, publications, grants details, etc. The operations can be add, delete, and modifying the faculty data.
***University Officials:*** For the sake of faculty evaluation, university officials are often interested in faculty members' recent publications. The system must provide a good summary of all or recent publications.
***Students and general users:*** The system must allow students to easily browse the bibliography information according to faculty names or recently published research topics. Using research topic keywords, they must be able to find all faculty members who have recently published papers in their respective fields.

### 1.4 Major Constraints

Since Faculty Information System is to be used on the web; there will be no system restraints on the users. Anyone with an Internet connection and a browser will be able to use the website to the fullest of its capabilities.
The working prototype of System is expected at the end of the semester, making the drop on 05/06/2003.

## 2. Data Design

### 2.1. Internal software data structure

#### 2.1.1 Major Data Objects:

The following data modules will be presented and managed by the system:

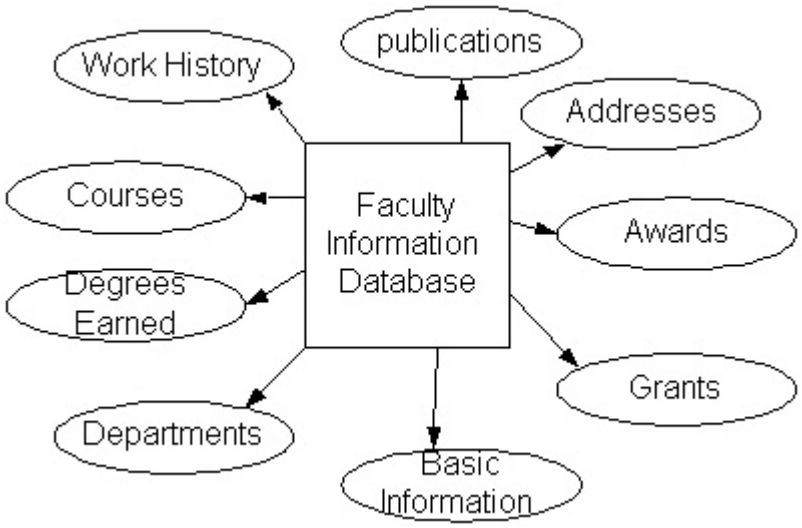
- User Authentication / Insertion Module
  This object will manage connections to the Users Database and retrieve appropriate information depending on the user login credentials. It will also manage the groups users belong to and accordingly give them access to privileged sections of the website. It will also contain processing to add a new user to specific group in the database.
- Data Entry Module
  This module will encompass all the data entry to be done in the system. Entries like the departments in the University, courses offered, office addresses, etc.
- Faculty Personal Information Module
  Here personal information for the faculty member can be entered or updated as per the privileges given.
- Work History Module.
  This module links the work history records of the faculty and displays it to the users. Users can add more records to their name depending on their rights. Updating of records is also permitted.
- Publication archiving and retrieval Module.
  This module will have the insertion and display of all the publications present in the database. Users can add or edit publications. Also search facilities will be available to search for a particular publication.
- Degrees earned by the faculty members Module.

This module links the degrees earned records of the faculty and display it to the users. Users can add more records to their name depending on their rights. Updating of records is also permitted.

- Grants/Awards given to faculty members Module.

This module links the Grants and Awards details of the faculty and displays it to the users. Users can add more records to their name depending on their rights. Updating of records is also permitted.

### 2.1.2 Relationships

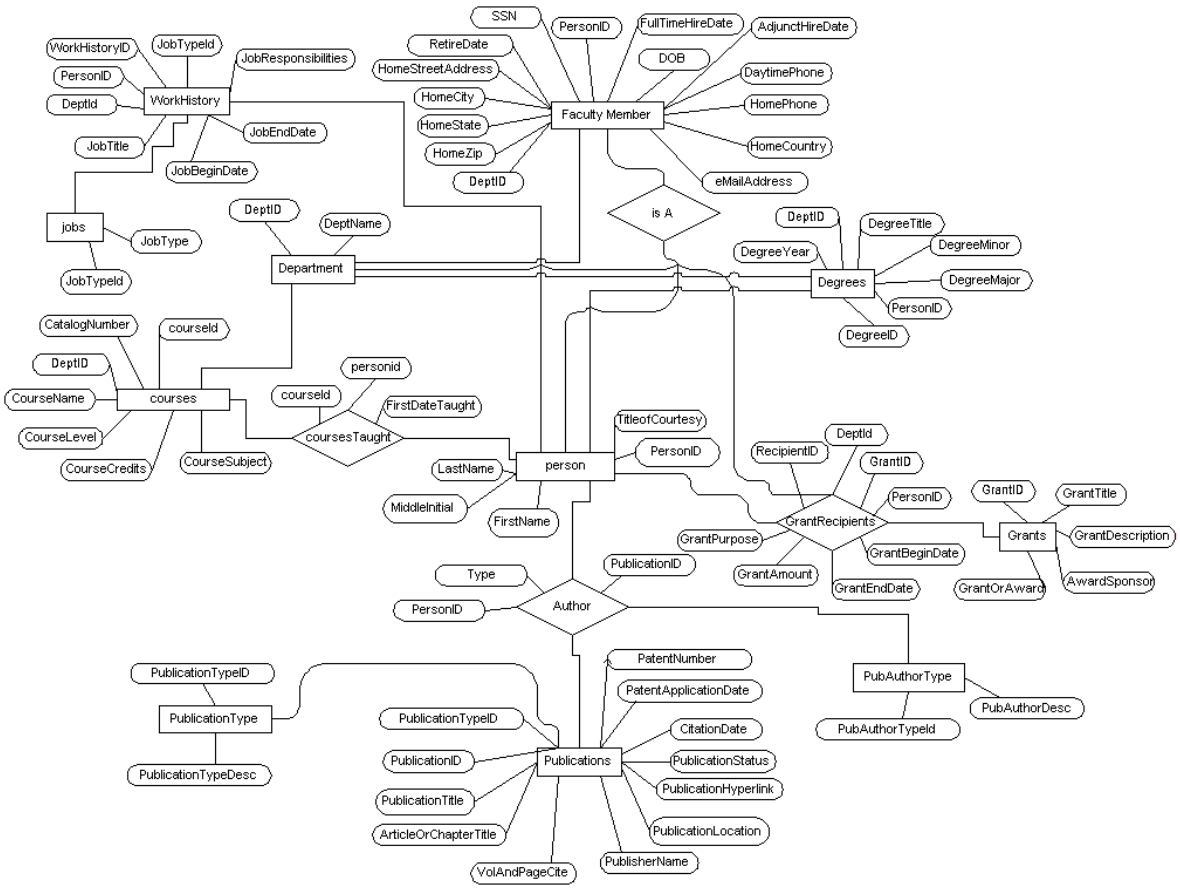



### 2.2 Database description

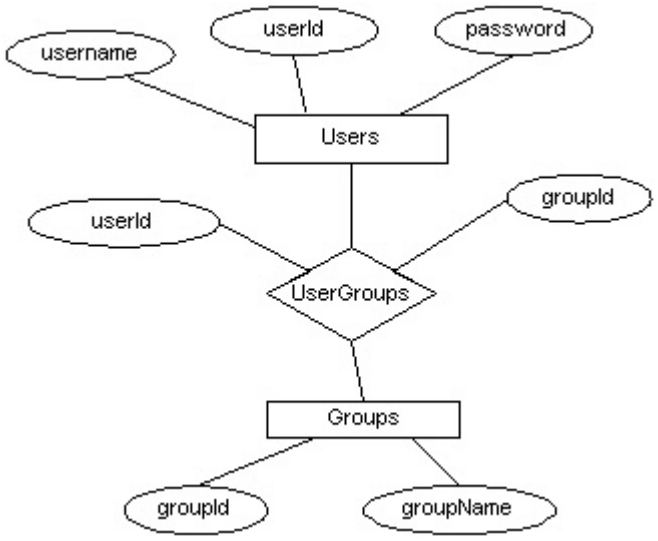
**Database Relationship Diagram**

Following is the ERD for Faculty System and its description.

ERD for Faculty Database




ERD for Users Database




**Users database**

The Users database will hold information regarding the user credentials of the System. It also has table to specify the groups the users fall in so as to grant them privileged

access. The user will get access to the Faculty System only after being authenticated by the users database.

**Users Table**

This table will hold the users in the system. It will have columns like userid, password, etc

**Groups Table**

This will hold all the groups the users fall in like administrators, instructors, students, etc.

**UserGroups Table**

This is a relationship table joining both the Users and the Groups table.

**Users Database Interface Description**

This interface will be available only to the administrator and will allow him to add new users to specific groups in the database. Users can be added to multiple groups. It will also allow the admin to remove users from a group if needed.

**Interfaces and Processing to access Faculty Information Database**

These interfaces will be accessible to users authenticated with the Users database. They will include search functions, complete faculty information screens, facilities like add and update and data entry screens.

**Interface Description**

**Search Screens**

Searches can be performed based on users, departments, publications, publication types, authors for the publication, events faculty takes part in and phone numbers to determine offices.

**Data Entry Screens**

Data entry screens will be provide to enter departments, courses offered, publication types, job types for the faculty, offices and their addresses and organizations.

**Information adding, editing and retrieval screens**

These screens will allow the user to edit and update his details. They will also allow him to add records like his/her awards or publications. The administrator will have rights to add records to any user in the database.

**Faculty Information Database**

Faculty Information System will be a repository to all the faculty information available using a web browser. It will have all search facilities, record based additions and updating, etc.

**Faculty Information Database Description**

Faculty Information System has twelve entities and four relations.

**Person and Faculty member** entity contain personal information about the faculty member.

**Publication** entity holds information regarding the publications.

**Work history** gives all the work history details for the faculty member.

**Courses** give the list of courses in the database.

**Degrees** entity will give all the information regarding the degrees earned by the faculty member.

**Grants/Awards** will display all the grants/awards earned by the members.

**Department** – List of all departments

**Jobs** – List of all jobs
**PublicationType** – List of all publication types
**PubAuthorType** – List of publication author type.
The relationship tables are
***Author*** with a *"many to many" relation* because one faculty writes multiple papers and one paper can be coauthored by many authors.
The mapping cardinality of the relationship called ***"Is key"*** is *"one to one"* because each person in the database is a faculty member.
***CoursesTaught*** is *"many to many" because* each faculty member can teach multiple courses and more than one faculty member can teach each course.
***GrantRecipients*** *is* "many to many" because each member can have more than one grant/award and each grant/award can have more than one recipient.

### 2.3 Database structure

| **Entity Users** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| userID | Int | primary key, User's ID |
| userName | Nvarchar (10) | Username for the user. |
| password | Nvarchar (10) | password |

| **Entity Groups** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| groupID | Int | primary key, Group's ID |
| groupName | Nvarchar (50) | Username for the user. |

| **Relation UserGroups (m-m)** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| userID | Int | Foreign key to Users table |
| groupId | Int | Foreign key to Groups table |

| **Entity Person** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| PersonID | Int | primary key, author's ID |
| TitleofCourtesy | Nvarchar (5) | author's title like Dr. |
| FirstName | Nvarchar (20) | author's first name |
| MiddleInitial | Nvarchar (3) | author's middle initial |

| | | |
|---|---|---|
| LastName | Nvarchar (20) | author's last name |

| **Entity FacultyMember** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| PersonID | Int | primary key, unique identification number |
| SSN | varchar(20) | Social Security Number |
| HomeStreetAddress | nvarchar (80) | Address |
| HomeCity | nvarchar (30) | City |
| HomeState | nvarchar (20) | State |
| HomeZip | nvarchar (10) | Zip Code |
| HomeCountry | nvarchar (20) | Country |
| HomePhone | nvarchar (15) | Home Phone Number |
| Daytime Phone | nvarchar (15) | DayTime Phone |
| AdjunctHireDate | datetime | Adjunct Hire Date for the faculty |
| FullTimeHireDate | datetime | Full Time Hire Date |
| RetireDate | datetime | Retire Date |
| eMailAddress | Nvarchar(30) | Email address |
| DOB | datetime | Date of Birth. |

| **Entity Publications** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| PublicationID | number(4) | primary key, unique identification number for the publication. |
| PublicationTitle | number(3) | Title for the publication. |
| ArticleOrChapterTitle | number(2) | Article or chapter title |
| VolAndPageCite | varchar2(50) | The volume and citation |
| PublisherName | Nvarchar(125) | Publisher for the publication |
| PublicationLocation | Nvarchar(100) | Location where published |
| PublicationHyperlink | ntext | Web link for the publication. |
| PublicationStatus | Nvarchar(30) | Status where on hold or published |
| CitationDate | Nvarchar(20) | Citation date |
| PatentApplicationDate | datetime | Patent application date |
| PatentNumber | (50) | Patent Number |
| PublicationTypeID | int | Publication type id like paper publications,journals,etc references PublicationType |

| Entity **WorkHistory** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| WorkHistoryID | int | primary key, unique identification number for table |
| PersonID | int | Foreign key references Person |
| DeptId | int | Foreign key references Department |
| JobTitle | Nvarchar(100) | Title for the job held by the faculty member. |
| JobBeginDate | datetime | Job starting date |
| JobEndDate | datetime | Job end date |
| JobResponsibilities | Nvarchar(250) | Job responsibilities |
| JobType | smallint | Foreign key references Jobs |

| Entity **Degrees** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| DegreeID | int | primary key, unique identification number. |
| PersonID | int | Foreign key references Person |
| DegreeMajor | nvarchar (200) | Degree Major for the member |
| DegreeMinor | nvarchar (40) | Degree Minor |
| DegreeTitle | nvarchar (20) | Title for the degree |
| DeptId | int | Foreign key references Department |
| DegreeYear | datetime | Year the Degree was earned. |

| Entity **Grants** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| GrantID | int | primary key, unique identification number. |
| GrantTitle | Nvarchar(150) | Grant/Award title |
| GrantDescription | Nvarchar(250) | Grant/Award description |
| AwardSponsor | Nvarchar(150) | Grant/Award sponsor |
| GrantOrAward | bit | Field to determine whether the entry is for a grant or the award. |

| Entity **Courses** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| CourseID | int | primary key for the table |
| DeptId | int | Foreign key references Department |

| | | |
|---|---|---|
| CatalogNumber | Nvarchar(15) | The catalog number for the course |
| CourseName | Nvarchar(100) | Course name |
| CourseLevel | Nvarchar(20) | Level like 504 or 600 |
| CourseCredits | Smallint | Course credits |
| CourseSubject | Nvarchar(5) | Course name. |

| **Entity Department** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| DeptId | int | Primary Key |
| DeptName | Nvarchar(50) | Department name |

| **Entity PubAuthorType** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| PubAuthorTypeId | smallint | Primary Key |
| PubAuthorDesc | Nvarchar(50) | Publication author type description |

| **Entity users** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| userId | Nvarchar(100) | User id |
| password | Nvarchar(100) | password |

| **Entity PublicationType** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| PublicationTypeID | Int | Primary key |
| PublicationTypeDescr | Nvarchar(50) | Type descriptions for publications. |

| **Entity Jobs** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| JobTypeId | smallint | Primary key |
| JobType | Nvarchar(100) | Type descriptions for publications. |

| **Relation Author (m-m)** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| Type | smallint | Foreign key references PubAuthorType. |
| PublicationID | number(4) | foreign key to Publications |

| | | |
|---|---|---|
| PersonID | int | foreign key to person |

| **Relation CoursesTaught (m-m)** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| CourseID | int | foreign key to Courses |
| PersonID | int | foreign key to Person |
| FirstDateTaught | number(2) | Date the course was first taught. |

| **Relation GrantRecipients (m-m)** | | |
|---|---|---|
| **Attribute** | **type** | **Description** |
| RecipientID | smallint | Foreign key references PubAuthorType. |
| DeptId | int | foreign key to Departments |
| GrantID | int | foreign key to Grants |
| PersonID | int | foreign key to Person |
| GrantBeginDate | datetime | Begin date |
| GrantEndDate | datetime | End Date |
| GrantAmount | money | Grant/Award Amount |
| GrantPurpose | Nvarchar(250) | Purpose |

## 3. Component-level design

### 3.1 Program Structure

JavaScript and html will be used to build the Faculty Information user interface. The Engine will be written using server-side scripts.

### 3.2 Description for Authenticator

#### 3.2.1 Processing narrative (PSPEC) for Authenticator

The Authenticator performs the login name and password validation for the Faculty Information System. The Authenticator receives the login name and password for the user. If it is valid in the system it returns a success indicator, if not a failure flag is returned.

#### 3.2.2 Authenticator interface description

*Input interface* - Login name and password.
*Output interface* - A flag indicating if Authenication was successful or not.

#### 3.2.3 Authenticator processing detail

##### 3.2.3.1 Algorithmic model (PDL)

*Authentication Procedure:*

Lookup Login name and Password in data files.
If login is found and valid
Return a success flag
Else return a failure flag

#### 3.2.3.2 Restrictions/limitations
This method of authentication is not secure.

#### 3.2.3.3. Local data structures
The local Data structures used in the Faculty Information System are the Faculty specific recordsets. Also data structures holding basic information will need to be created.

#### 3.2.3.4 Performance issues
None

#### 3.2.3.5 Design constraints
None

## 3.3 Description for Search Engine

### 3.3.1 Processing narrative (PSPEC) for Search Engine
The Search Engine is responsible for retrieving all information about a faculty. It is also responsible for saving all data when modifying a faculty's information, and for saving the data that has been updated after the completion of each update.

### 3.3.2 Search Engine interface description
*For retrieval:*
*Input interface* - Search by first name, Search by last name, Search by course offered.
*Output interface* – Information Screens with desired information.

### 3.3.3 Search Engine processing detail

#### 3.3.3.1 Algorithmic model (PDL)
*Store Database Procedure:*
Extract all data from Database.
Save data in file.

*Retrieve Information Procedure:*
Find faculty in file.
Extract data out of files and put into Information screens and data structures.
Return desired information screens.

*Save Update Procedure:*
Extract data out of Faculty Information data structure.
Save data in file.

#### 3.3.3.2 Restrictions/limitations
This component is limited to simple file access, but could be modified for use with a relational database.

#### 3.3.3.3 Local data structures
The local data structures the search engine would use are:
Faculty First name and Last name
Courses being Offered

#### 3.3.3.4 Performance issues
The performance would only be an issue if too many users are on the system at the same time and are trying to access the same files.

#### 3.3.3.5 Design constraints
A constraint for design is that the search engine must be written using server-side scripting so that it has access to the data files on the server. It also must have intimate knowledge about how the file system on the server is set up so it knows how to access the appropriate files.

## 4. User Interface Design

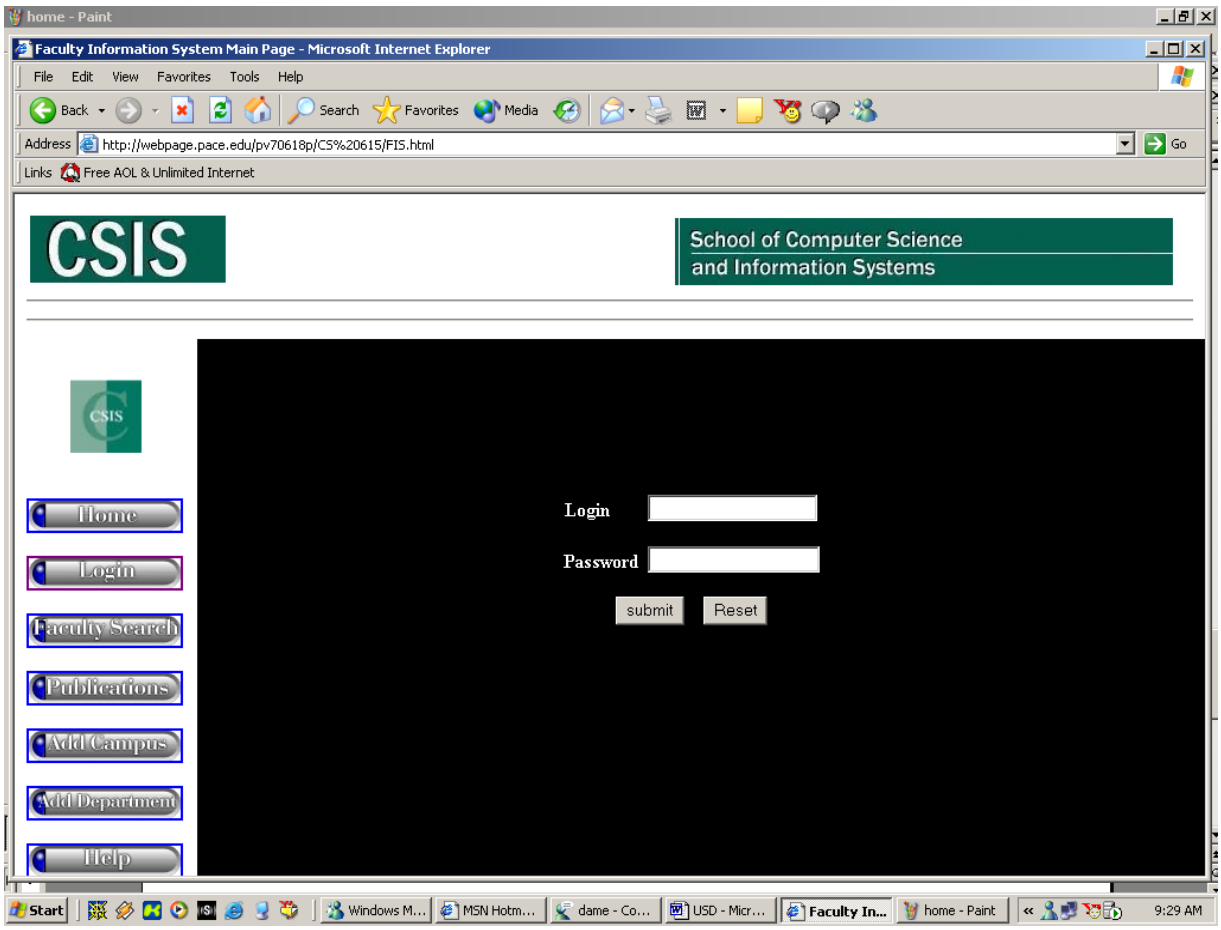
### 4.1 Description of the user interface
The following sections explain the basic screens necessary for collecting the necessary user input for creating and taking a quiz.

#### 4.1.1 Screen images
**Introductory: Home Page**
The first screen that the user will face when wanting to get any type of information about the faculties.

Home Page


**Login: User Login Screen**
The User Login screen allows the user to input their name and their password. It has inputs for such information as the Login Name, Password, Reset Button and Submit Button.

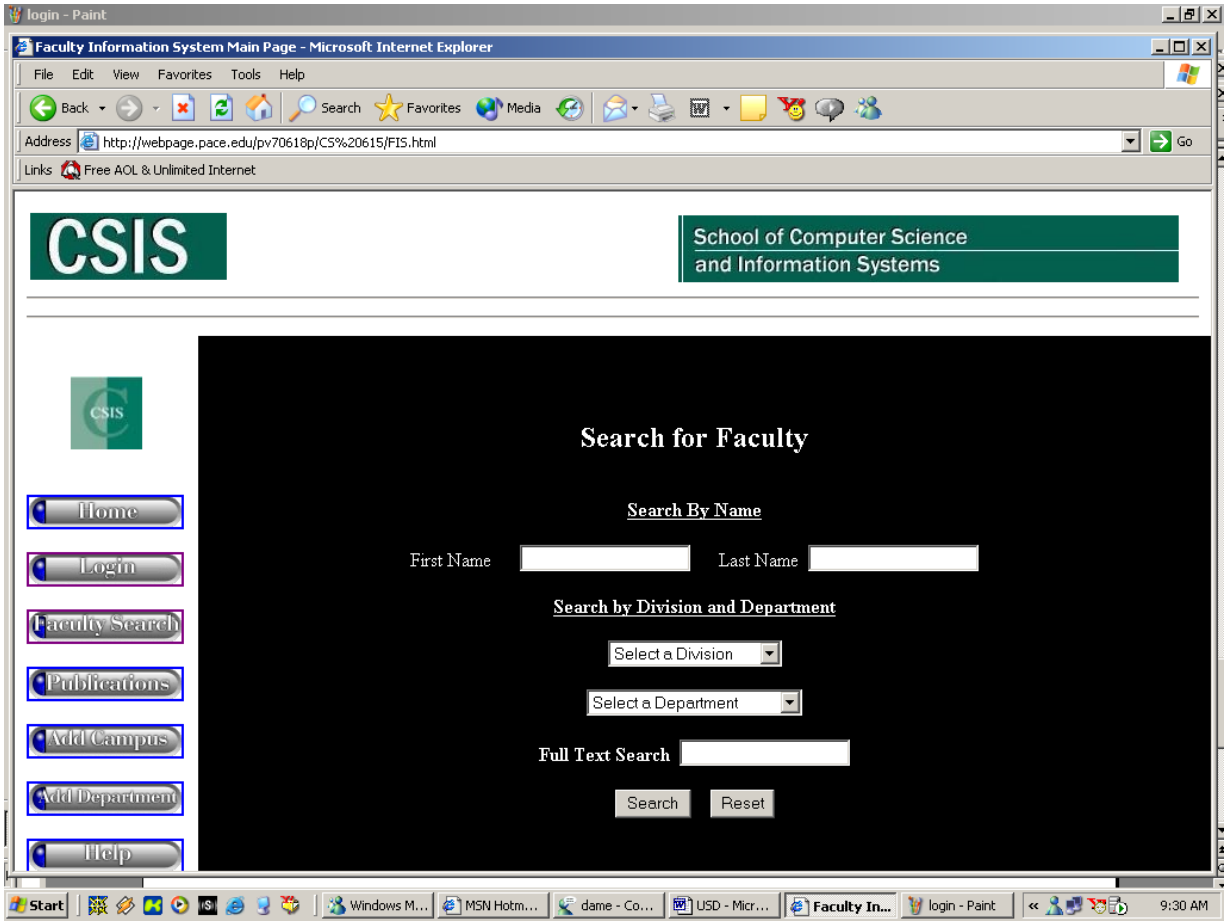
User Login




**Search Page: Faculty Search Screen**

The Faculty search screen is where the user inputs the first name, last name and course offered.

Faculty Search Engine




### 4.1.2 Objects and actions

Introductory*: Home Page*

This screen allows the users to get a feel of the database. It has several links to the login screen, Faculty search screen, Help screen and link back to home page

Login*: User Login Screen*

This screen allows the user to enter their id and password in the available textbox objects, and submit them using the appropriate button to trigger the validation action.

Search Screen*: Faculty Search Screen*

This screen has textbox objects for entering Faculties First name, Last name. It haws also drop down lists through which the user is able to search for the faculty he is looking for.There are necessary text boxes for doing a full search as well. The search action is performed by clicking on the search button. The reset Button click will help the user to start the search all over again.

Help*: Faculty Help Page*

This screen basically helps the user to optimize his search by giving a detailed information about the different ways through which a search can be performed.

## 5. Restrictions, Limitations, and Constraints

The following items include the limitations and constraints that are part of the specifications.

1. No time limitations will be imposed on the users.
2. Each faculty member can edit only his or her own information
3. Only administrator will have all rights.
4. Since Faculty Information System is completely browser-based, the users will need an acceptable browser program in order to use the system.

## 6. Validation Criteria

### 6.1 Description for software behavior

We will be using the following well-established Software testing methodologies to uncover errors.

- Unit Testing.
- Integration Testing.
- Validation Testing.

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

Integration testing is a systematic technique for constructing the program. Structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

Validation testing demonstrates traceability to software requirements, and System testing validates software once it has been incorporated into a larger system. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer. Software validation is achieved through a series of Black-Box tests that demonstrate conformity with requirements. A test plan outlines the classes of tests to be conducted and test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Black-Box tests are designed to validate functional requirements without regard to the internal workings of a program. Black box testing techniques focus on the information domain of the software, deriving test cases by partitioning the input and output domain of a program in a manner that provides through test coverage. Equivalence

partitioning divides the input domain into classes of data that are likely to exercise specific software functions.