15th March,2013

Integrated Campus

Software Design Document
Increment - II

Group 2

Revision History:

Version	Primary Author(s)	Description of Version	Reviewed By	Date Completed
v_2.0	Sushant Pritmani, Nalin Patidar	This is the first version of SDD Increment - 2	Vipul Garg, Pinky J Meena	15-03-2013

Introduction

Purpose

The purpose of this document is to present a detail description about design aspect of the software under the project: “Integrated Campus”. The document presents all the design considerations of the software that would be implemented in the first increment of the product.. The document also mentions the various design constraints, assumptions and dependencies which the software suffers from.

Document Overview

This is a design document for **INTEGRATED CAMPUS** –

1. The first part is the overview of the design of this application.
2. The second part is about design considerations.
3. The third part is about technical design including class design.
4. The fourth part introduces the data design.
5. The document concludes with activity diagram.

Target Audience

The document contains detailed information regarding implementation procedure and it is intended for the technical team.

Scope

With the exponentially growing dependency of the academic institutes on the internet, accessibility and manageability of the resources on both the student’s and the faculty's end is the need of the hour. Many a times user has to wander from one application to other for different kinds of information which is time consuming and it also increases the work load on the administration too which has to manage all these applications. Instead of carrying a separate basket for each egg, putting them all together in a single one seems to be a much better way out.

And that is what the purpose of the project is. It is all about getting the important pieces of the academic picture on one canvas in an eye soothing way.
'INTEGRATED CAMPUS' as the name suggests, integrates all the essential requirements of an academic institutes from both the student's and the faculty's perspective on one platform whether it is academics related, interaction related, student attendance related, results related, so on and so forth.

Scope of the document

This document is a part of the software development for the web application INTEGRATED-CAMPUS. As we are following evolutionary incremental model for software development, this document is the design specification document for the first version of the application.

This document contains both high level designs and low level designs of the application using Data Flow diagram, Entity Relationship diagram, Sequence Diagram, Use Cases and Use Cases diagram and User Interface Diagrams.

System Overview

This document is created after the requirements are clearly understood in Requirements phase. This document describes in detail how various modules are implemented. The system has various users, and it is designed such that every user has certain functionalities, and the modules work independently, each user is included in some or other module. The system will follow the three-tier architectural style and be organized into three layers:

- Interface layer
- Application layer
- Storage layer.

The Interface layer will be the graphical user interface that allows the users to interact with the system. It will be implemented using Dreamweaver, sublime etc. The Application layer will contain the logic and algorithm depending on which privileges are assigned with user and database could be queried. Finally, the Storage layer will form a database to store the metadata required for the system

Definitions, Acronyms, Abbreviations

- PDF - Portable Document Format
- PHP-PHP Hypertext Pre-processor
- RAM-Random Access Memory
- MySQL- My Structured query language
- XAMPP – X *(any of four different operating systems)*, Apache, MySQL, PHP and Perl
- PC -Personal Computer
- OS -Operating System

References

S.E.N, 2013, Group #2, Feasibility Report_v1.0
S.E.N, 2013, Group #2, Project Proposal_v2.0
S.E.N, 2013, Group #2, Project Plan_v1.0
S.E.N, 2013, Group #2, Survey
IEEE Guide to Software Design Document

Design Considerations

Assumptions

- English is the only language used in GUI.
- Every user should be comfortable using computer.
- There exists sufficient time to complete the project.

Constraints

- Some team members need to be trained in the programming languages and the environments chosen.
- Scheduling conflicts between the clients and team members may lead to difficulties

Goals and Guidelines

- We plan to follow the KISS principle ("KEEP IT SIMPLE STUPID").
- Our goal will be to make the portal as user-friendly as possible.
- We will try to make the interface as 'catchy' as possible.

Development Method

As we have been following the evolutionary incremental model for the project, our design phase will comprehend to the same development strategy. Reviewing will be carried out after every step during the design and subsequent implementation.

System Architecture

This part of the document contains the design details regarding the system.

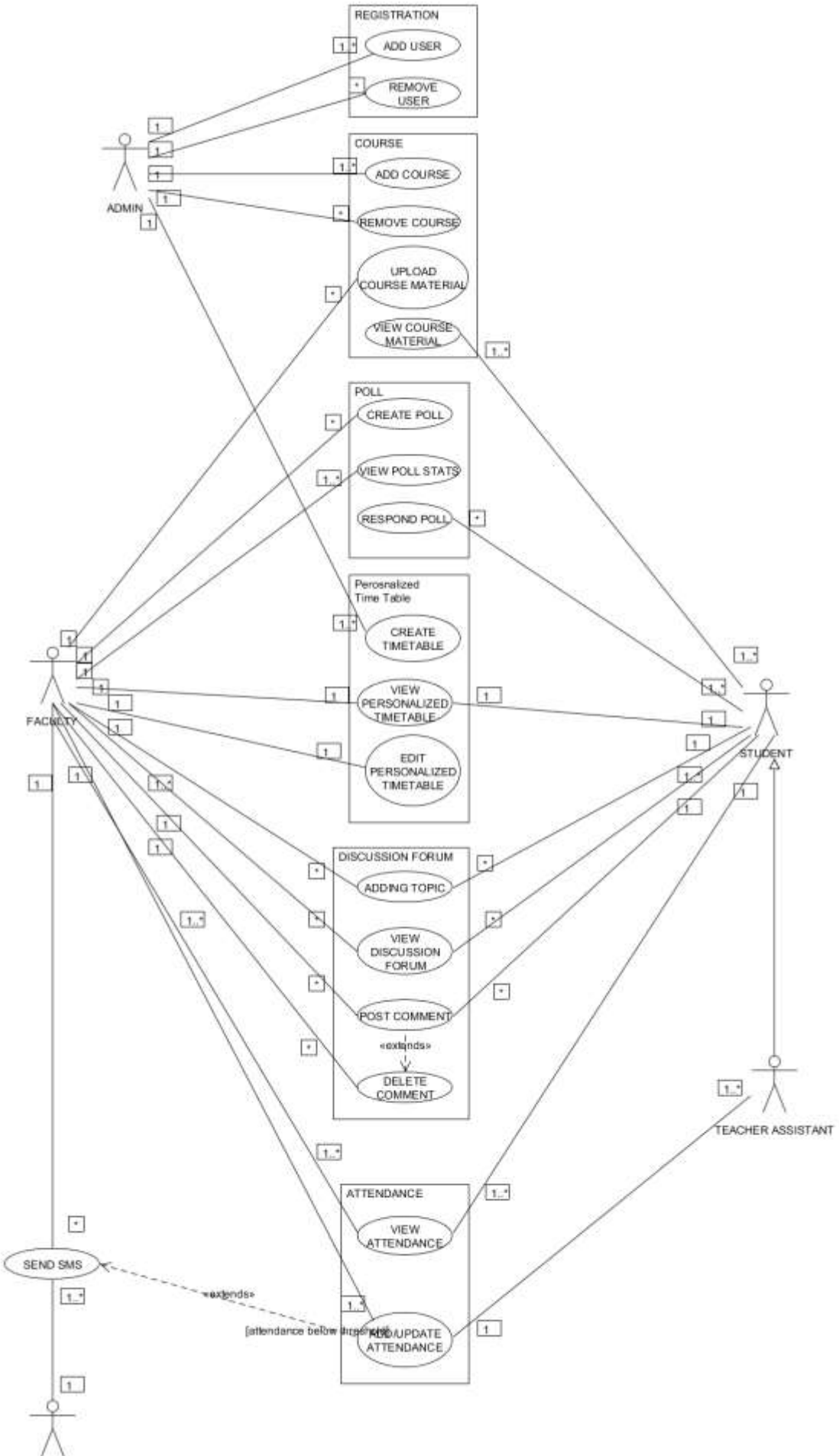
Section 3.1: Architectural Design

Section 3.2: UML Class diagram, UML sequence diagram.

Architectural Design:

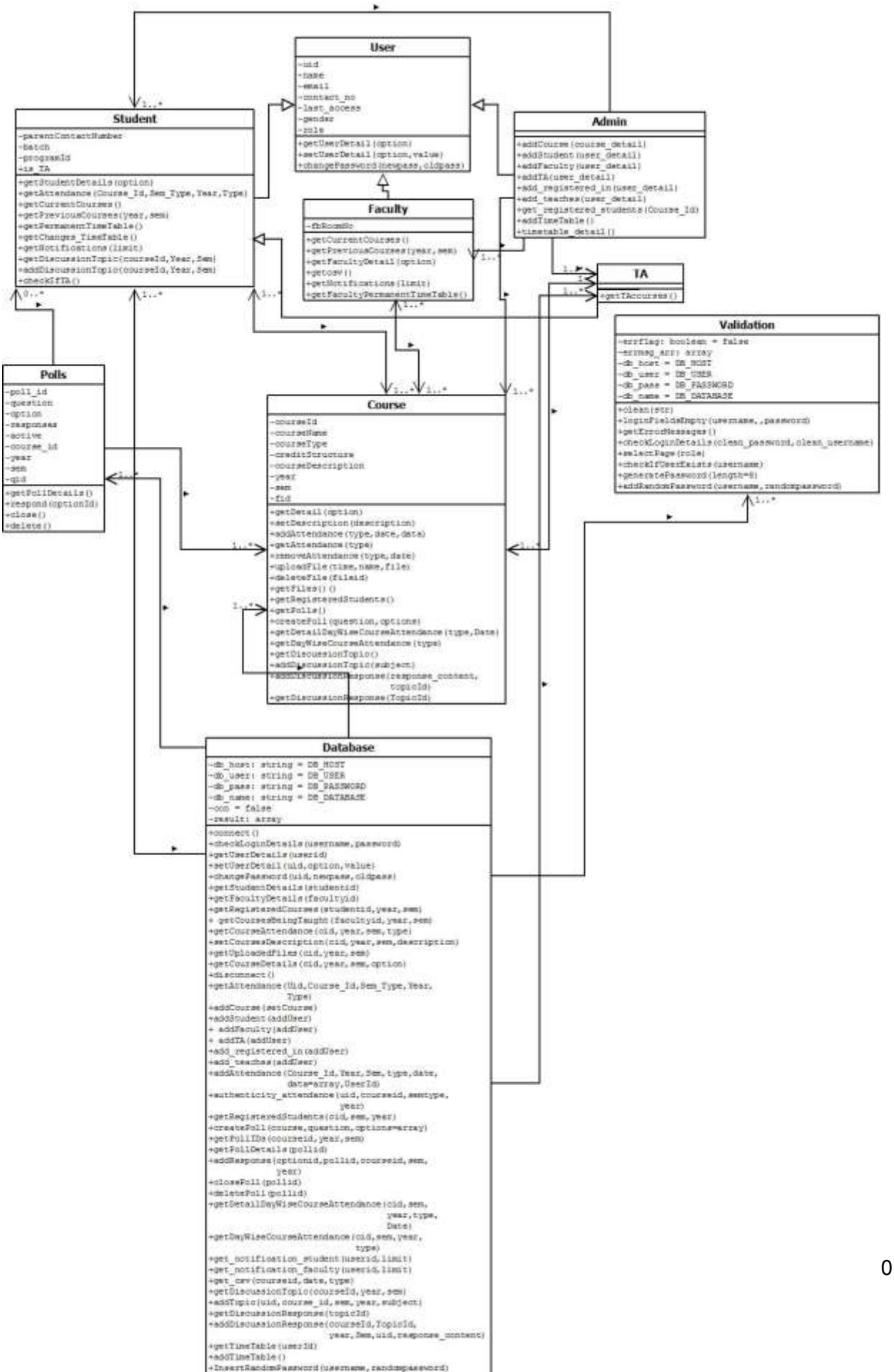
The whole system has been divided according to the roles of the users namely

- Student
- Faculty
- Admin
- Teacher Assistant


REGISTRATION
ADD USER
REMOVE USER
ADMIN
COURSE
ADD COURSE
REMOVE COURSE
UPLOAD COURSE MATERIAL
VIEW COURSE MATERIAL
POLL
CREATE POLL
VIEW POLL STATS
RESPOND POLL
Perosnalized Time Table
CREATE TIMETABLE
VIEW PERSONALIZED TIMETABLE
EDIT PERSONALIZED TIMETABLE
FACULTY
STUDENT
DISCUSSION FORUM
ADDING TOPIC
VIEW DISCUSSION FORUM
POST COMMENT
«extends»
DELETE COMMENT
TEACHER ASSISTANT
ATTENDANCE
VIEW ATTENDANCE
SEND SMS
«extends»
[attendance below threshold]
ADD/UPDATE ATTENDANCE

System Component Diagrams

UML Class Diagram


User
-uid
-name
-email
-contact_no
-last_access
-gender
-role
+getUserDetail(option)
+setUserDetail(option,value)
+changePassword(newpass,oldpass)
Student
-parentContactNumber
-batch
-programId
-is_TA
+getStudentDetails(option)
+getAttendance(Course_Id,Sem_Type,Year,Type)
+getCurrentCourses()
+getPreviousCourses(year,sem)
+getPermanentTimeTable()
+getChanges_TimeTable()
+getNotifications(limit)
+getDiscussionTopic(courseId,Year,Sem)
+addDiscussionTopic(courseId,Year,Sem)
+checkIfTA()
Faculty
-fbRoomNo
+getCurrentCourses()
+getPreviousCourses(year,sem)
+getFacultyDetail(option)
+getosv()
+getNotifications(limit)
+getFacultyPermanentTimeTable()
Admin
+addCourse(course_detail)
+addStudent(user_detail)
+addFaculty(user_detail)
+addTA(user_detail)
+add_registered_in(user_detail)
+add_teaches(user_detail)
+get_registered_students(Course_Id)
+addTimeTable()
+timetable_detail()
TA
+getTAccourses()
Validation
-errflag: boolean = false
-errmsg_arr: array
-db_host = DB_HOST
-db_user = DB_USER
-db_pass = DB_PASSWORD
-db_name = DB_DATABASE
+clean(str)
+loginFieldsEmpty(username, ,password)
+getErrorMessages()
+checkLoginDetails(clean_password,clean_username)
+selectPage(role)
+checkIfUserExists(username)
+generatePassword(length=8)
+addRandomPassword(username,randompassword)
Polls
-poll_id
-question
-option
-responses
-active
-course_id
-year
-sem
-qid
+getPollDetails()
+respond(optionId)
+close()
+delete()
Course
-courseId
-courseName
-courseType
-creditStructure
-courseDescription
-year
-sem
-fid
+getDetail(option)
+setDescription(description)
+addAttendance(type,date,data)
+getAttendance(type)
+removeAttendance(type,date)
+uploadFile(time,name,file)
+deleteFile(fileid)
+getFiles()()
+getRegisteredStudents()
+getPolls()
+createPoll(question,options)
+getDetailDayWiseCourseAttendance(type,Date)
+getDayWiseCourseAttendance(type)
+getDiscussionTopic()
+addDiscussionTopic(subject)
+addDiscussionResponse(response_content,
topicId)
+getDiscussionResponse(TopicId)
Database
-db_host: string = DB_HOST
-db_user: string = DB_USER
-db_pass: string = DB_PASSWORD
-db_name: string = DB_DATABASE
-con = false
-result: array
+connect()
+checkLoginDetails(username,password)
+getUserDetails(userid)
+setUserDetail(uid,option,value)
+changePassword(uid,newpass,oldpass)
+getStudentDetails(studentid)
+getFacultyDetails(facultyid)
+getRegisteredCourses(studentid,year,sem)
+ getCoursesBeingTaught(facultyid,year,sem)
+getCourseAttendance(cid,year,sem,type)
+setCoursesDescription(cid,year,sem,description)
+getUploadedFiles(cid,year,sem)
+getCourseDetails(cid,year,sem,option)
+disconnect()
+getAttendance(Uid,Course_Id,Sem_Type,Year,
Type)
+addCourse(setCourse)
+addStudent(addUser)
+ addFaculty(addUser)
+ addTA(addUser)
+add_registered_in(addUser)
+add_teaches(addUser)
+addAttendance(Course_Id,Year,Sem,type,data,
date=array,UserId)
+authenticity_attendance(uid,courseid,semtype,
year)
+getRegisteredStudents(cid,sem,year)
+createPoll(course,question,options=array)
+getPollIDs(courseid,year,sem)
+getPollDetails(pollid)
+addResponse(optionid,pollid,courseid,sem,
year)
+closePoll(pollid)
+deletePoll(pollid)
+getDetailDayWiseCourseAttendance(cid,sem,
year,type,
Date)
+getDayWiseCourseAttendance(cid,sem,year,
type)
+get_notification_student(userid,limit)
+get_notification_faculty(userid,limit)
+get_csv(courseid,date,type)
+getDiscussionTopic(courseId,year,sem)
+addsTopic(uid,course_id,sem,year,subject)
+getDiscussionResponse(topicId)
+addDiscussionResponse(courseId,TopicId,
year,Sem,uid,response_content)
+getTimeTable(userId)
+addTimeTable()
+InsertRandomPassword(username,randompassword)

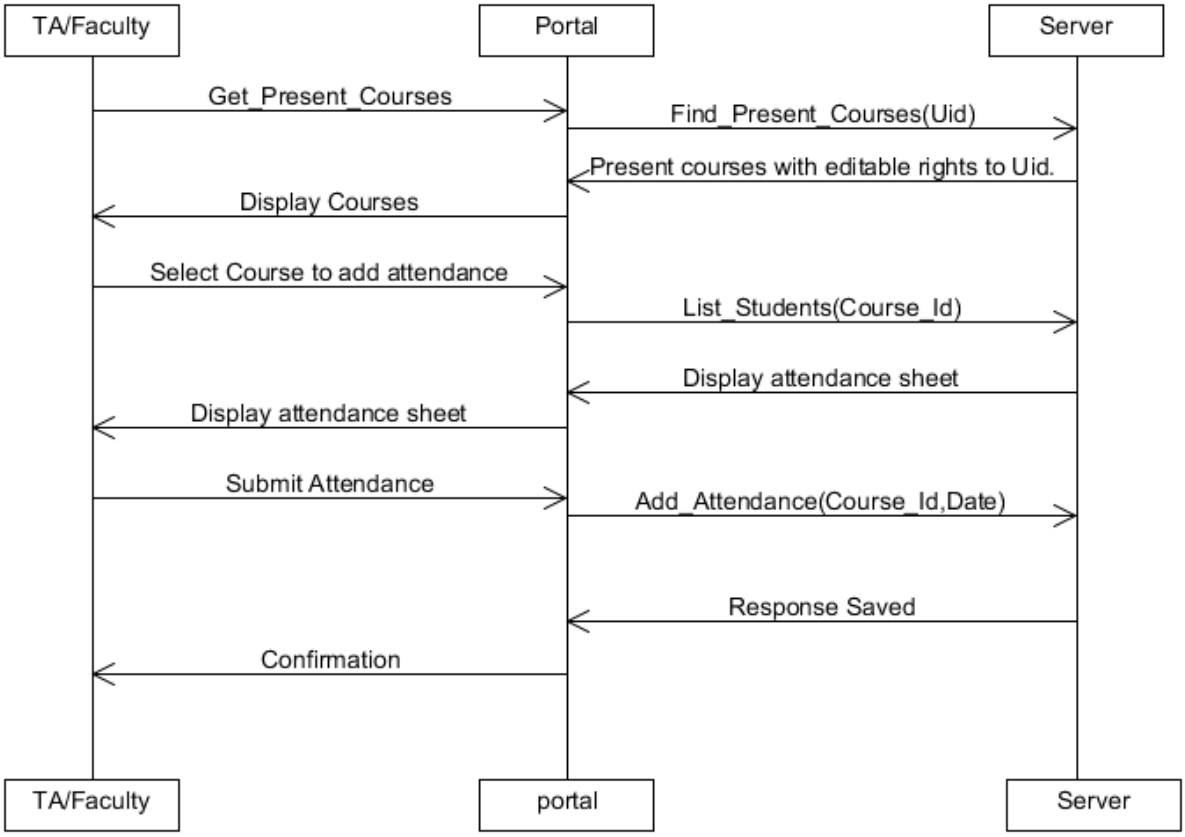
Object-oriented design is used as part of the system design process. Our design involves designing object classes and the relationship between these classes. These classes define the objects in the system and their interactions. This diagram is the class design for the application. Each class has its main attributes and methods.

ER-Diagram

UML Sequence Diagram

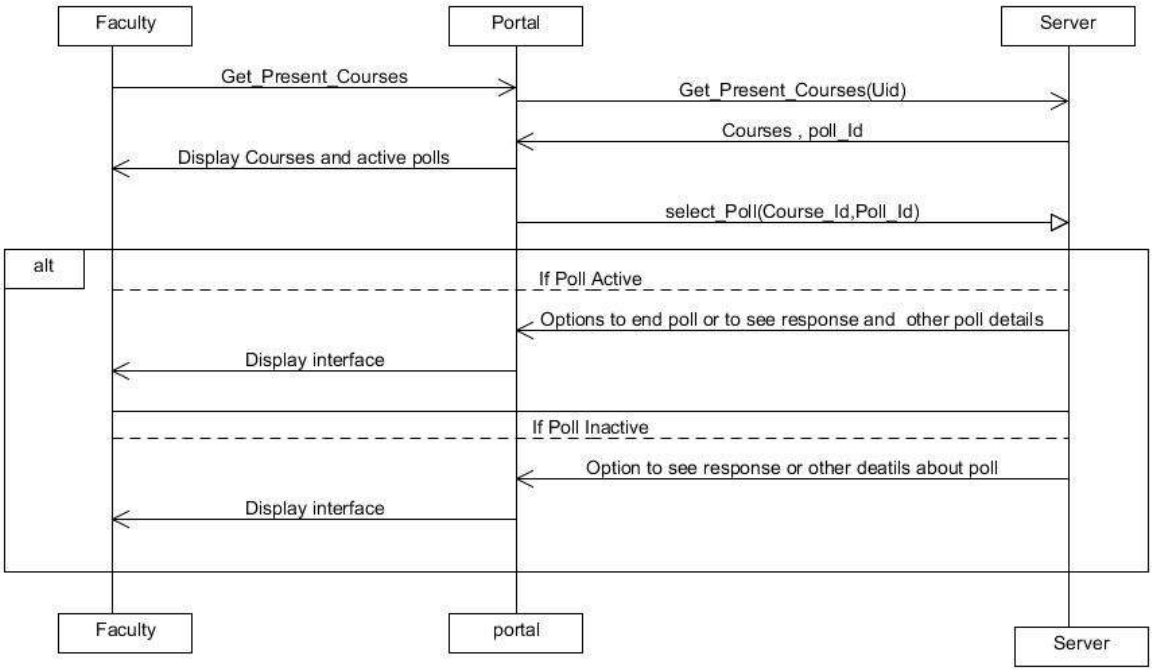
UML sequence diagram is used to model the behavior of objects. This modelling process is based on how the methods provided by the objects are used. This diagram will show the sequence of actions involved in a use-case.

Attendance:



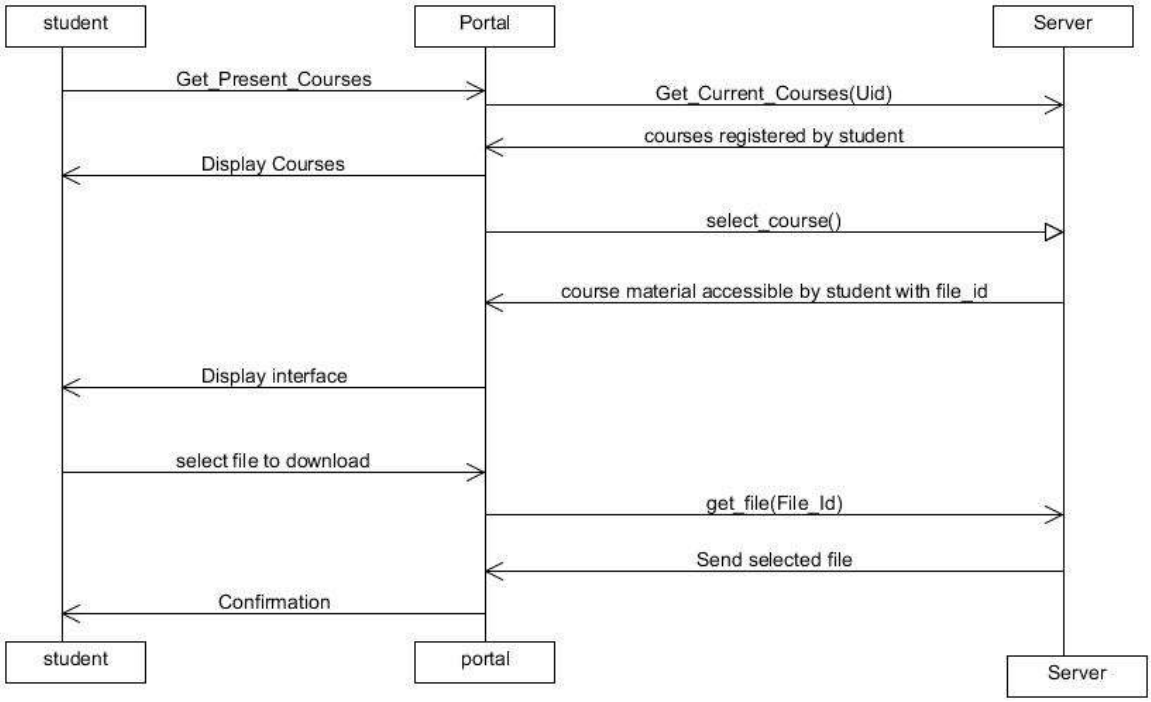
- Get_Present_Courses:
- Find_Present_Courses(Uid) –All the courses which faculty teaches/ courses of which person is TA .
- List_Students(Course_Id)- All the students enrolled in the courses.
- Add_Attendance(Course_Id,Date)- attendance of the students is added.

Faculty Poll end:



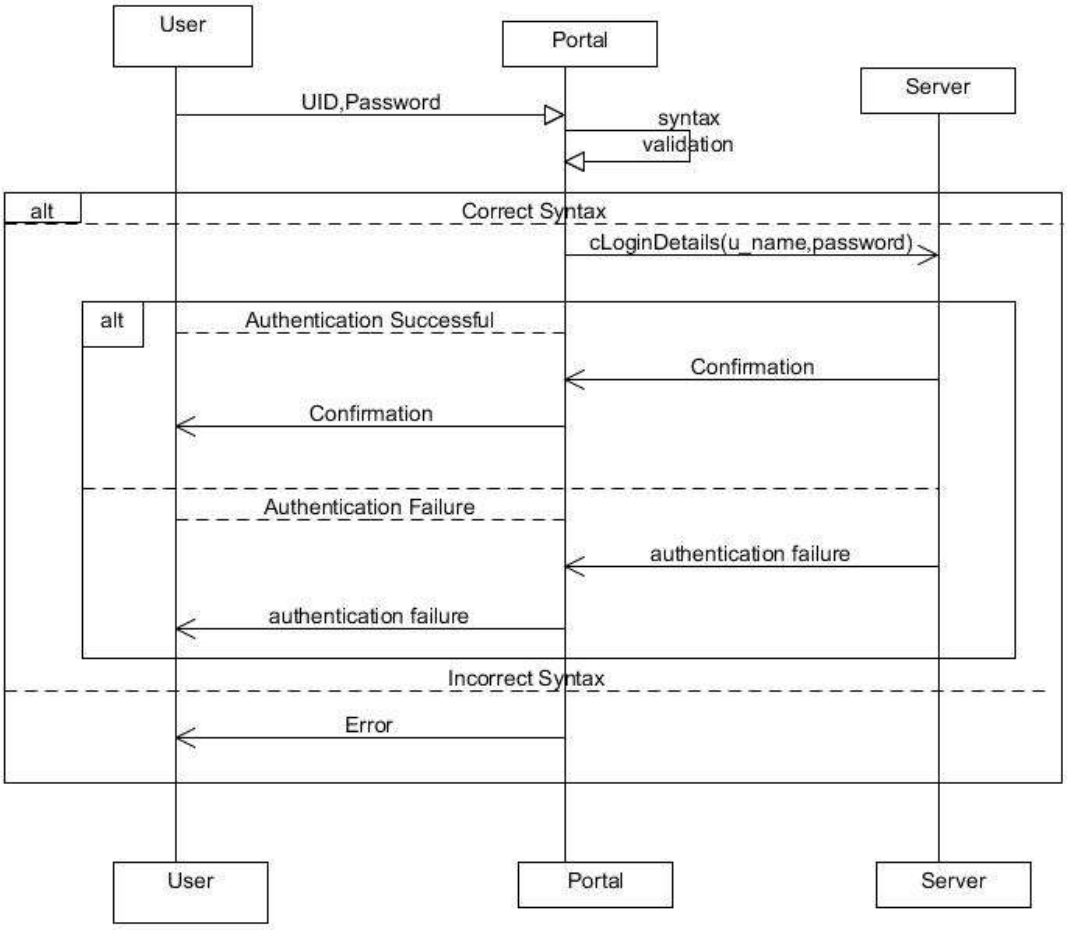
- Get_Present_Courses(Uid)- Courses which he teaches.
- select_Poll(Course_Id,Poll_Id) – To select the polls that are already created and see the responses.

Get file:



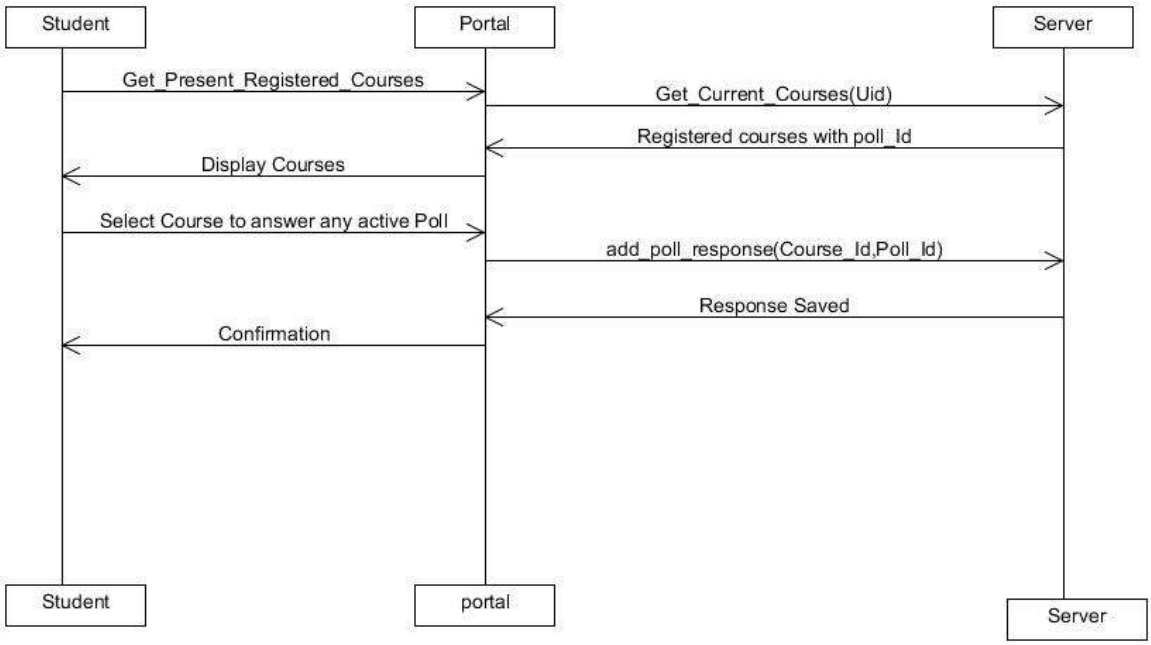
- Get_Current_Courses(Uid)- Courses in which student is presently enrolled.
- select_course() – To select a particular courses and access its material.
- get_file(File_Id) – To get a particular file uploaded on the site.

Login:



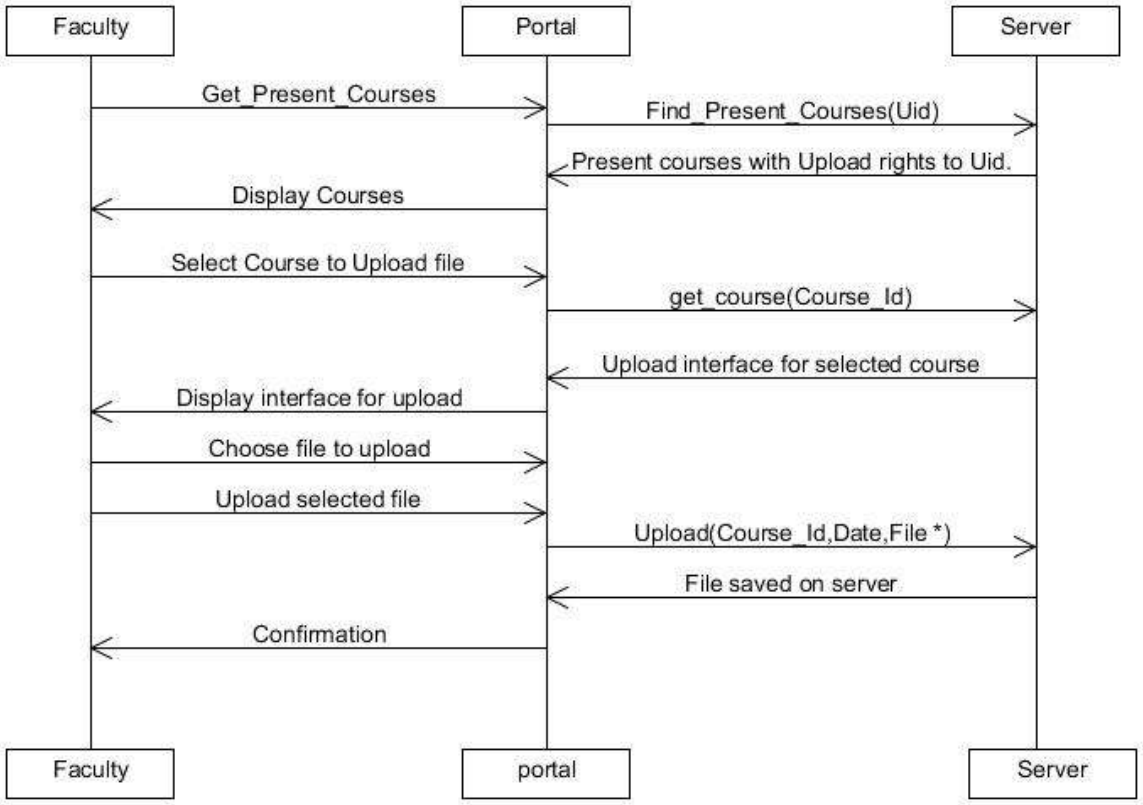
- cLoginDetails(u_name,password) – Checks the login details of the user.

Poll Student:



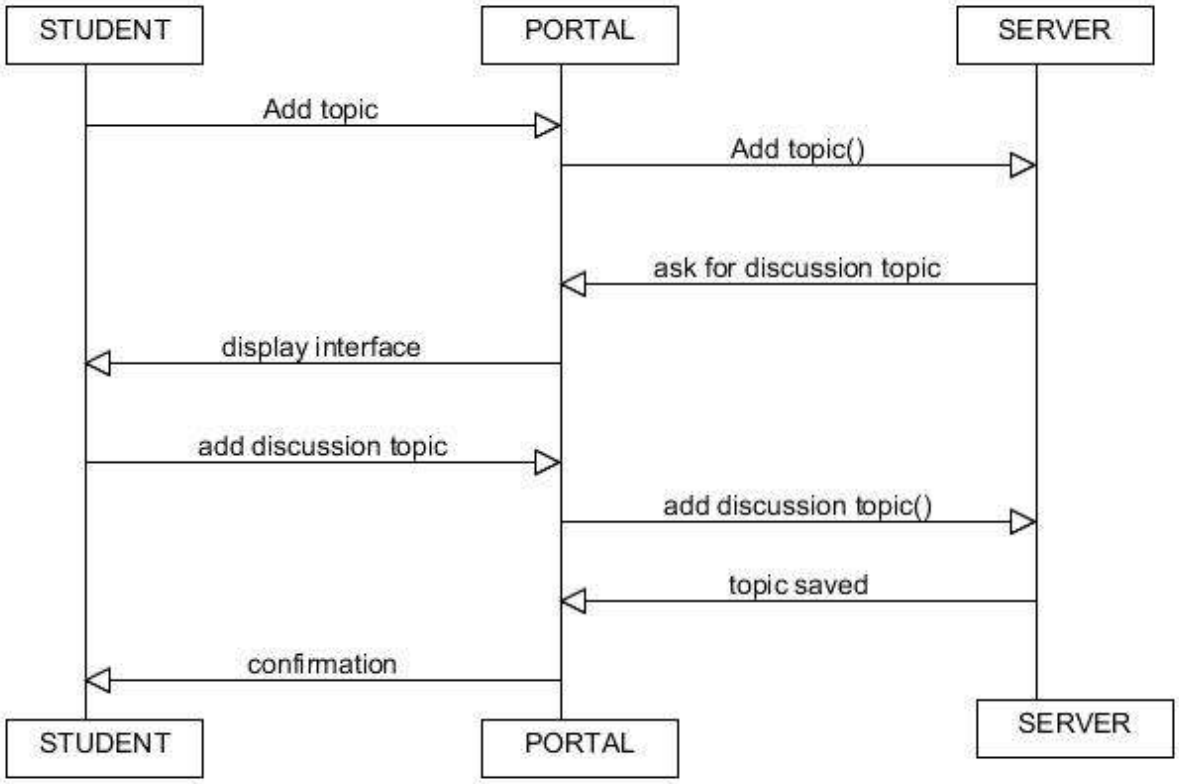
- Get_Current_Courses(Uid) – All the courses in which student is presently enrolled.
- add_poll_response(Course_Id,Poll_Id) – To save the response to the poll created.

Poll Upload:



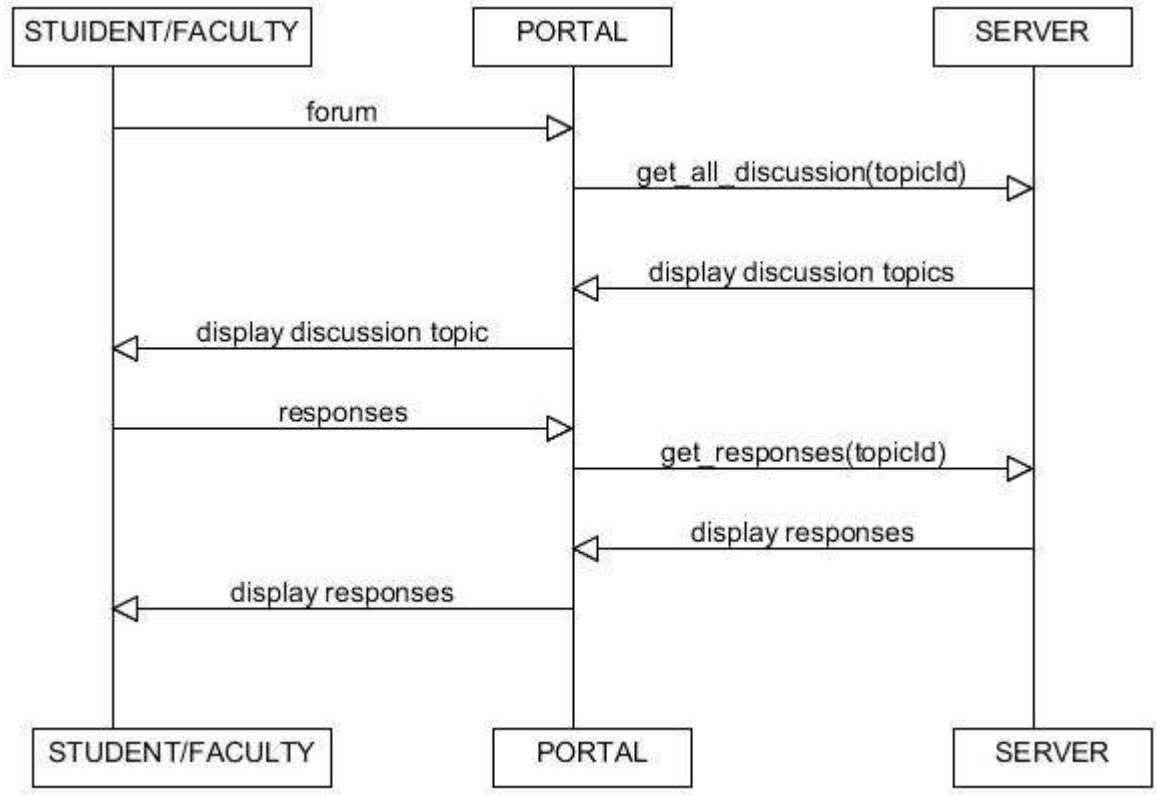
- Find_Present_Courses(Uid) – to get the list of all the courses that he teaches
- get_course(Course_Id) – to get a particular course to upload the file
- Upload (Course_Id,Date,File *) – to upload the file on the server.

AddTopic:



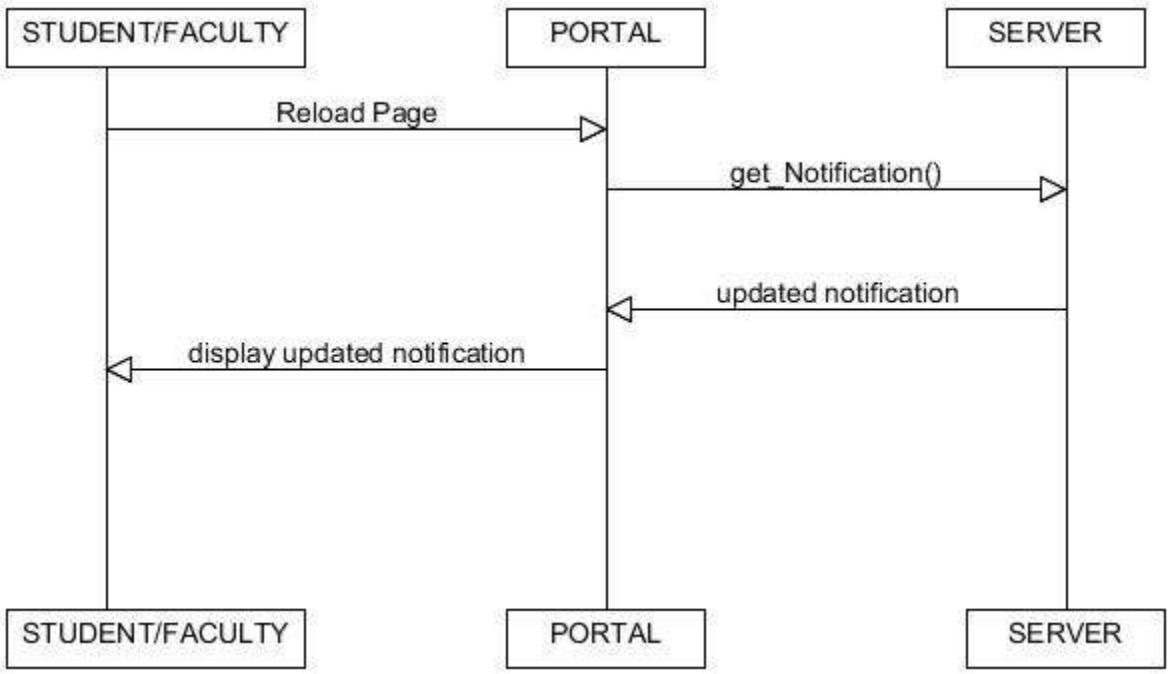
Add topic () - to add the topic for discussion.
Add discussion topic () - to save the topic to the server

Forum:



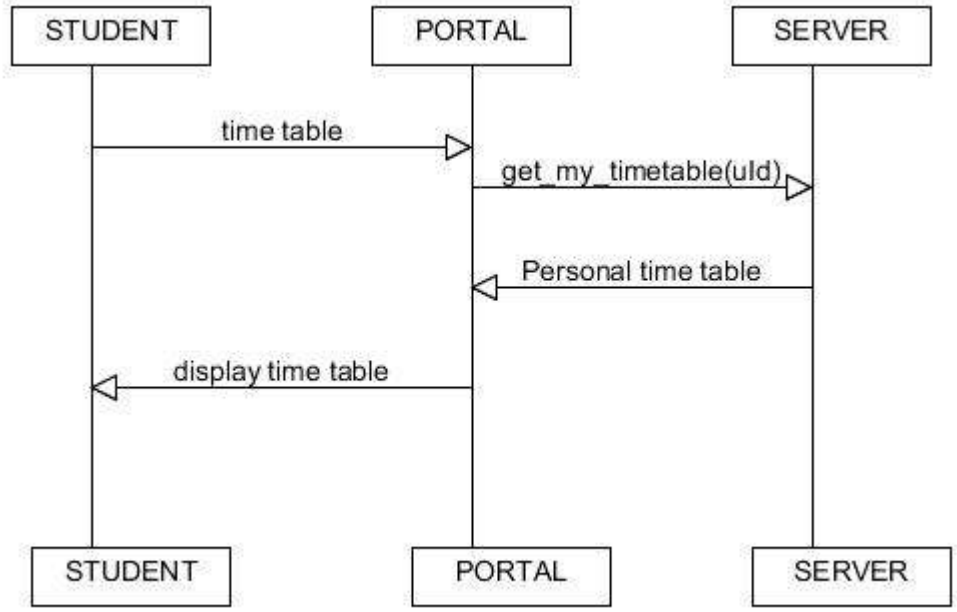
get_all_discussion(topicId)- to get all current topics up for discussion.
get_responses(topicId) - to get the responses of the the discussion topic.

Notification:



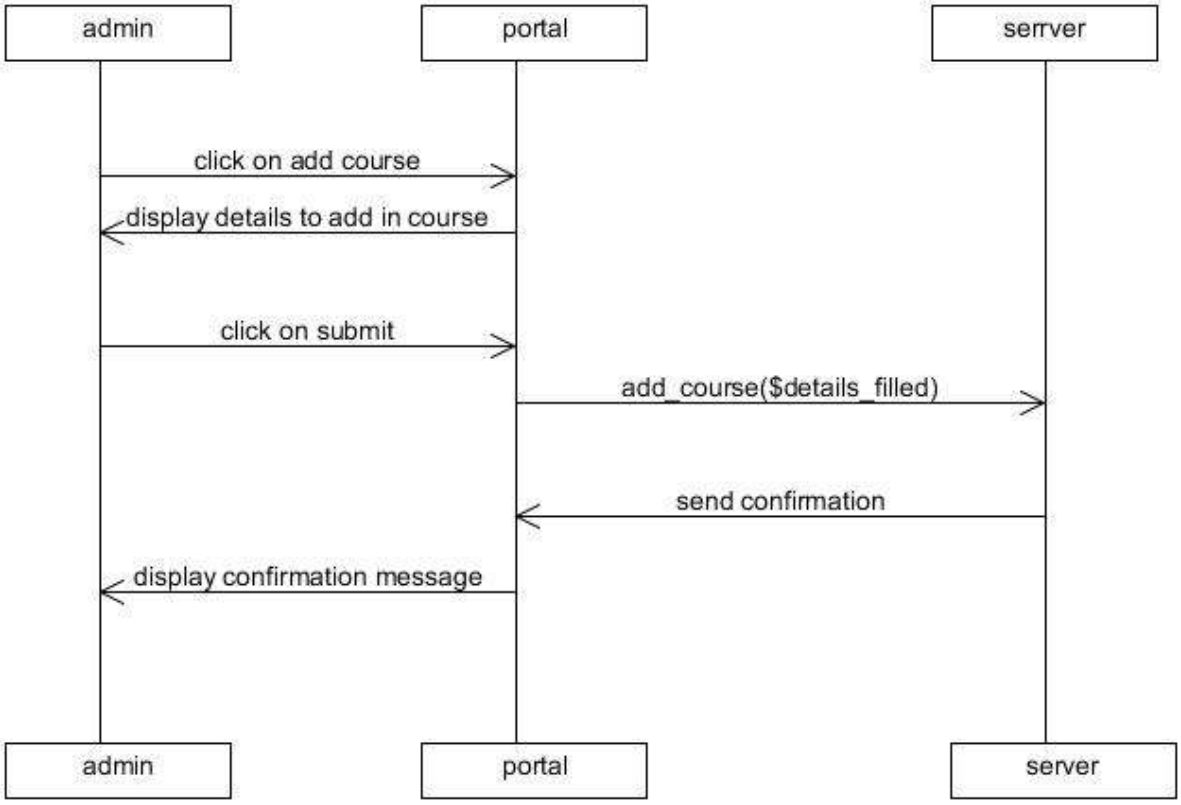
get_Notification()- to get all the notifications

Time Table:



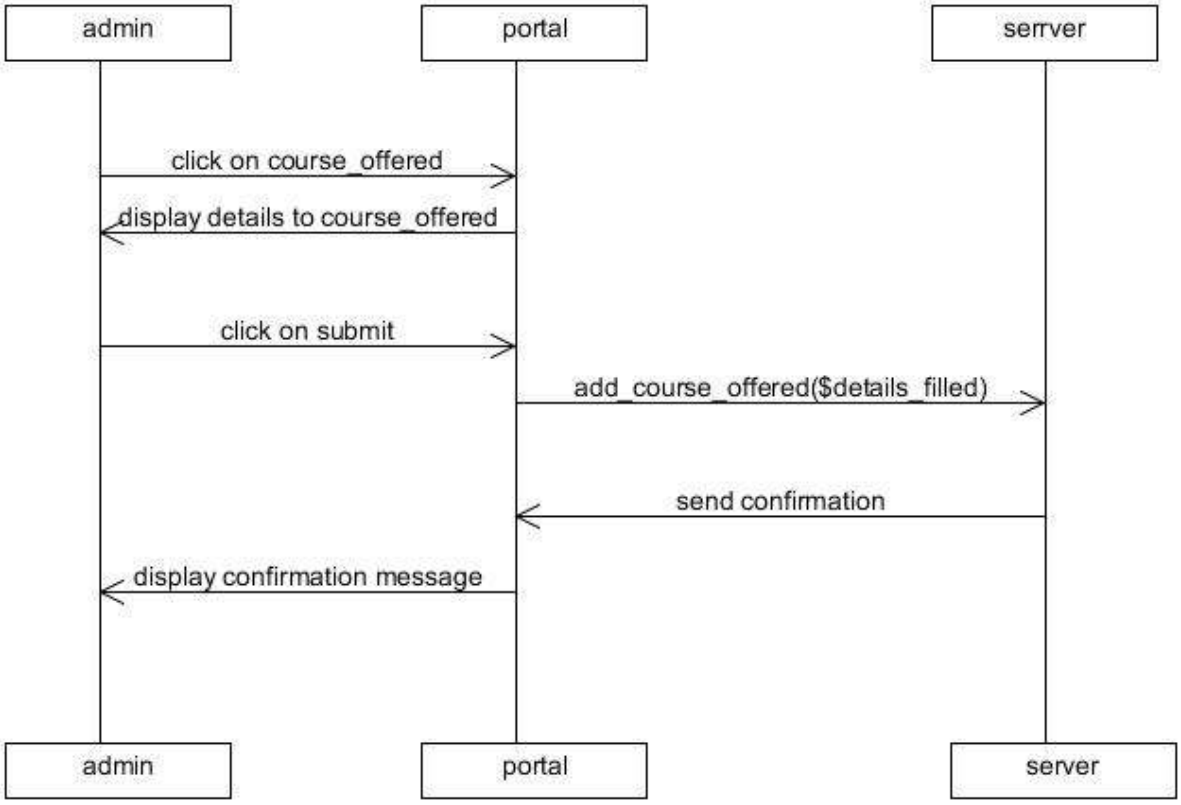
get_my_timetable(uId)- to get the personal time table of the student.

Add Course Admin:



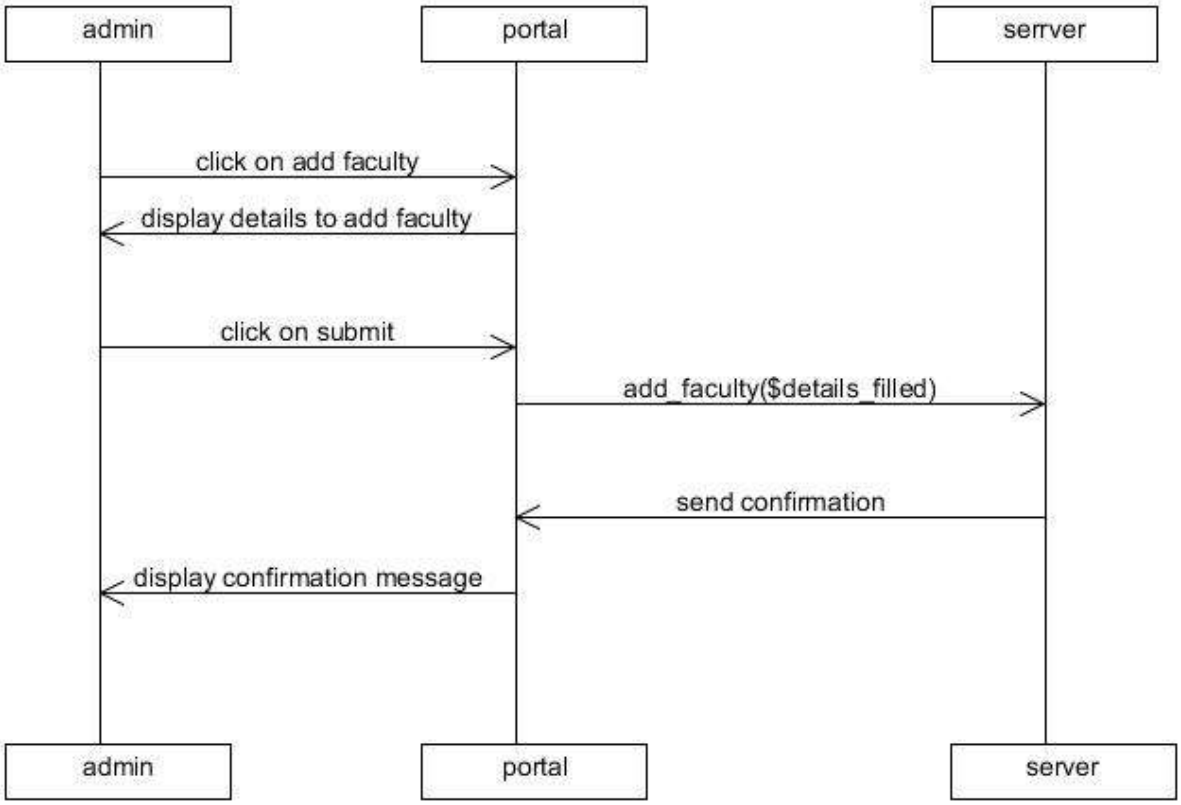
add_course($details_filled)- adds course to the database

Add Course Offered:



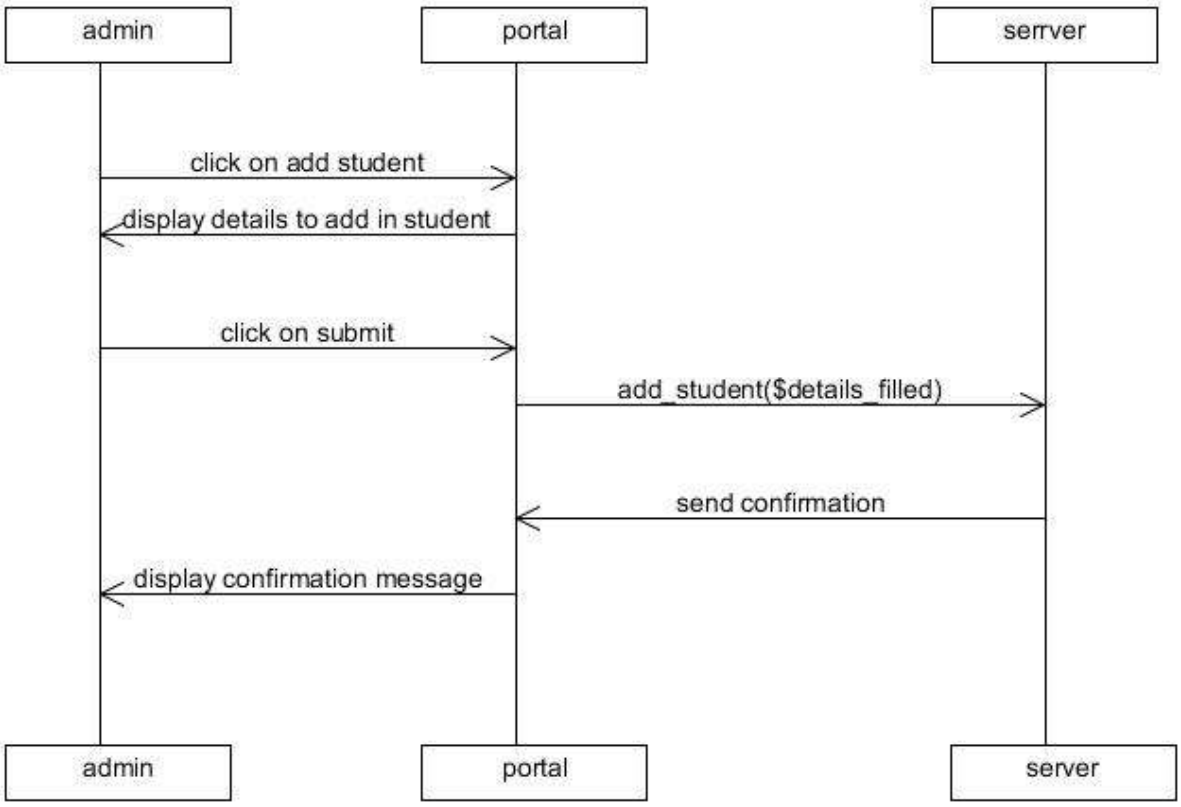
add_course_offered($details_filled)- course assigned to faculties in a particular semester.

Add Faculty Admin:



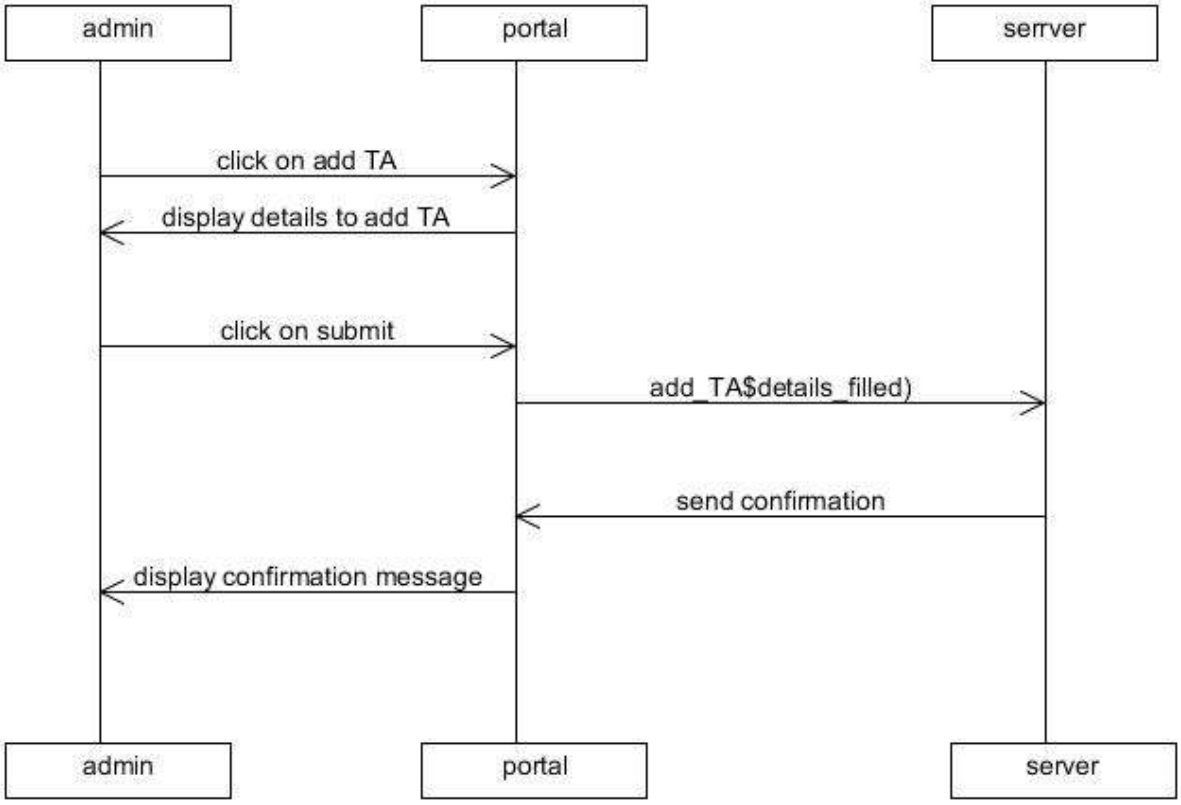
add_faculty($details_filled)- adds new faculty user to the database.

Add Student Admin:



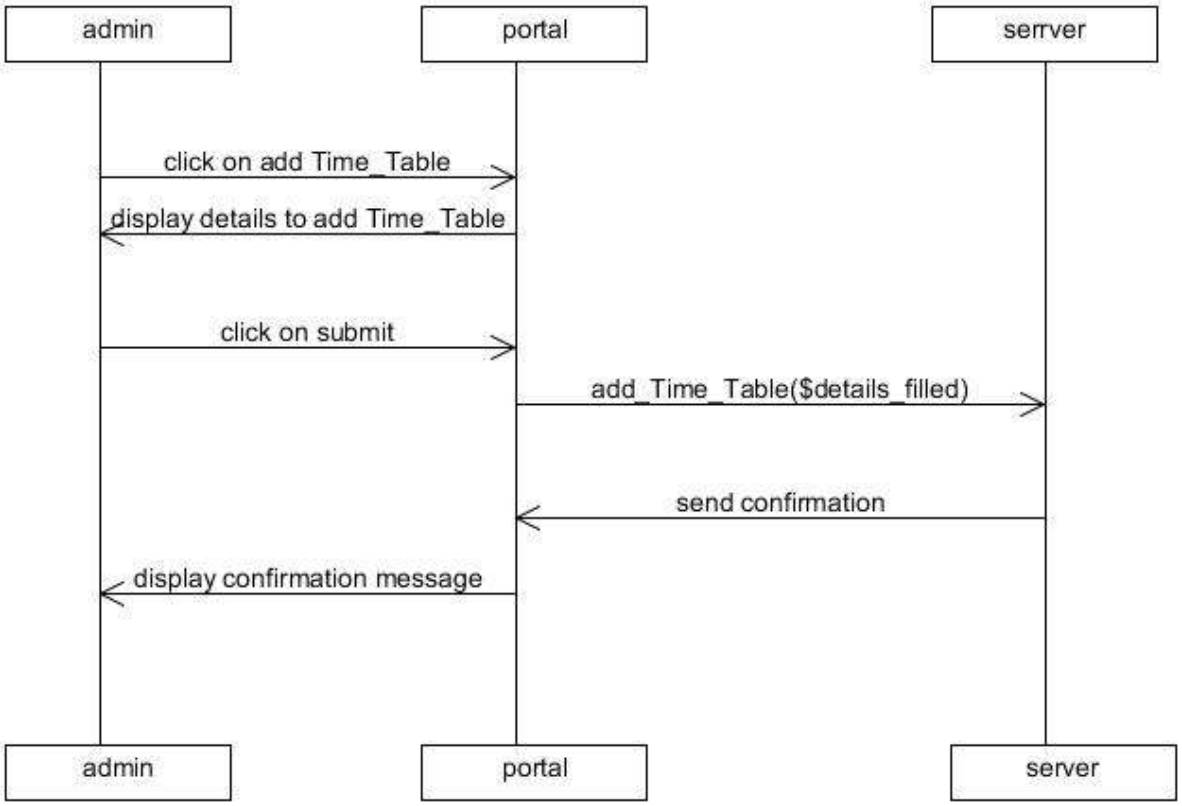
add_student($details_filled)- to add a new student to database

Add TA Admin:



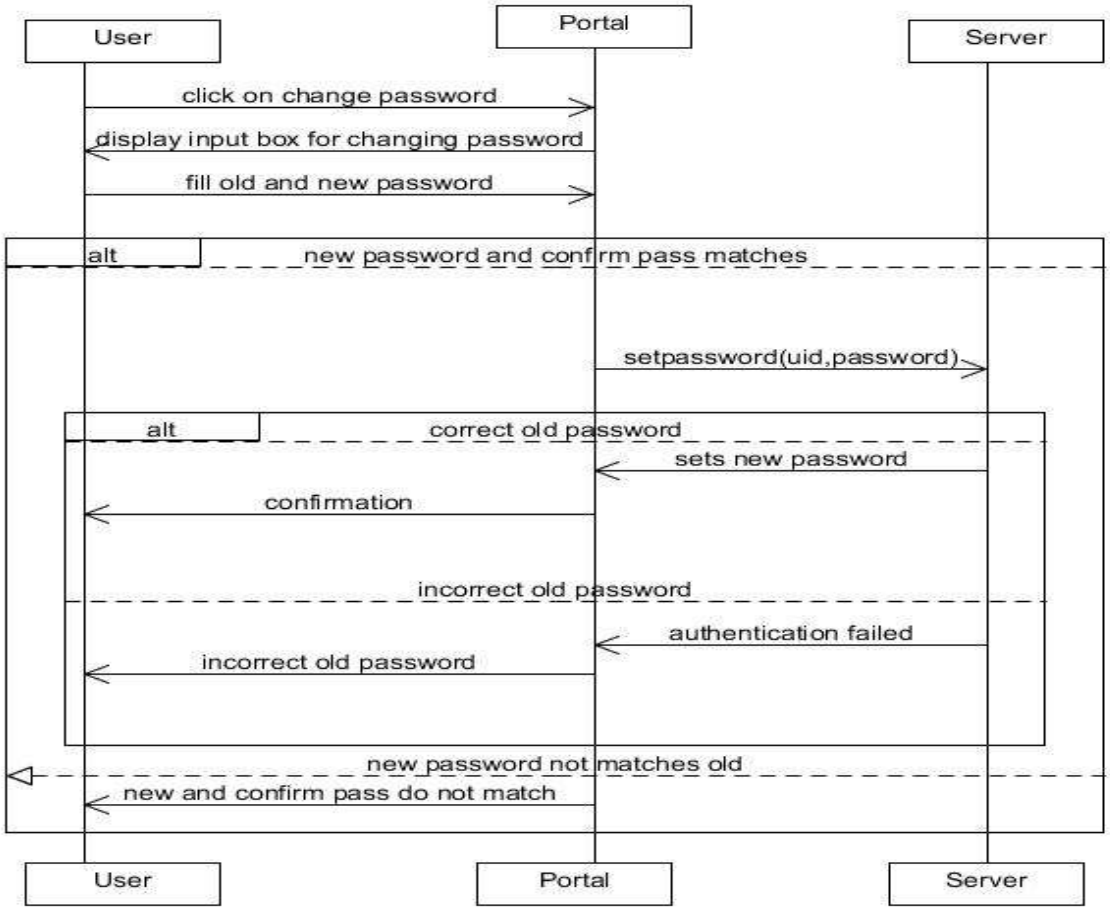
add_TA($details_filled)- to add a new TA to a pre-existing course.

Add Time Table



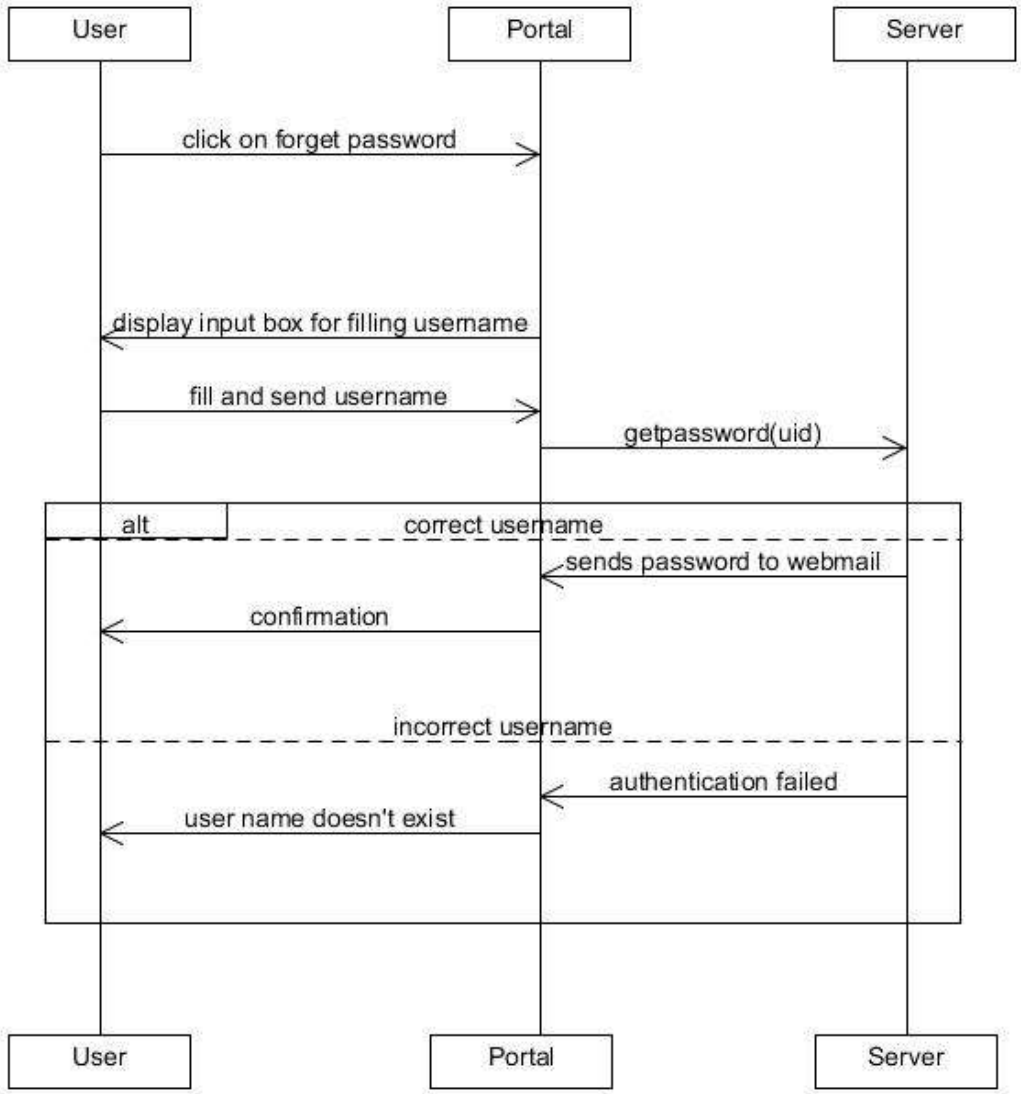
add_Time_Table($details_filled)- to add personalized time table

Change Password:



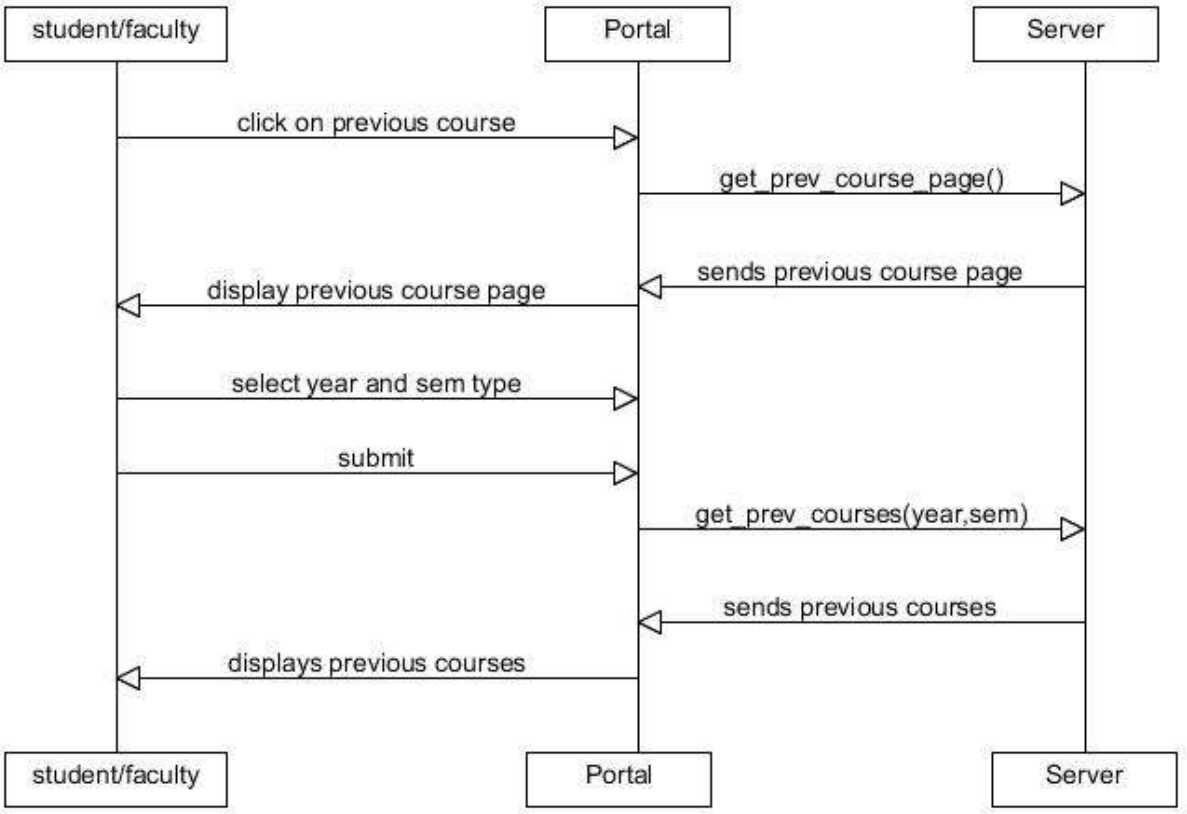
setpassword(uid,password)- sends new password to server and if authentication passed then new password is set

Forget Password:



Getpassword(uid)- to get password for the requested username.

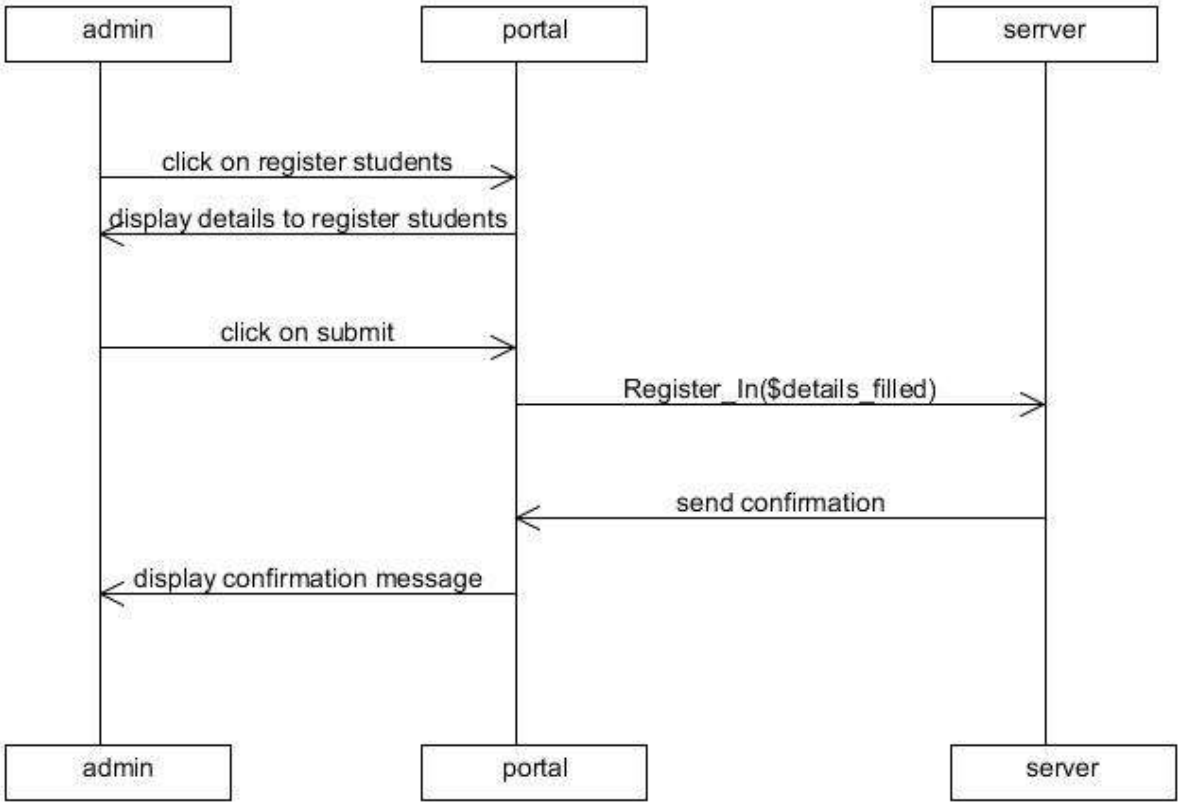
Previous Course:



get_prev_course_page : directs you to a form where required details can be filled

get_prev_courses- displays courses offered in previous semesters

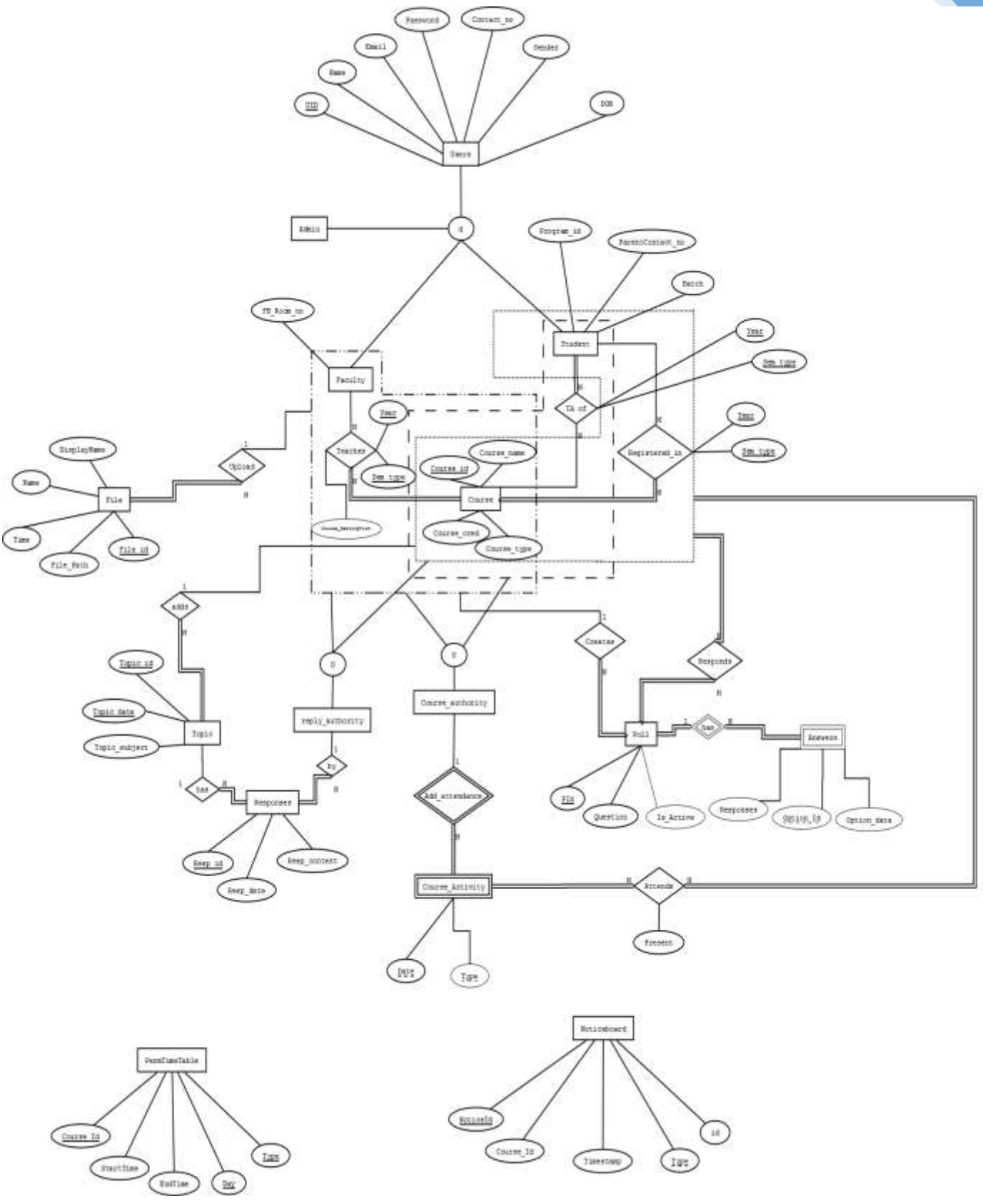
Register in Admin:



Register_In($details_filled)- to register a new student to database

Data Description

When the Interface is run for the first time, a SQL database is created. The meta-data of the file is read and its entry is made in the appropriate tables. The following is the ER diagram of the database which depicts the relationship of the tables –


Password
Contact_no
Email
Gender
Name
VID
DOB
Users
Admin
Program_id
ParentContact_no
Batch
FD_Room_no
Student
Year
Sem_type
Faculty
TA of
Year
Teaches
Registered_in
Year
Sem_type
Course_name
Course_id
Sem_type
Course
DisplayName
Upload
Name
File
Time
File_id
File_Path
Course_cred
Course_type
adds
Creates
Responds
Topic_id
Topic_date
Topic_subject
Topic
reply_authority
Course_authority
Poll
User
Answers
has
Response
Add_attendance
File
Questions
Is_Active
Responses
Option_Name
Resp_id
Resp_content
Resp_date
Course_Activity
Attends
Present
Type
Type
PermTimeTable
Course_Id
StartTime
EndTime
Day
Type
NoticeBoard
NoticeId
Course_Id
Timestamp
Type
Id

Tables:

course

Field	Type	Null	Default	Comments	MIME
Course_Id	varchar(6)	No			
Course_Name	varchar(30)	No			
Course_Type	varchar(10)	No			
Course_Credit	varchar(20)	No			

course_activity

Field	Type	Null	Default	Links to	Comments	MIME
Date	date	No				
Uid	int(11)	No		course_authority -> UID		
Sem_Type	varchar(15)	No		course_authority -> Sem_Type		
Course_Id	varchar(20)	No		course_authority -> Course_Id		
year	year(4)	No		course_authority -> Year		
Type	varchar(15)	No				

course_authority

Field	Type	Null	Default	Comments	MIME
UID	int(11)	No			
Year	year(4)	No			
Sem_Type	varchar(15)	No			
Course_Id	varchar(20)	No			

faculty

Field	Type	Null	Default	Links to	Comments	MIME
Uid	int(11)	No		users -> Uid		
FB_Room_No	varchar(5)	No				

file

Field	Type	Null	Default	Links to	Comments	MIME
File_Id	int(11)	No				
File_Path	varchar(150)	No				
Time	datetime	No				
Uid	int(11)	No		teaches -> Uid		
Course_Id	varchar(20)	No		teaches -> Course_Id		
Year	year(4)	No		teaches -> Year		
Sem_Type	varchar(15)	No		teaches -> Sem_Type		
Name	varchar(40)	No				
Display_name	varchar(60)	No				

noticeboard

Field	Type	Null	Default	Comments	MIME
NoticeId	int(11)	No			
Course_Id	varchar(15)	No			
TimeStamp	timestamp	No	CURRENT_TIMESTAMP		
Type	varchar(15)	No			
Id	varchar(50)	No			

permtimetable

Field	Type	Null	Default	Comments	MIME
Course_Id	varchar(15)	No			
StartTime	time	No			
EndTime	time	No			
Day	varchar(15)	No			
Type	varchar(15)	No			

poll

Field	Type	Null	Default	Links to	Comments	MIME
Poll_Id	int(11)	No				
Uid	int(11)	No		teaches -> Uid		
Course_Id	varchar(20)	No		teaches -> Course_Id		
Year	year(4)	No		teaches -> Year		
Sem_Type	varchar(15)	No		teaches -> Sem_Type		
Questions	varchar(100)	No				
Is_Active	int(11)	No				

registered_in

Field	Type	Null	Default	Links to	Comments	MIME
Uid	int(11)	No		student -> Uid		
Course_Id	varchar(20)	No		course -> Course_Id		
Year	year(4)	No				
Sem_Type	varchar(15)	No				

reply_authority

Field	Type	Null	Default	Comments	MIME
Uid	int(11)	No			
Year	year(4)	No			
Sem_Type	varchar(15)	No			
Course_Id	varchar(20)	No			

responds

Field	Type	Null	Default	Links to	Comments	MIME
UID	int(11)	No		registered_in -> Uid		
Course_Id	varchar(20)	No		registered_in -> Course_Id		
Sem_Type	varchar(15)	No		registered_in -> Sem_Type		
year	year(4)	No		registered_in -> Year		
Poll_Id	int(11)	No		poll -> Poll_Id		

responses

Field	Type	Null	Default	Links to	Comments	MIME
Resp_Id	int(11)	No				
Resp_Date	timestamp	No	CURRENT_TIMESTAMP			
Resp_Content	varchar(100)	No				
Uid	int(11)	No		reply_authority -> Uid		
Sem_Type	varchar(15)	No		registered_in -> Sem_Type		
year	year(4)	No		reply_authority -> Year		
Course_Id	varchar(20)	No		reply_authority -> Course_Id		
Topic_Id	int(11)	No		topic -> Topic_Id		

student

Field	Type	Null	Default	Links to	Comments	MIME
Uid	int(11)	No		users -> Uid		
Parent_Contact_No	varchar(14)	No				
Batch	int(11)	No				
Program_Id	varchar(6)	No				
Is_Ta	int(11)	No				

ta_of

Field	Type	Null	Default	Links to	Comments	MIME
Course_Id	varchar(6)	No		course -> Course_Id		
Uid	int(11)	No		student -> Uid		
Year	year(4)	No				
Sem_Type	varchar(6)	No				

teaches

Field	Type	Null	Default	Links to	Comments	MIME
Uid	int(11)	No		faculty -> Uid		
Course_Id	varchar(20)	No		course -> Course_Id		
Year	year(4)	No				
Sem_Type	varchar(15)	No				
Course_Description	varchar(1000)	No				

topic

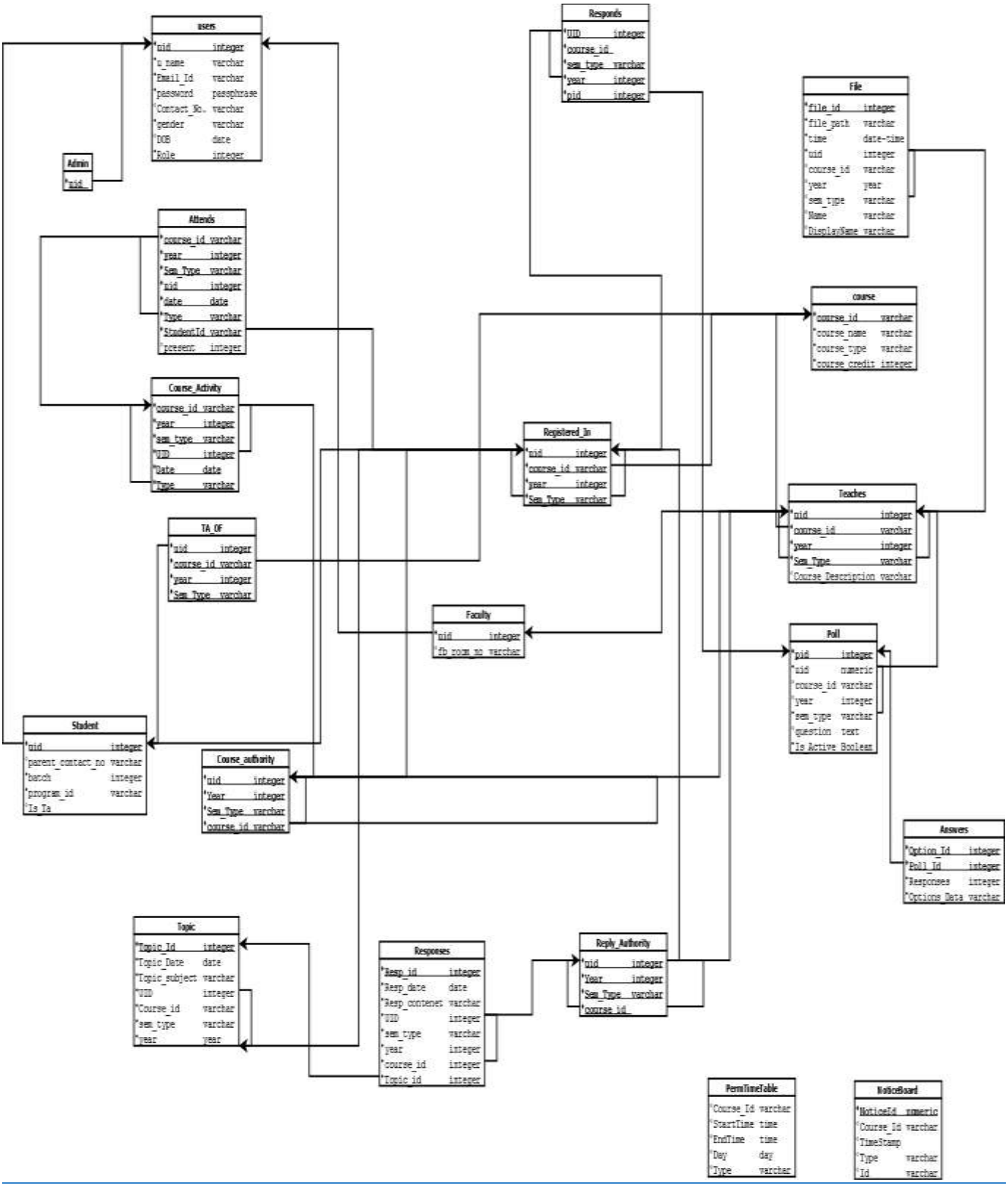
Table comments: topic

Column	Type	Null	Default	Links to	Comments	MIME
Topic_Id	int(11)	No				
Topic_Date	timestamp	No	CURRENT_TIMESTAMP			
Topic_Subject	varchar(50)	No				
Uid	int(11)	No		registered_in -> Uid		
Course_Id	varchar(20)	No		registered_in -> Course_Id		
Sem_Type	varchar(15)	No		registered_in -> Sem_Type		
Year	year(4)	No		registered_in -> Year		

users

Field	Type	Null	Default	Comments	MIME
Uid	int(11)	No			
U_Name	varchar(25)	No			
Email_Id	varchar(25)	No			
Password	varchar(50)	No			
Contact_No	varchar(14)	Yes	NULL		
Gender	varchar(6)	No			
Last_Access	datetime	Yes	NULL		
DOB	date	No			
Role	int(11)	No			

Relational Schema



Detailed Design

Activity diagram