DETAILED DESIGN

Project Code:	CMS
Project Name:	College Management System

_____	_____	_____
Prepared by/Date	**Reviewed by/Date**	**Approved by/Date**

Table of Contents

1.Introduction

1.1 Purpose

This system can be used as an application for the **College Management System** to manage the college information and students' information. The system is an online application that can be accessed throughout the organization and outside customers as well with proper login provided, which will give better service to the customers.

1.2 Scope

This system can be used as the Office of Alumni and College Relations seeks to protect the privacy of its alumni and friends, and thus, endeavors to safeguard the use of information in its custody. To that end, the Office of Alumni and College Relations provides constituent information to requestors only under the conditions.

1.3 Overview

Overall description consists of background of the entire specific requirement. It also gives explanation about actor and function which is used. It gives explanation about architecture diagram and it also gives what we are assumed and dependencies. It also supports specific requirement and also it support functional requirement, supplementary requirement other than actor which is used. It also gives index and appendices. It also gives explanation about any doubt and queries.

Once a student graduates from the institute, his/her professional life or career begins, with higher education playing an important role in establishing himself/herself in the profession. In respect of college, it has been our experience that from the very beginning, the alumni have maintained personal contacts with one another, rather than use the channel of Alumni Association.

The advancements in information technology have certainly helped in creating new resources such as alumni web pages, list servers etc., so as to permit greater interactions between the alumni. The College Alumni Association has been quite active in this respect and has revived the interest among the Alumni to remain in touch.

2.System Analysis

2.1 Existing System

The Existing system is a computerized system but which is maintained at individual databases i.e in

excel sheets, it's a time delay process. And maintaining all the records in Excel sheets is difficult. If they want any record, they have to search all the records. It doesn't provide multiple user accessibility and also doesn't have different user privileges. So, the system is not accessible for all the employees of the organization.

2.1.1 Limitations in Existing System

- The current system is not completely complete computerized and manual system in entering students and staff data and handling it.
- There is no centralized database maintenance
- There is no easy access to the particular students record
- The student cannot easily navigate through the database

2.2 Proposed System

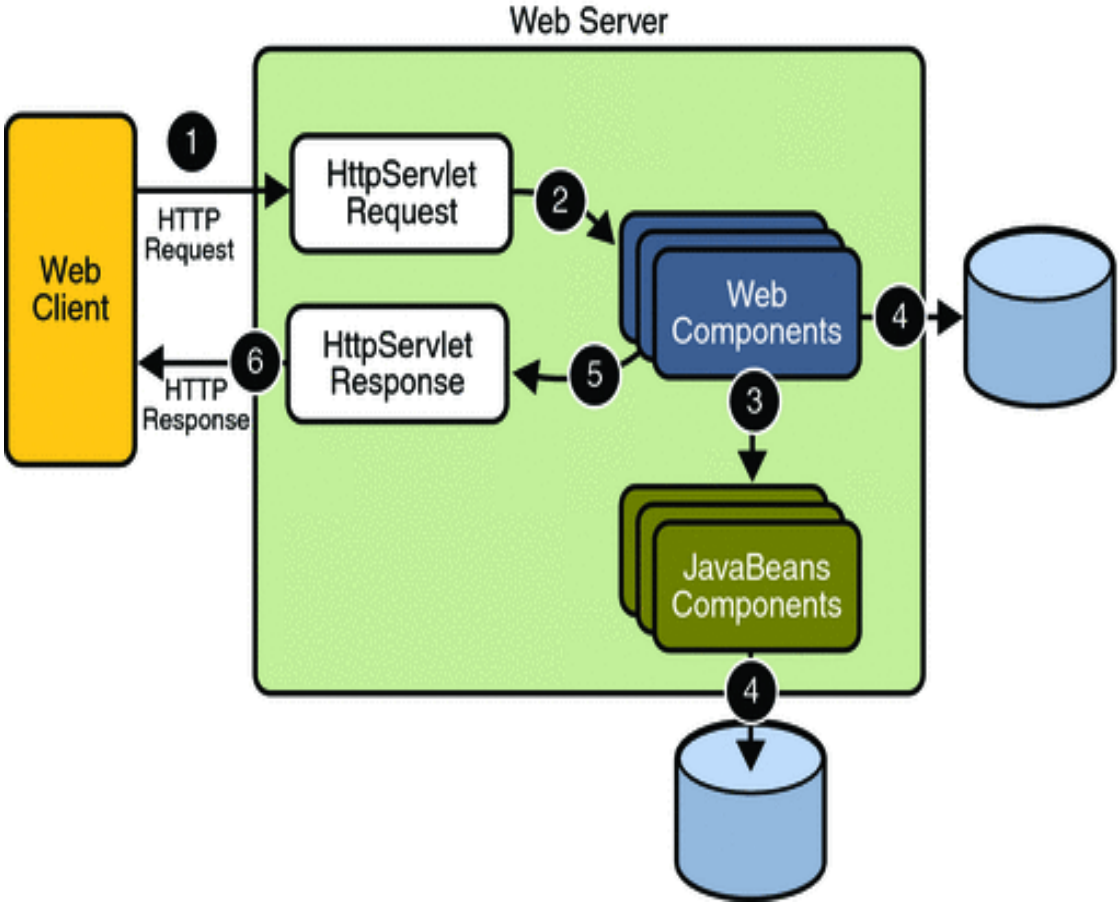
The Proposed system is a computerized system but which is maintained at Centralized databases i.e., in automated forms it's a very fast process. And maintaining all the records in online systems database which makes it very easy to access and retrieve data from the database. If they want any record, they can easily search all the records. It provides multiple user accessibility and also has different user privileges. So, the system is accessible for all the employees of the organization.

2.2.1 Advantages over Existing System

- It is completely automated system in handling the college database
- This system provides centralized database maintenance
- This system provides easy access to the particular students account or his complete details
- This system provides student to easily navigate through the application for more information in a most secure manner.

3.System Design

3.1 Architecture Diagram



3.2 Authentication

3.2.1 Functional Description

a. Login to the system through the first page of the application.

b. Change the password after login to the application.

c. See his/her details and change it.

d. Help from the system.

3.2.2 Functions

- ✓ Admin Users
- ✓ Student
- ✓ User Interface
- ✓ Faculty

3.3 Maintenance

3.3.1 Functional Description

Following is a list of functionalities of the system. More functionality that you find appropriate can be added to this list. And, in places where the description of functionality is not adequate, you can make appropriate assumptions and proceed.

A person should be able to
Click on a button or link to be taken to the specific application.
The first page of the application should provide the user (student only) with the options of saving his/her details in the database and searching for alumni.

The page through which the student enters his/her details should allow saving of user details like name, year of passing, roll number, contact number, present address, permanent address, etc. This page will be having 2 buttons Save and Cancel.

The save page should have proper validation before allowing the user to save user details into the database.

The cancel button should take the user back to the home page (first page).
The search page should allow search on the basis of name of the user or roll number. Partial search should be allowed.

3.3.2 General Operations:

Student

- ✓ User can Register to Alumni
- ✓ Students can login to the system
- ✓ Update the Profile
- ✓ Students can see all student's information in the alumni
- ✓ They can see the events details
- ✓ Students also can see the campus interview details

Admin users

- ✓ Has full access to all the modules of this system.
- ✓ Responsible for the accounts of all students.
- ✓ Update, modify or delete event details
- ✓ Update, modify or delete campus interviews details
- ✓ Prepares and submits also Daily reports, user reports, event reports, etc.

User Interface:

Soothing Graphical User Interface with Context Sensitive Help

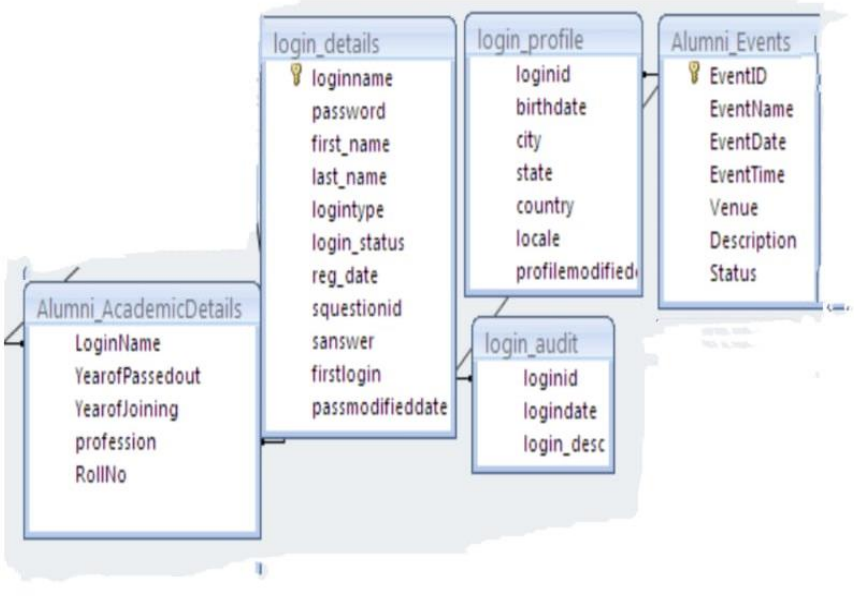
Totally Menu Driven, with Keyboard Shortcuts for frequently used forms

All reports for specified period selected using calendar

Faculty:

- ✓ Faculty can add the tutorials
- ✓ Faculty can register to get their individual id through mail
- ✓ Faculty can see the results.
- ✓ Faculty can prepare the exam question paper.

3.4 E - R Diagrams



3.5 UML Diagrams

UNIFIED MODELING LANGUAGE DIAGRAMS

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

User Model View

This view represents the system from the user's perspective.

The analysis representation describes a usage scenario from the end-user's perspective.

Structural model view

In this model the data and functionality are arrived from inside the system.

This model view models the static structures.

Behavioral Model View

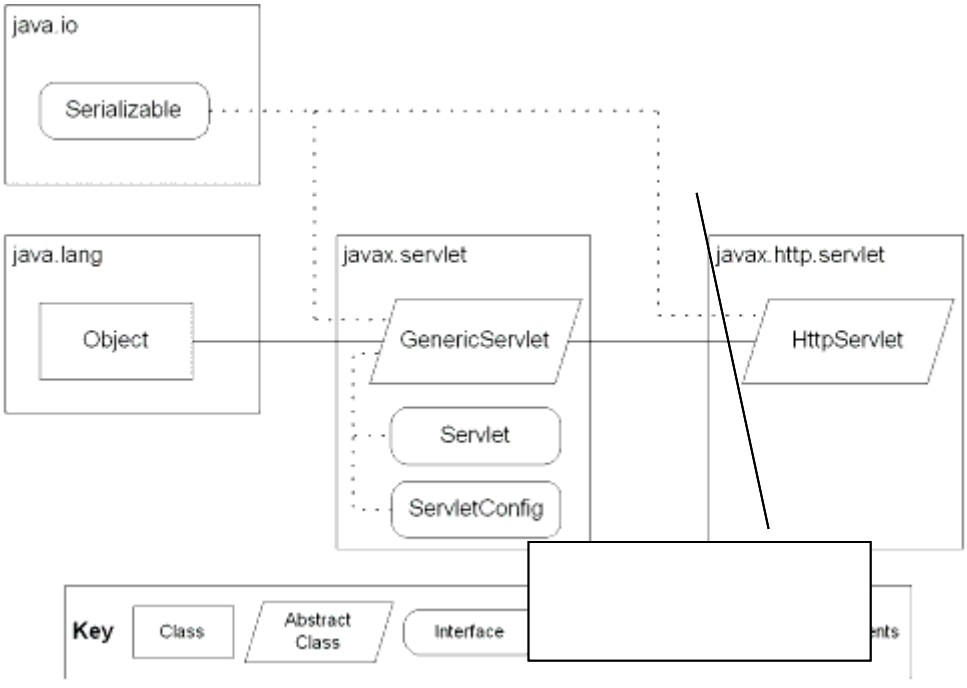
It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

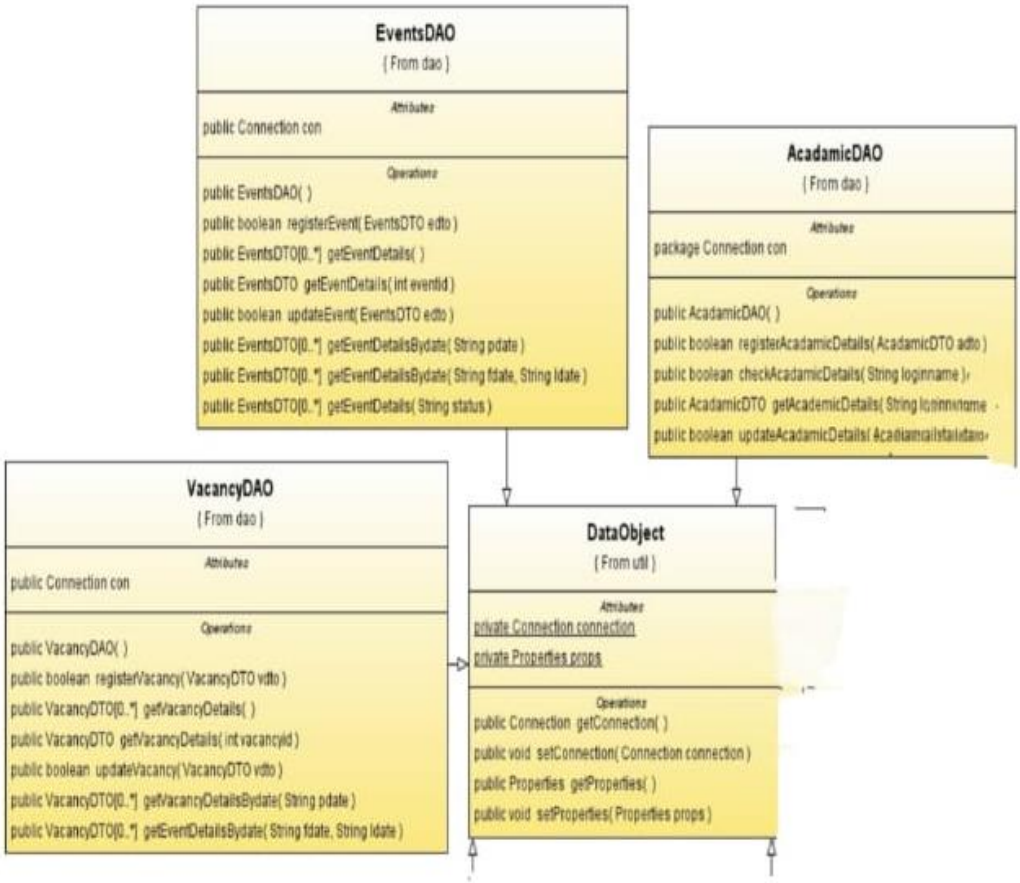
In this the structural and behavioral as parts of the system are represented as they are to be built.

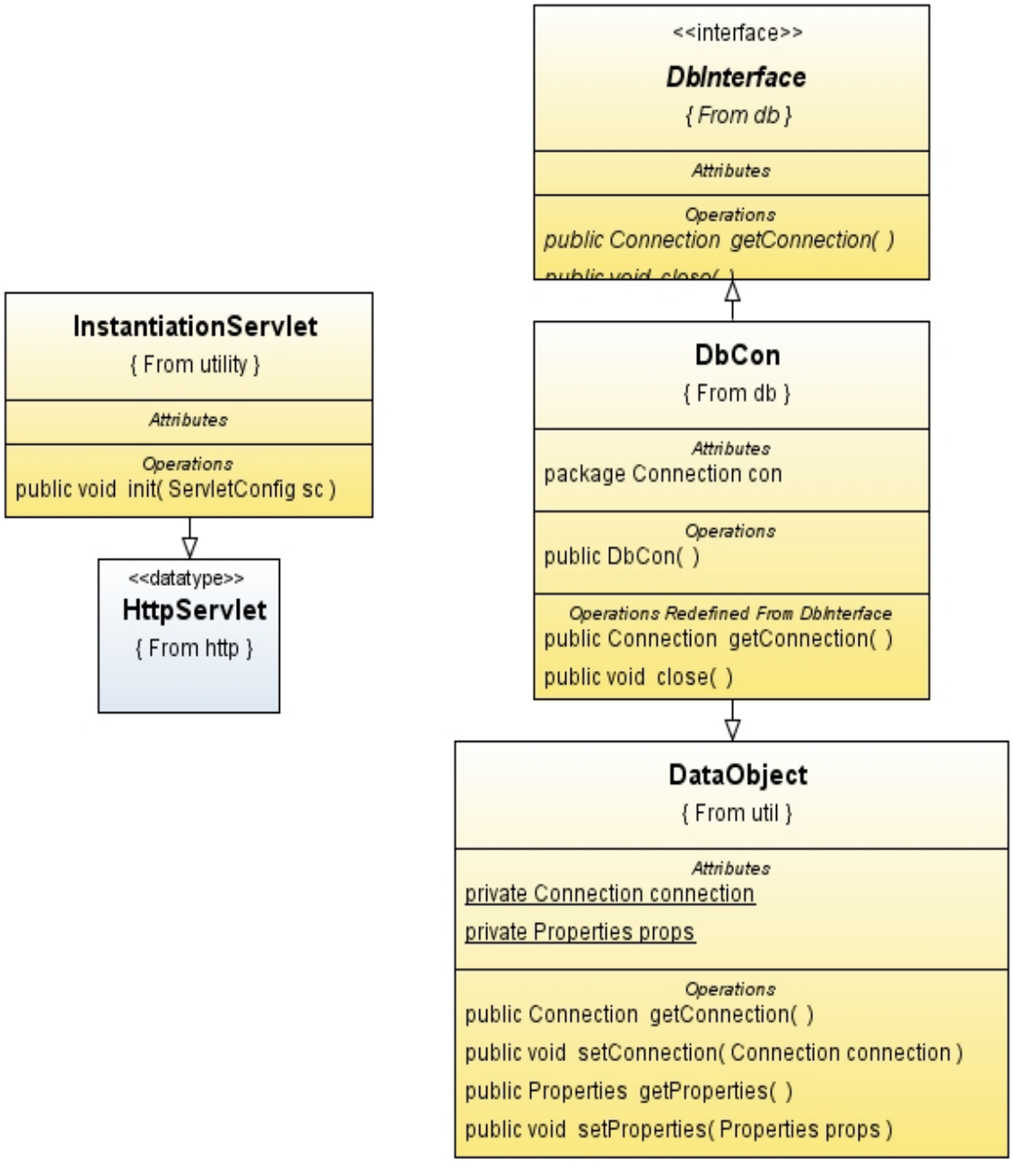
3.5.1 Class Diagram

Class diagrams describe the structure of the system in terms of classes and objects. The servlet Api class diagram will be as follows.

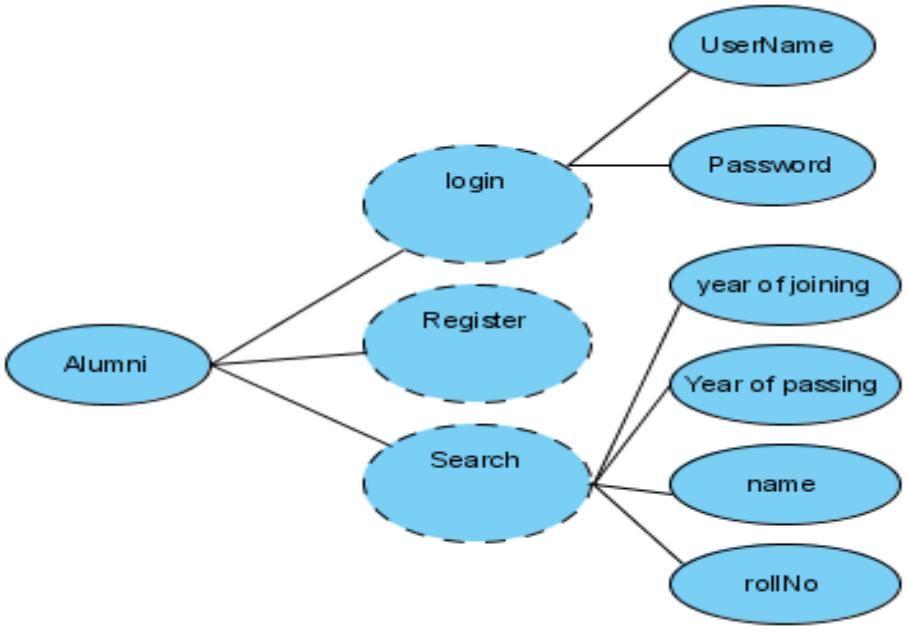


JSP: Implicit Objects

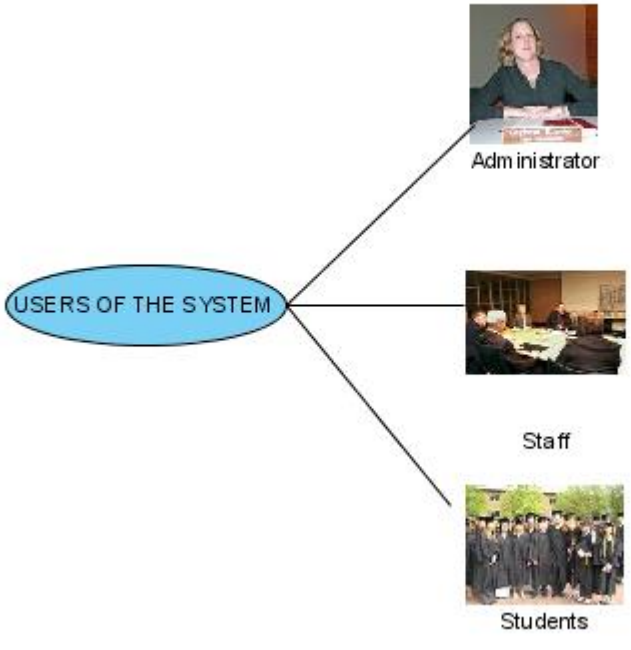

EventsDAO
{ From dao }
Attributes
public Connection con
Operations
public EventsDAO( )
public boolean registerEvent(EventsDTO edto)
public EventsDTO[0..*] getEventDetails( )
public EventsDTO getEventDetails(int eventId)
public boolean updateEvent(EventsDTO edto)
public EventsDTO[0..*] getEventDetailsBydate(String pdate)
public EventsDTO[0..*] getEventDetailsBydate(String fdate, String ldate)
public EventsDTO[0..*] getEventDetails(String status)
AcadamicDAO
{ From dao }
Attributes
package Connection con
Operations
public AcadamicDAO( )
public boolean registerAcadamicDetails(AcadamicDTO adto)
public boolean checkAcadamicDetails(String loginname)
public AcadamicDTO getAcademicDetails(String loginnname
public boolean updateAcadamicDetails(Acadiamcdtaildato
VacancyDAO
{ From dao }
Attributes
public Connection con
Operations
public VacancyDAO( )
public boolean registerVacancy(VacancyDTO vdto)
public VacancyDTO[0..*] getVacancyDetails( )
public VacancyDTO getVacancyDetails(int vacancyId)
public boolean updateVacancy(VacancyDTO vdto)
public VacancyDTO[0..*] getVacancyDetailsBydate(String pdate)
public VacancyDTO[0..*] getEventDetailsBydate(String fdate, String ldate)
DataObject
{ From util }
Attributes
private Connection connection
private Properties props
Operations
public Connection getConnection( )
public void setConnection(Connection connection)
public Properties getProperties( )
public void setProperties(Properties props)


<<interface>>
DbInterface
{ From db }
Attributes
Operations
public Connection getConnection( )
InstantiationServlet
{ From utility }
Attributes
Operations
public void init(ServletConfig sc)
<<datatype>>
HttpServlet
{ From http }
DbCon
{ From db }
Attributes
package Connection con
Operations
public DbCon( )
Operations Redefined From DbInterface
public Connection getConnection( )
public void close( )
DataObject
{ From util }
Attributes
private Connection connection
private Properties props
Operations
public Connection getConnection( )
public void setConnection(Connection connection)
public Properties getProperties( )
public void setProperties(Properties props)

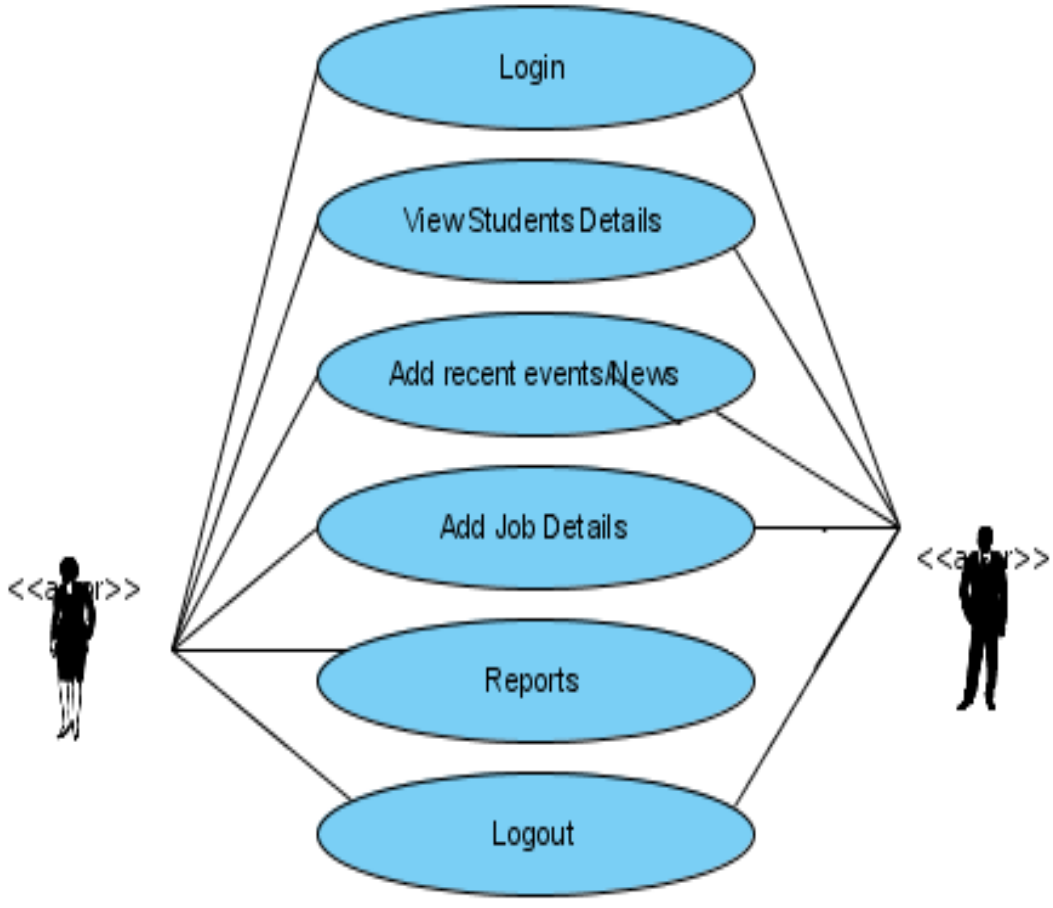
3.5.2 Use-case Diagram



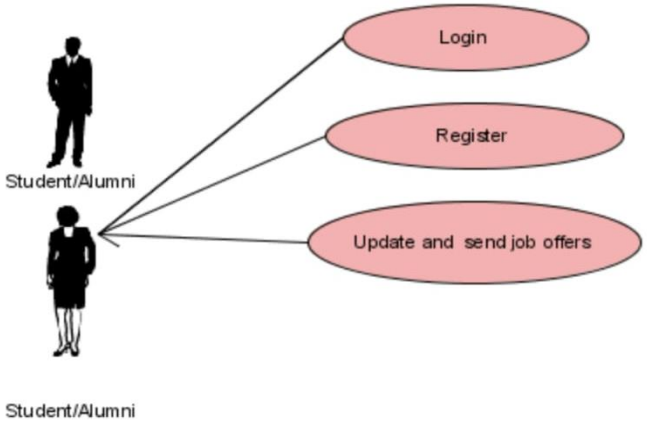
3.5.3 USERS OF THE SYSTEM



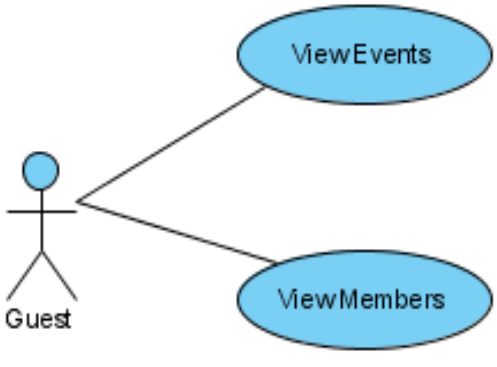
3.5.4 Administrator Use Case



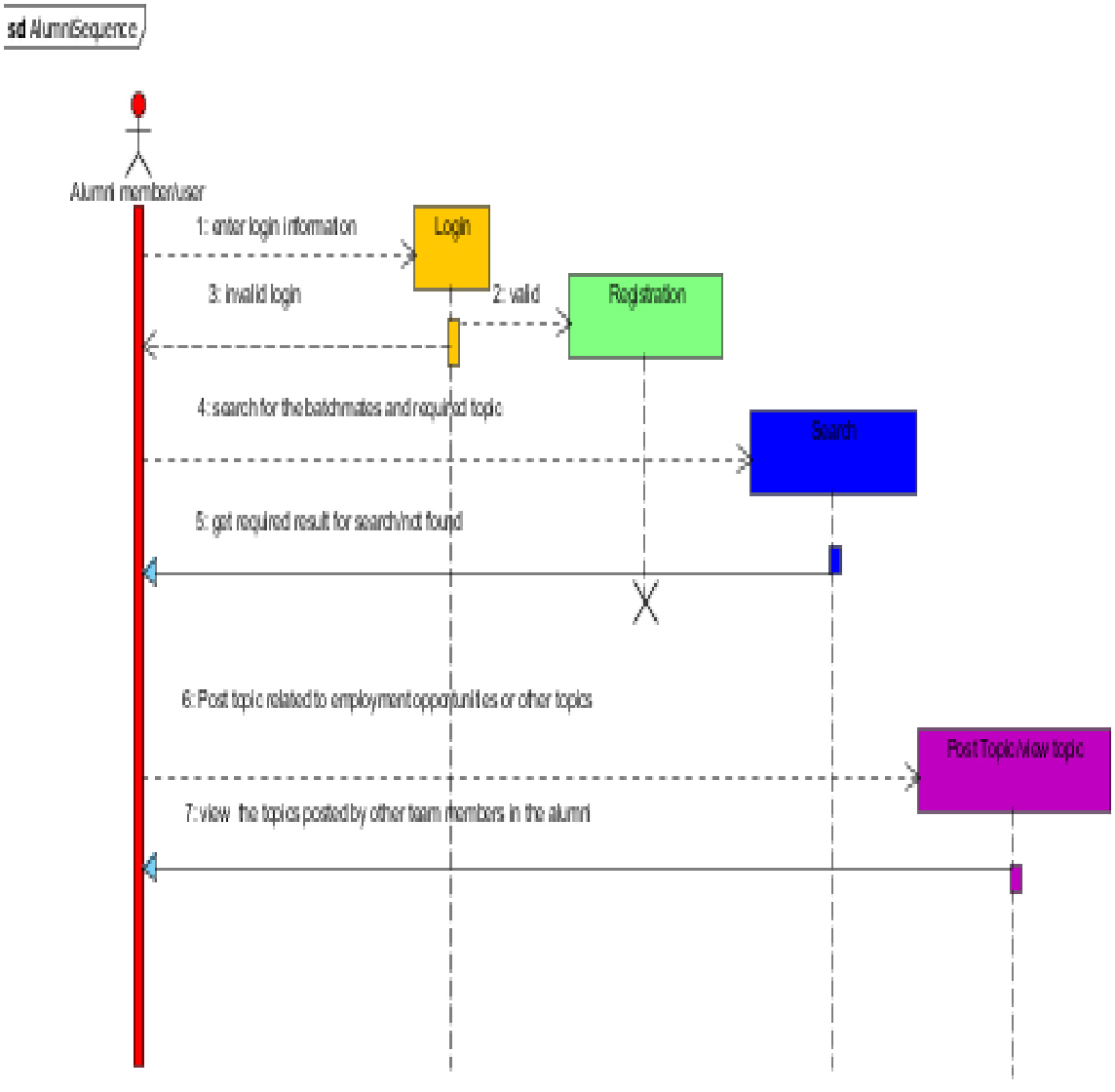
3.5.5 Student/Alumni UseCase



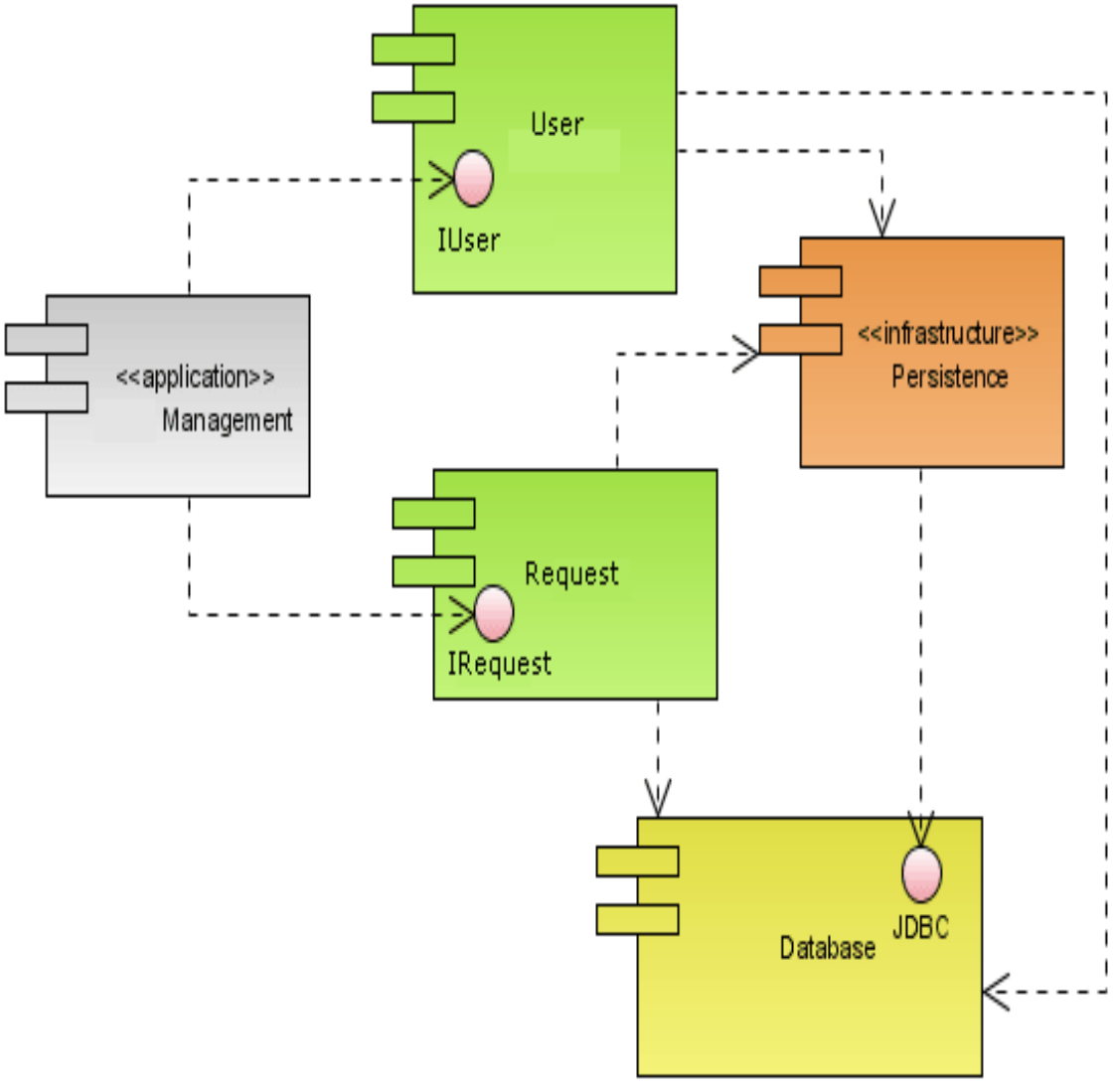
3.5.6 Guest UseCase



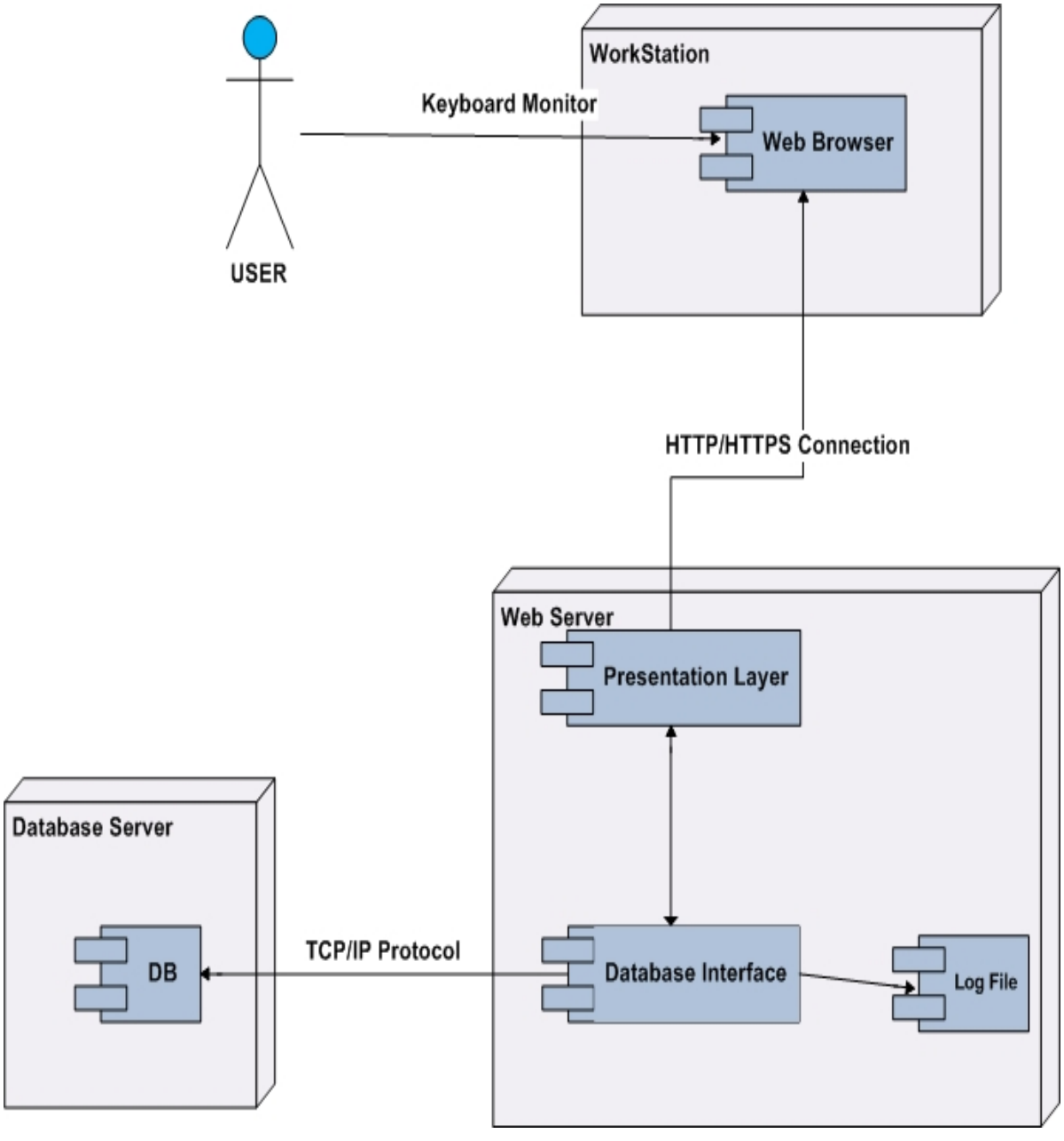
3.5.7 SEQUENCE DIAGRAM FOR ALUMNI



3.5.8 Component Diagram



3.5.9 Deployment Diagram



4.Data Dictionary

4.1 Alumni Events:

ColumnName	DataType	Size
EventID	Number	255
EventName	text	255
EventDate	date/time	10
EventTime	text	255
Venue	text	255
Description	text	255
Status	text	255

4.2 LoginProfile:

ColumnName	DataType	Size
loginid	text	255
birthdate	date/time	10
city	text	255
state	text	255
country	text	255
locale	text	255
profilemodifieddate	date/time	10

4.3 LoginAudit:

ColumnName	DataType	Size
loginid	text	255
logindate	date/time	10
login_desc	text	255

4.4 LoginDetails:

ColumnName	DataType	Size
loginname	text	255
password	text	255
first_name	text	255
last_name	text	255
logintype	text	255
login_status	number	10
reg_date	date/time	10
squestionid	text	255
sanswer	text	255

firstlogin	number	10
passmodifieddate	date/time	10

4.5 Questionbase:

ColumnName	DataType	Size
question_Id	number1	10
question_detail	text	255

5.Limitations and Scope for Future Enhancements

5.1 Limitations of the system:

- Only the permanent employees can access the system.
- System works in all platforms and its compatible environments.
- Advanced techniques are not used to check the authorization.

5.2 Future Enhancements:

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Because it is based on object-oriented design, any further changes can be easily adaptable.
- Based on the future security issues, security can be improved using emerging technologies.
- Attendance module can be added
- sub admin module can be added.

6. Requirement Specification

Operating System	:	Windows XP/2003 or Linux/Solaris
User Interface	:	HTML, CSS
Client-side Scripting	:	JavaScript
Programming Language	:	Java
Web Applications	:	JDBC, JNDI, Servlets, JSP
IDE/Workbench	:	My Eclipse
Database	:	Oracle/Access
Server Deployment	:	Tomcat

7. Project Summary

The **College management system** is a web-based application for primarily providing training to the employees who provide customized solutions to meet organizational needs.

This application software has been computed successfully and was also tested successfully by taking "test cases". It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using Java as front end and Oracle as back end in Windows environment. The goals that are achieved by the software are:

- ✓ Instant access.
- ✓ Improved productivity.
- ✓ Optimum utilization of resources.
- ✓ Efficient management of records.
- ✓ Simplification of the operations.
- ✓ Less processing time and getting required information.
- ✓ User friendly.