Group 5
KSU Portal
Software Design Document

Name (s):
N/A

Date: (10/06/2015)

Table Of Contents

1. INTRODUCTION 5

1.1. PURPOSE
1.2. SCOPE
1.3. DEFINITIONS AND ACRONYMS
1.4. REFERENCE MATERIAL

2. DESIGN OVERVIEW
2.1. INTRODUCTION
2.2. SYSTEM ARCHITECTURE
2.2.1. *Architectural Design*
2.3. SYSTEM INTERFACES
2.3.1. *User Interfaces*
2.3.2. *Hardware Interface*
2.3.3. *Software Interface*
2.3.4. *Communication Interface*
2.4. CONSTRAINTS AND ASSUMPTIONS

3. SYSTEM OBJECT MODEL
3.1. INTRODUCTION
3.2. SUBSYSTEMS
3.3. SUBSYSTEM INTERFACES

4. OBJECT DESCRIPTIONS
4.1. COMPONENT DESIGN
4.1.1. *Login*
4.1.2. *Logout*
4.1.3. *Semester*
4.1.4. *Academic History*
4.1.5. *Schedule*
4.1.6. *Add/Drop Courses*
4.1.7. *Search*
4.1.8. *Calendar*
4.1.9. *Map*
4.1.10. *Faculty Information*

4.1.11. Student Database

5. HUMAN INTERFACE DESIGN

5.1. Overview of User Interface

Screen Images

6. Dynamic Model

6.1. Sequence Diagrams

6.1.1. Student Sequence Diagrams:

6.1.2. Faculty and Staff Sequence Diagrams:

6.1.3. It Admin Sequence Diagrams:

6.2. Object Collaboration Diagrams

Object Behavior Diagrams

7. DATA DESIGN

7.1. Database Description

7.2. Database Design

7.2.1. Map

7.2.2. Calendar

7.2.3. Course

7.2.4. Student

7.2.5. Faculty_Staff

7.2.6. Student Database

7.2.7. Course Database

7.2.8. Map Database

7.2.9. Calendar Database

7.3. Data Dictionary

7.3.1. Academic History

7.3.2. Add/Drop Courses

7.3.3. Calendar

7.3.4. Calendar Database

7.3.5. Course Database

7.3.6. Faculty Database

7.3.7. Faculty Directory

7.3.8. Faculty Information

7.3.9. Login

7.3.10. Logout

7.3.11. Map

7.3.12. Map Database

7.3.13. Schedule

7.3.14. Search

7.3.15. Semester
7.3.16. Student Database

8. NON-FUNCTIONAL

9. APPENDIX

Introduction

Purpose

The Purpose of this Software Design Description is to give specific detail on the functionality of KSU Portal, a mobile application, specifically for Kennesaw State University Students. This document will be providing more technical details about how the mobile application will be created, what will be involved in the creation. It will be a more in depth description from the Software Requirement Specification.

Scope

Kennesaw State University students rely on Owl Express for almost everything they need to do, pertaining to school registration, culinary services, parking, voting and etc. Most students are always on the move and in this day and age, we rely on our mobile deceives more than our computers. The downside to the current state of owl express is that, there is no mobile friendly version provided. When Owl Express is accessed on a mobile device, the website still does not convert to a mobile version like most website.

Definitions and Acronyms

- Owl Express - The Student login portal for Kennesaw State University students.
- KSU - Kennesaw State University
- h/m/s - hours / minutes / seconds
- CRN - course number
- Student - person attending Kennesaw State University.
- Faculty- person who works at Kennesaw State University

Reference Material

- AWS | Amazon Elastic Compute Cloud (EC2) - Scalable Cloud Hosting. (n.d.). Retrieved October 6, 2015.
- Propel API. (n.d.). Retrieved October 6, 2015.
- Start Developing iOS Apps (Swift): Jump Right In. (n.d.). Retrieved October 6, 2015.

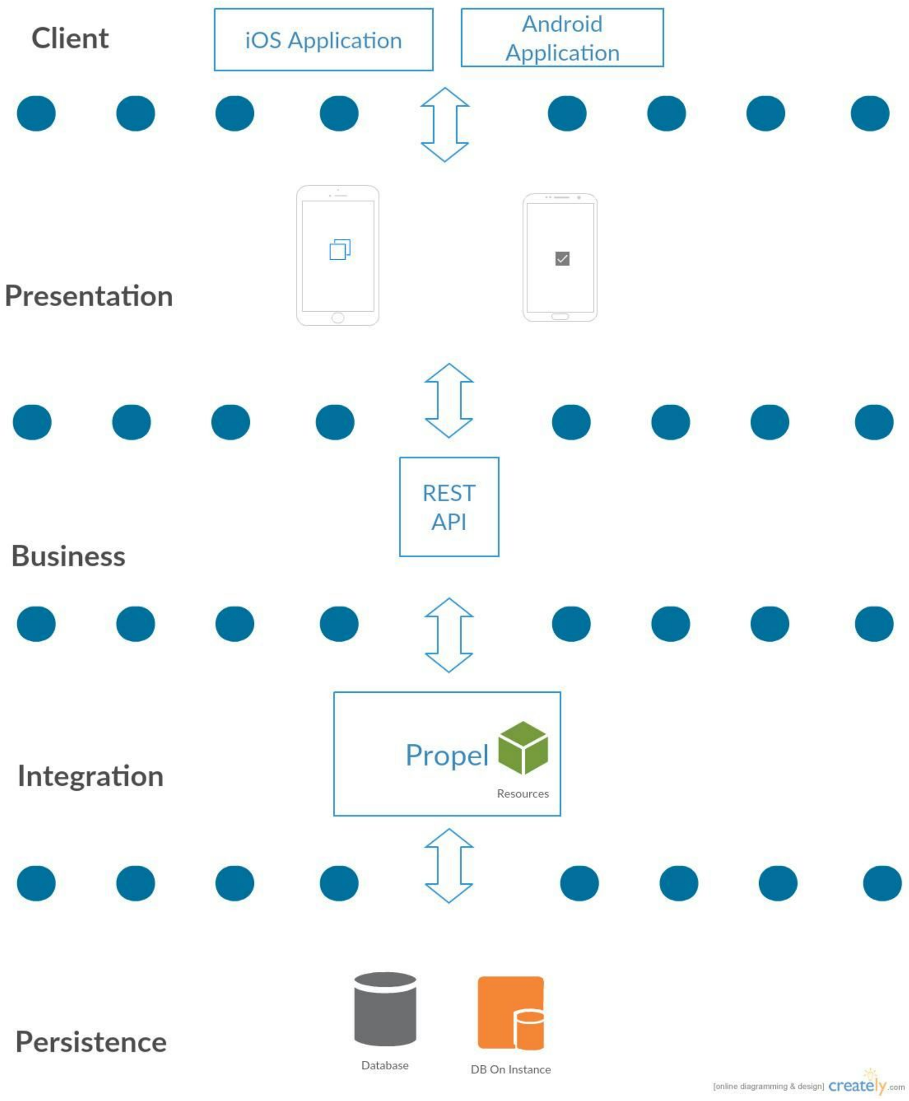
Design Overview

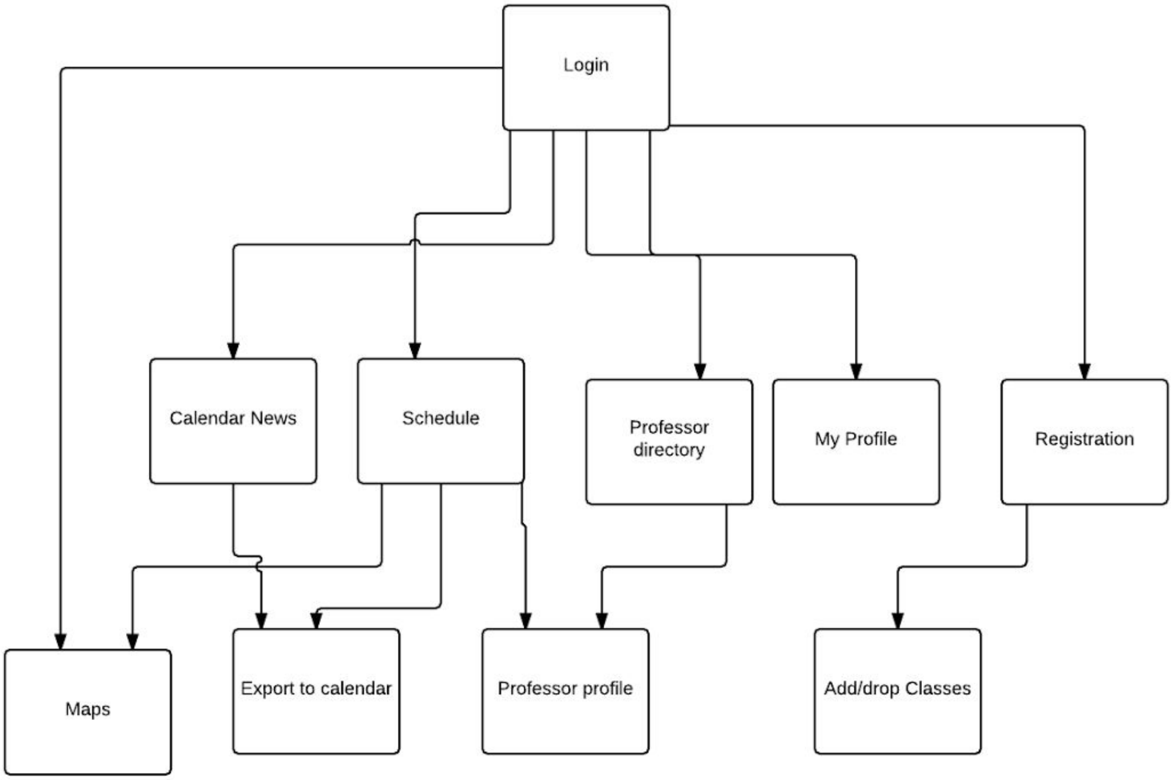
Introduction

The Design Overview gives a brief overview of the design of the system. The System Architecture gives an overview of the entire view of the system from the Client to the Database layer of the system. The system interfaces explains the different interfaces for the system including user, software, hardware, and communication. Finally, the constraint and assumptions explains the things we assume to use the system and the necessary feature needed for the system.

System Architecture

1.1.1. Architectural Design





In order to access the features of the application, the user must login using his or her password. Then the user is able to access the 4 main features, Professor directory, Current Schedule, My Profile, and Registration. The 2 extra features are Maps and Calendar News. In order to view a Professor's profile information, the user must select that professor either from the professor directory feature by searching for the professor or by the current schedule feature by selecting a link for the professor that's instructing one of the user's classes. Another feature of the KSU Portal application is exporting the user's current schedule to a google calendar account. This is only achieved by choosing that option in the current schedule screen. Another instance of exporting to the user's google calendar is through the Calendar News. In order for a user to know where his/her class is, the user would select the location name/number from the current schedule to link over t the maps and highlight where that class is.

System Interfaces

1.1.2. User Interfaces

The first time the user opens the application, brings them to a login screen just like figure 1.1. The student is prompted by a username and password which is directly related to their Internet identification and password given by KSU. After logging into the application, the user will be brought up to figure 1.2 which is displaying the tabs within the application. Looking at it closely, the tabs in which the students can select from my profile, current schedule, registration, and professor directory.

- In figure 1.3, this should be my profile tab in the application. This view should display the name of the student, with the KSU number, address, phone number, age, and campus email. This should allow the user to make sure that this their account, and also validate that their information in this section is all correct.
- In figure 1.4, this view is looking at the faculty and staff directory acquired from the faculty and staff database. This should show all the professors and should allow the user to click on the professor to get more information about them.
- In figure 1.5, this view is looking at the course Calculus I that was acquired from the course database. It gives a lot of information such as class hours, professor, days, and many more information.

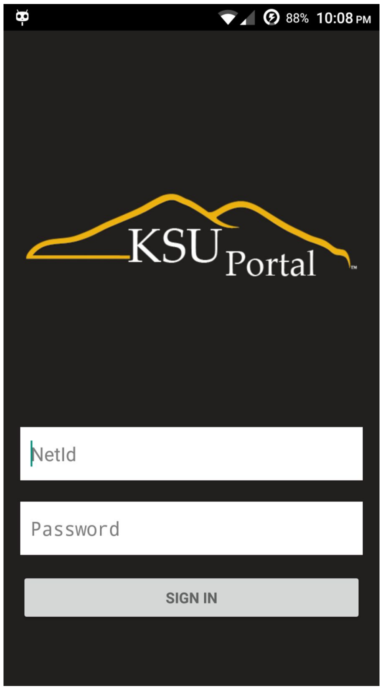


Figure 1.1

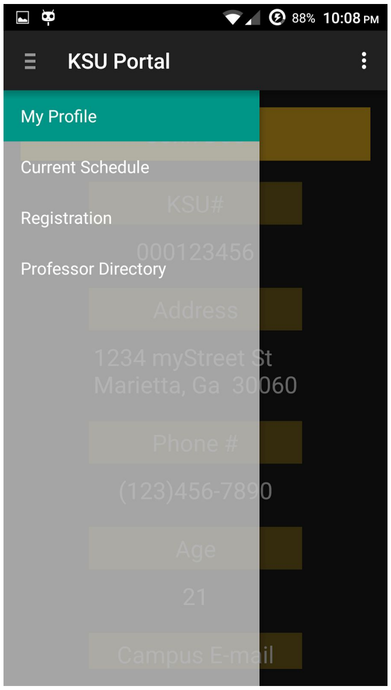
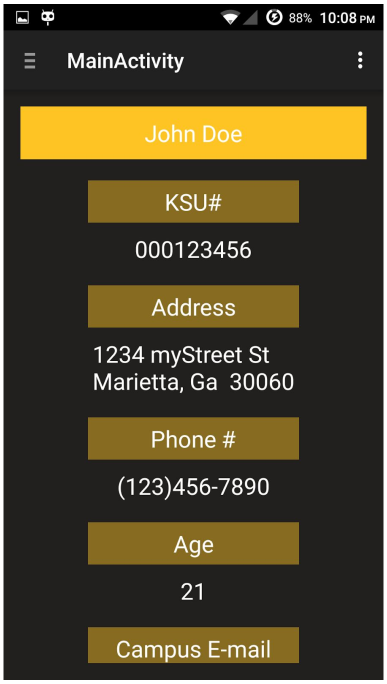


Figure 1.2



John Doe

KSU#

000123456

Address

1234 myStreet St
Marietta, Ga 30060

Phone #

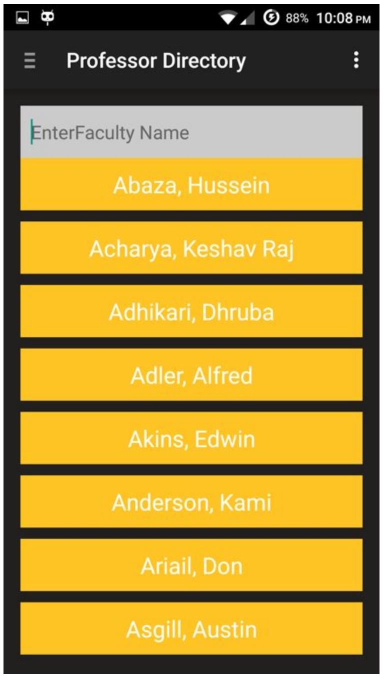
(123)456-7890

Age

21

Campus E-mail

Figure 1.3

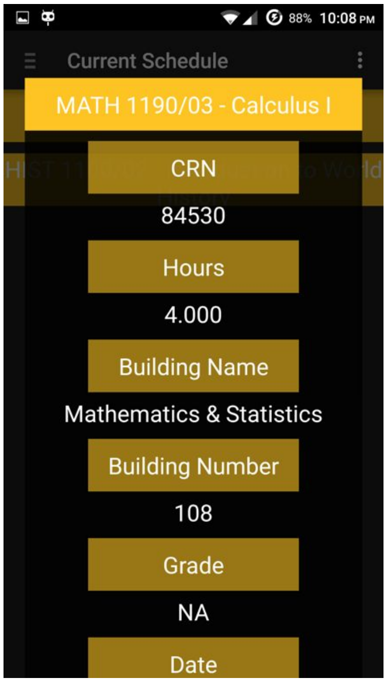


Professor Directory

EnterFaculty Name

Abaza, Hussein
Acharya, Keshav Raj
Adhikari, Dhruba
Adler, Alfred
Akins, Edwin
Anderson, Kami
Ariail, Don
Asgill, Austin

Figure 1.4



Current Schedule

MATH 1190/03 - Calculus I

CRN
84530

Hours
4.000

Building Name
Mathematics & Statistics

Building Number
108

Grade
NA

Date

Figure 1.5

1.1.3. Hardware Interface

- This application has no designated hardware, nor does it have any designated hardware interface.

1.1.4. Software Interface

- The mobile application interacts with many other features outside the application. The databases for maps, calendar, and professor directory provided by the Amazon EC2 server will interact with the application. Also, google calendar will interact with the application to upload events for personal use. Also, the maps will use google maps for location and food places.

1.1.5. Communication Interface

- The communication between the mobile application and the database is important because she depends on each other. Communication between the application and database is the key important feature.

Constraints and Assumptions

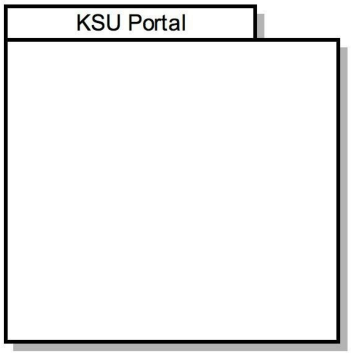
- The user is a Kennesaw State student.
- The student or faculty member has a KSU account with a username and password.
- Need a database to store student, course, faculty, map and calendar information.
- Need a server to use the databases.

System Object Model

Introduction

The System Object Model Section allows a description of each of the subsystems that are going to be used in our application. This section describes the system in an overall manner to show the different groupings of parts into respective systems. For the System Under Design, only one system is used and no subsystems are specified.

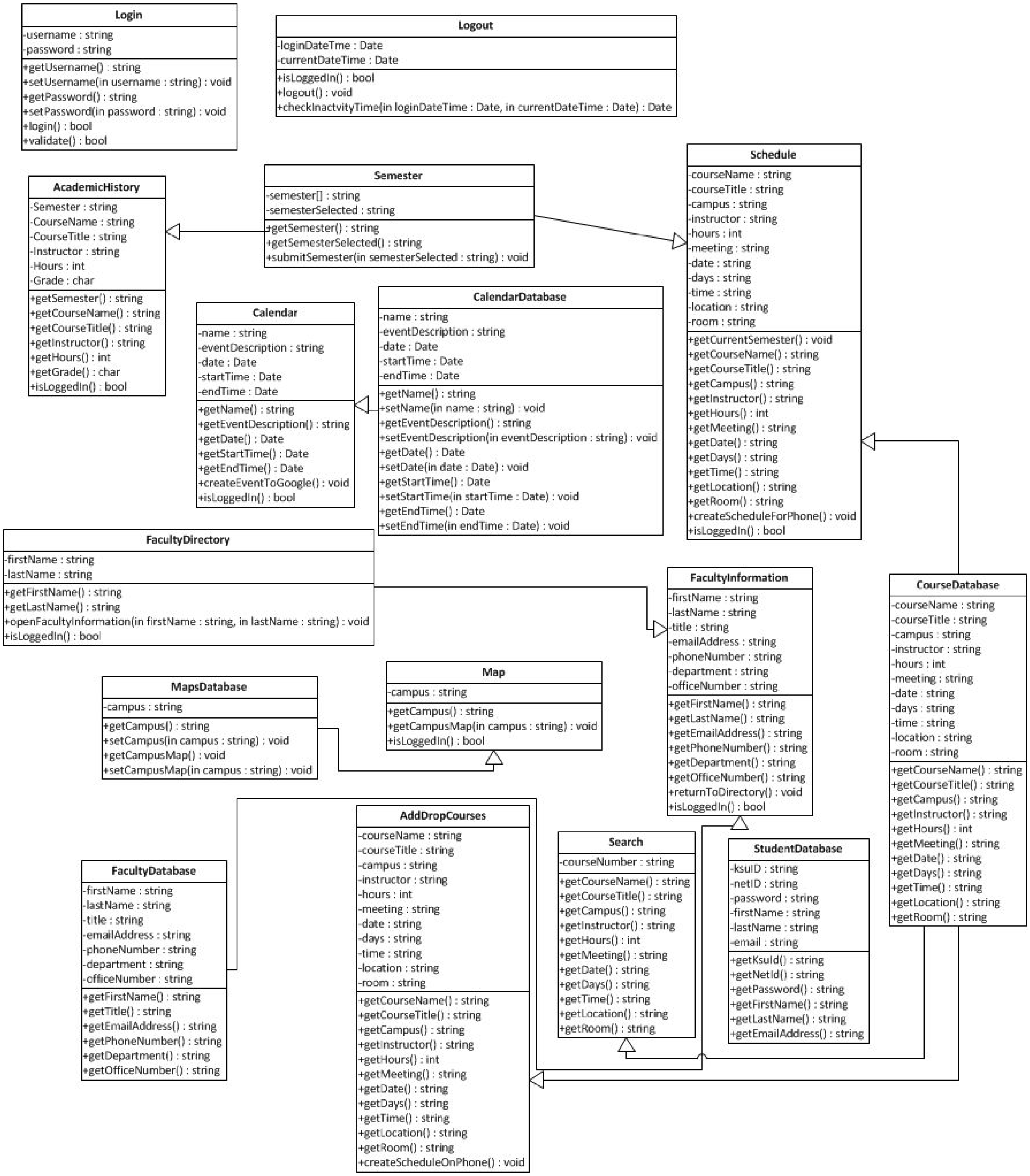
Subsystems



Subsystem Interfaces

None defined as external interfaces are used however there is no need for internal interfaces.

Object Descriptions


Login
-username : string
-password : string
+getUsername() : string
+setUsername(in username : string) : void
+getPassword() : string
+setPassword(in password : string) : void
+login() : bool
+validate() : bool
Logout
-loginDateTme : Date
-currentDateTime : Date
+isLoggedIn() : bool
+logout() : void
+checkInactvityTime(in loginDateTime : Date, in currentDateTime : Date) : Date
Schedule
-courseName : string
-courseTitle : string
-campus : string
-instructor : string
-hours : int
-meeting : string
-date : string
-days : string
-time : string
-location : string
-room : string
+getCurrentSemester() : void
+getCourseName() : string
+getCourseTitle() : string
+getCampus() : string
+getInstructor() : string
+getHours() : int
+getMeeting() : string
+getDate() : string
+getDays() : string
+getTime() : string
+getLocation() : string
+getRoom() : string
+createScheduleForPhone() : void
+isLoggedIn() : bool
AcademicHistory
-Semester : string
-CourseName : string
-CourseTitle : string
-Instructor : string
-Hours : int
-Grade : char
+getSemester() : string
+getCourseName() : string
+getCourseTitle() : string
+getInstructor() : string
+getHours() : int
+getGrade() : char
+isLoggedIn() : bool
Semester
-semester[] : string
-semesterSelected : string
+getSemester() : string
+getSemesterSelected() : string
+submitSemester(in semesterSelected : string) : void
Calendar
-name : string
-eventDescription : string
-date : Date
-startTime : Date
-endTime : Date
+getName() : string
+getEventDescription() : string
+getDate() : Date
+getStartTime() : Date
+getEndTime() : Date
+createEventToGoogle() : void
+isLoggedIn() : bool
CalendarDatabase
-name : string
-eventDescription : string
-date : Date
-startTime : Date
-endTime : Date
+getName() : string
+setName(in name : string) : void
+getEventDescription() : string
+setEventDescription(in eventDescription : string) : void
+getDate() : Date
+setDate(in date : Date) : void
+getStartTime() : Date
+setStartTime(in startTime : Date) : void
+getEndTime() : Date
+setEndTime(in endTime : Date) : void
FacultyDirectory
-firstName : string
-lastName : string
+getFirstName() : string
+getLastName() : string
+openFacultyInformation(in firstName : string, in lastName : string) : void
+isLoggedIn() : bool
FacultyInformation
-firstName : string
-lastName : string
-title : string
-emailAddress : string
-phoneNumber : string
-department : string
-officeNumber : string
+getFirstName() : string
+getLastName() : string
+getEmailAddress() : string
+getPhoneNumber() : string
+getDepartment() : string
+getOfficeNumber() : string
+returnToDirectory() : void
+isLoggedIn() : bool
CourseDatabase
-courseName : string
-courseTitle : string
-campus : string
-instructor : string
-hours : int
-meeting : string
-date : string
-days : string
-time : string
-location : string
-room : string
+getCourseName() : string
+getCourseTitle() : string
+getCampus() : string
+getInstructor() : string
+getHours() : int
+getMeeting() : string
+getDate() : string
+getDays() : string
+getTime() : string
+getLocation() : string
+getRoom() : string
MapsDatabase
-campus : string
+getCampus() : string
+setCampus(in campus : string) : void
+getCampusMap() : void
+setCampusMap(in campus : string) : void
Map
-campus : string
+getCampus() : string
+getCampusMap(in campus : string) : void
+isLoggedIn() : bool
AddDropCourses
-courseName : string
-courseTitle : string
-campus : string
-instructor : string
-hours : int
-meeting : string
-date : string
-days : string
-time : string
-location : string
-room : string
+getCourseName() : string
+getCourseTitle() : string
+getCampus() : string
+getInstructor() : string
+getHours() : int
+getMeeting() : string
+getDate() : string
+getDays() : string
+getTime() : string
+getLocation() : string
+getRoom() : string
+createScheduleOnPhone() : void
Search
-courseNumber : string
+getCourseName() : string
+getCourseTitle() : string
+getCampus() : string
+getInstructor() : string
+getHours() : int
+getMeeting() : string
+getDate() : string
+getDays() : string
+getTime() : string
+getLocation() : string
+getRoom() : string
StudentDatabase
-ksuID : string
-netID : string
-password : string
-firstName : string
-lastName : string
-email : string
+getKsuId() : string
+getNetId() : string
+getPassword() : string
+getFirstName() : string
+getLastName() : string
+getEmailAddress() : string
FacultyDatabase
-firstName : string
-lastName : string
-title : string
-emailAddress : string
-phoneNumber : string
-department : string
-officeNumber : string
+getFirstName() : string
+getTitle() : string
+getEmailAddress() : string
+getPhoneNumber() : string
+getDepartment() : string
+getOfficeNumber() : string

Component Design

1.1.6. Login

The Login Class request for a String username and String Password, which will be outputted to end users as "Net ID" and "Password" on the GUI. After the String has been taken in, the Boolean function we will validate the "username" and "password" with to database to see if they match. Once it is verified "true", the students will be able to access all the functionality of "Owl Portal".

```
    String username;
    String password;

    System.out.println("NetId:");
    System.out.println("Password ");
    username = input.nextLine();

  // Username and Password is then verified with database

public bool validate(){
    if((username == accounts[0][0]) && (password == accounts[0][1]))
        return true;
    else
        return false;
```

1.1.7. Logout

The Logout class is a feature that is feature that is requested by the user to close the information portal by the requested student. The "Logout" feature is implement in a couple of ways. The first way is when the students request for the portal to be closed and, the second option closes the portal via idleness after – minutes. This is due to security reasons.

```
public Date checkInactivityTime (Date loginDateTime)


   String loginDateTime;
   String CurrentDateTime;


// if ( no activitiy on phone)

   Action logout = new AbstractAction()


system.out.println ("Loggout due to inactivity")
```

1.1.8. Semester

The semester class provides several information to students when it is accessed. When the "Semester" class is called, past semester academic information and current semester academic information can be viewed.

```
public String [] semester{

   String [] semester = new int [];
   String semesterSelected;


   //int semester (store previous and current semester by year)

   public String getSemeterSelected()
   // students select year to view

   //database is queried for information of the semester

   system.out.println (semesterSelected);
```

1.1.9. Academic History

When the "AcademisHistory" is called, it allows a logged in student to view past academic history.

```
String couresName;
String courseTitle;
String campus;
String instructor;
int hours;
String meeting;
String date;
String time;
String location;
String room;


// assuming student in already logged in
   public String getCourseName(){
  // request courseName from database
   system.out.print (courseName)
   public String getCourseTItle(){
  // request courseTitle from database
   system.out.print (courseTitle)
 public String getcampus(){
   // request campus from database
   system.out.print (campus)
  public String getInstructor(){
   // request instructor from database
   system.out.print (instructor)
    public int getHours(){
   // request hours from database
   system.out.print (hours)
    public String getMeeting(){
   // request Meeting from database
   system.out.print (meeting)
    public String getcampus(){
   // request campus from database
   system.out.print (campus)

 }
```

1.1.10. Schedule

The “Schedule” class is used to view academic history. When a logged in student log calls the “schedule” class, the student will be able to request information from the server which include “courseName”, “courseTitle”, “campus”, “instructor”, “hours”, “meeting”, “date”, “days”, “time”, “location”, “room”. Each value holds fields relevant to student’s schedule that is displayed in a format when requested by the student.

```
String couresName;
String courseTitle;
String campus;
String instructor;
int hours;
String meeting;
String date;
String time;
String location;
String room;
String days;

public String getCourseName(){
 //system.out.print "courseName requested by student
}
public String getCourseTitle(){
//system.out.print "courseTitle" requested by student
}
public String getCampus(){
//system.out.print "campus" requested by student
}
public String getInstructor(){
//system.out.print "isntructor" requested by student
}
public int getHours(){
//system.out.print "hours" requested by student
}
```

```
public String getMeeting(){
//system.out.print "meeting" requested by student

}
public String getDate(){
//system.out.print "date" requested by student

}
public String getDays(){
//system.out.print "days" requested by student
}
public String getTime(){
//system.out.print "time" requested by student
}
public String getLocation(){
//system.out.print "location" requested by student
}
public String getRoom(){
//system.out.print "room" requested by student
}

public void createScheduleOnPhone(){
BufferedImage off_Image =
  new BufferedImage(courseName, courseTitle, campus,instructor,
                    hours,meeting,date,days,time,location,room);

// Create Image on phone

}
public String bool isLoggedIn(){

//is student logged in
if (logged in)
  //requested information
  else
  // redirect to login page

}
```

1.1.11. Add/Drop Courses

When “addDrop” class is called by a logged in student, allows the student to add and drop classes within a current semester only. Methods in the “addDrop class returns “CRN” numbers, title of students classes, campus, in which the student attends, time, date of class selected and the type of lecture it is.

```
String couresName;
String courseTitle;
String campus;
String instructor;
int hours;
String meeting;
String date;
String time;
String location;
String room;

// assuming student in already logged in
   public String getCourseName(){
  // request courseName from database
   system.out.print (courseName)
   //add choosen courseName
   public String getCourseTItle(){
  // request courseTitle from database
  // add selected courseTitle
   system.out.print (courseTitle)
 public String getcampus(){
   // request campus from database
   // add selected campus
   system.out.print (campus)
  public String getInstructor(){
   // request instructor from database
   // add selected instructor
   system.out.print (instructor)
    public int getHours(){
   // request hours from database
   // add selected hours
   system.out.print (hours)
    public String getMeeting(){
   // request Meeting from database
   // add selected meeting
   system.out.print (meeting)


 ...
```

1.1.12. Search

The “Search” class gives a logged in student the ability to search through the database and display courses through specific classes.

```
String courseName;

public String getCourseName(){
// holds value for a selected Coursename
   system.out.print coursename
}
public String getCourseTitle(){
// holds value for a selected Coursetitle
  system.out.print courseTitle
}
public String getcampuses(){
// holds value for a selected campus
  system.out.print campus
}
public String getIstructor(){
// holds value for a selected instructor
  system.out.print instructor
}
public int getHours(){
// holds value for a selected hour integer
  system.out.print Hour
}
public String getMeetings(){
// holds value for a selected meeting
  system.out.print meeting
}
public String getDate(){
// holds value for a selected Date
  system.out.print date
}
public String getDays(){
// holds value for a selected days
  system.out.print days
}
public String getTime(){
// holds value for a selected time
  system.out.print time
}
public String getLocation(){
// holds value for a selected location
  system.out.print lcoation
}
public String getroom(){
// holds value for a selected room
  system.out.print room

}
```

1.1.13. Calendar

When the “Calendar” class is called, a logged in student will be able to view events around the Kennesaw State University Campus. A method will also for student to be able to create an event with the google calendar. For this class to be fully functional, the google API will be called to allow students store events on their devices.

```
String name;
String eventDescription;
String date;
String startTime;
String endTime;

public String getName(){
//returns the stored name from the database requested by the user
}
public String geteventDescription(){
//returns the stored event from the database requeted by the user
}
public Srting getDate(){
 //student chooses event from eventDescription
 //returns the date for eventDescription
}

public String getstartTime(){
 //student chooses event from eventDescription
 //returns the startTime for eventDescription
}
public String getendTime(){
 //student chooses event from eventDescription
 //returns the endTime for eventDescription
}
public void createEventToGoogle(){
//Call google API
//student select event from avaialbel eventDescription
//add to google celendar
}
public String bool isLoggedIn(){

//is student logged in
if (logged in)
   //return true
   else
   // redirect to login page

}
```

Update - The calendar feature for KSU events will be a future endeavor. Currently the application will export the user’s schedule to their native calendar application.

1.1.14. Map

The “Map” class will use the Google API for full functionality. It will provide a logged in student visual access to both the Kennesaw and the Marietta Campus. The map will show various buildings on the mobile application to aid student in getting familiar with their campuses.

```
// Call Google Maps API displays either a select
String Campus;
public String getcampus(){
 string kennesawCampus
 string MariettaCampus
}
public void getCampusMap(String campus){
//Student chooses from either the Marietta or Kennesaw Campus
 //if (student == kennesawCampus) {
           display Kennesaw Campus Map
        } else if (student == mariettaCampus) {
           displayMarietta campus Map
}
```

Faculty Directory

The “facultyDirectory” class will provide a logged in student access to the database that features just the faculty directory

```
String firstName;
String lastName;

public String getfirstName(){
// Holds faculty first name

}
public String getlastName(){
//hold faculty last name

}
public void openFacultyInformation(firstName lastName){

system.out.print (firstName lastName)
}
public String bool isLoggedIn(){

//is student logged in
if (logged in)
   //return true
   else
   //redirect to Login Page

}
```

1.1.15. Faculty Information

The “facultyInformation” class will provide a logged in student access to the database that features various details of faculty that includes “firstName”, “lastName”, “title”, “emailAddress”, “phoneNumber”, “department”, “officeNumber”.

```
String firstName;
String lastName;
String title;
Sting emailAddress;
String phoneNumber;
String department;
String officeNumber;

public String getfirstName(){
//holds stored first name of faculty

}
public String getlastName(){
//holds stored last name of faculty
}
public String gettitle(){
//holds stored title of professor
}
public String getephoneNumber(){
//holds stored phone number of faculty
}
public String getdepartment(){
//holds stored department of faculty
}
public String getofficenumber(){
//holds stored office number of faculty
}
public void returnToDirectory(){
if (cannot find faculty)
    else
    (return)
}
public String bool isLoggedIn(){

//is student logged in
if (logged in)
  //return true
  else
  system.out.println("Log in is required")

}
```

1.1.16. Student Database

The “studentDatabase” class

```
String ksuID;
String netID
String password;
String firstName;
String lastName;
String email;

public String getksuID(){
//holds students KSU ID
}
public String getnetID(){
// holds student netID
}
public String getpassword(){
//holds student password

public String getfirstName(){
//holds students first name
}
public String getlastName(){
//holds student last name
}
public String getemail(){
//holds students email
}
```

HUMAN INTERFACE DESIGN

Overview of User Interface

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

Screen Images

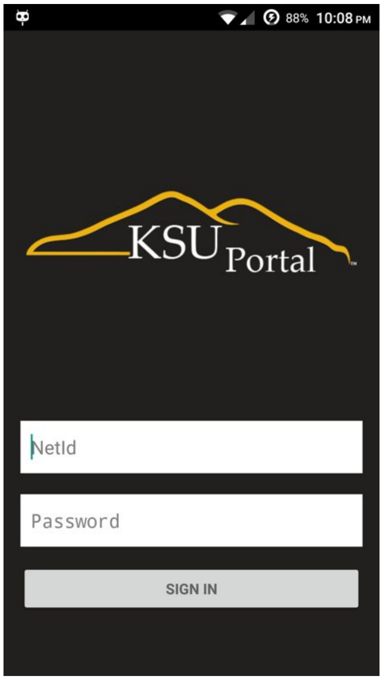


Figure : Login Screen

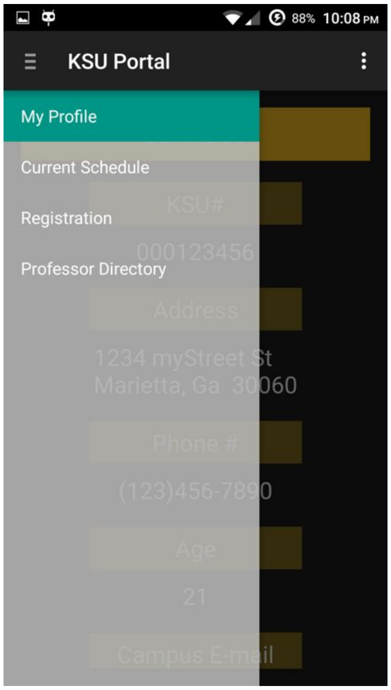
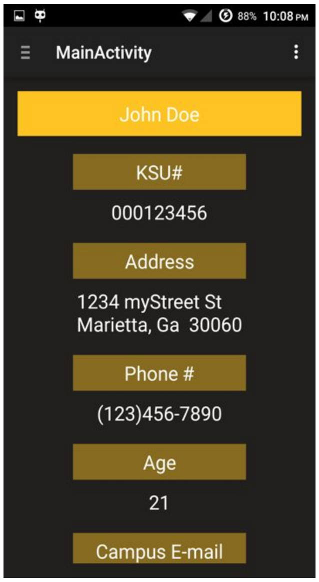


Figure : GUI Options Screen



John Doe

KSU#

000123456

Address

1234 myStreet St
Marietta, Ga 30060

Phone #

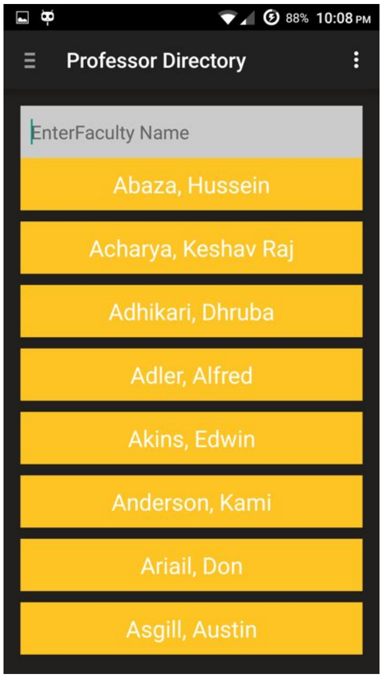
(123)456-7890

Age

21

Campus E-mail

Figure : Student Profile Screen



88% 10:08 PM

Professor Directory

EnterFaculty Name

Abaza, Hussein

Acharya, Keshav Raj

Adhikari, Dhruba

Adler, Alfred

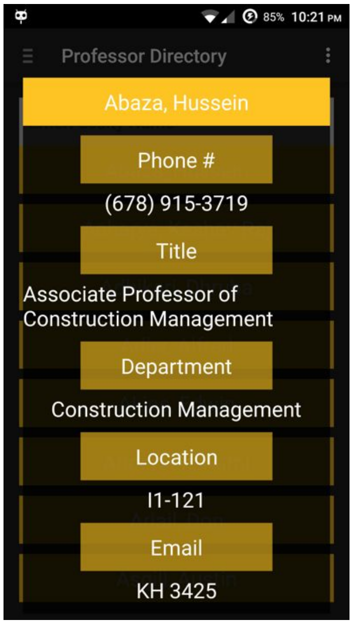
Akins, Edwin

Anderson, Kami

Ariail, Don

Asgill, Austin

Figure : Professor Directory



85% 10:21 PM

Professor Directory

Abaza, Hussein

Phone #

(678) 915-3719

Title

Associate Professor of Construction Management

Department

Construction Management

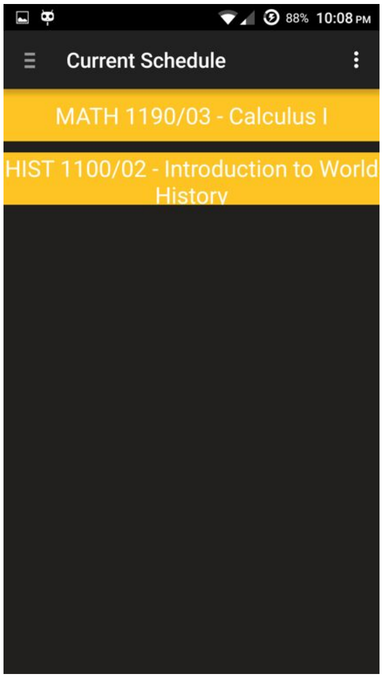
Location

I1-121

Email

KH 3425

Figure 5: Professor Profile



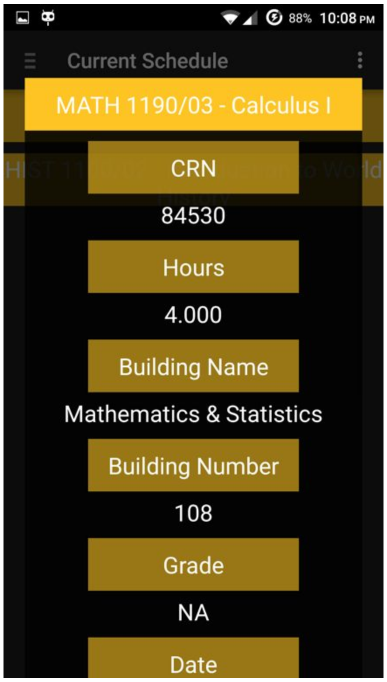
88% 10:08 PM

Current Schedule

MATH 1190/03 - Calculus I

HIST 1100/02 - Introduction to World History

Figure 6: Schedule Screen



88% 10:08 PM

Current Schedule

MATH 1190/03 - Calculus I

CRN

84530

Hours

4.000

Building Name

Mathematics & Statistics

Building Number

108

Grade

NA

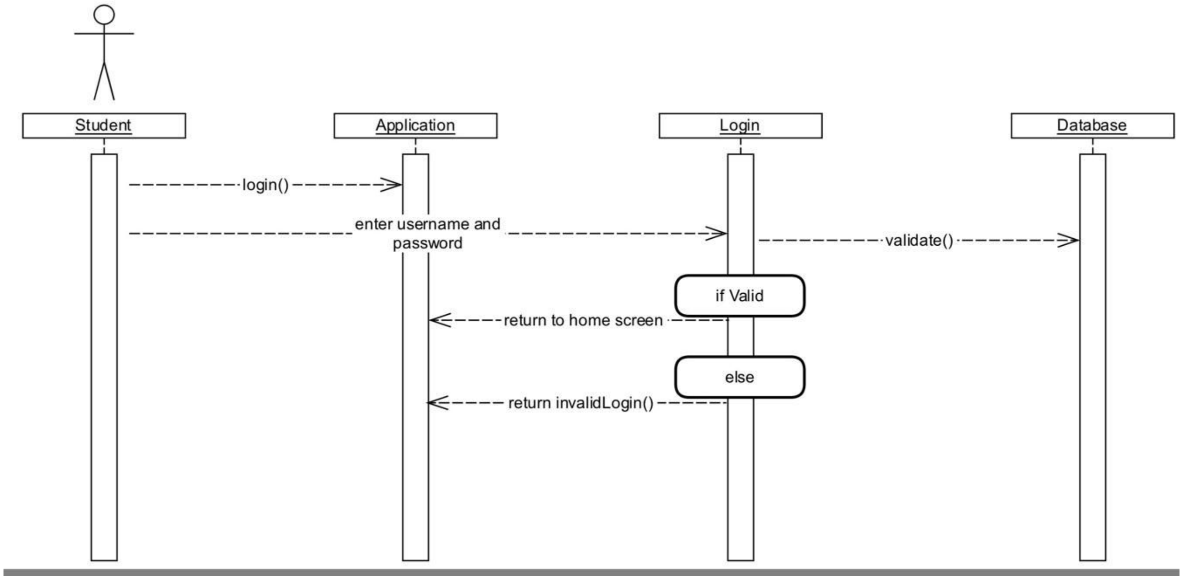
Date

Figure 7: Selected Course Screen

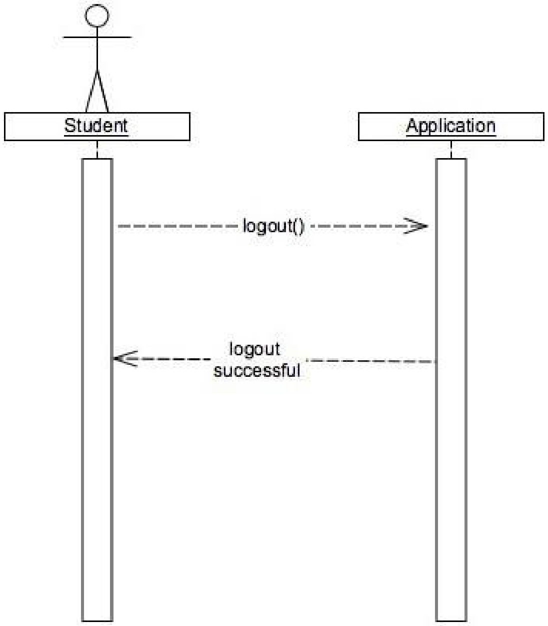
Dynamic Model

Sequence Diagrams

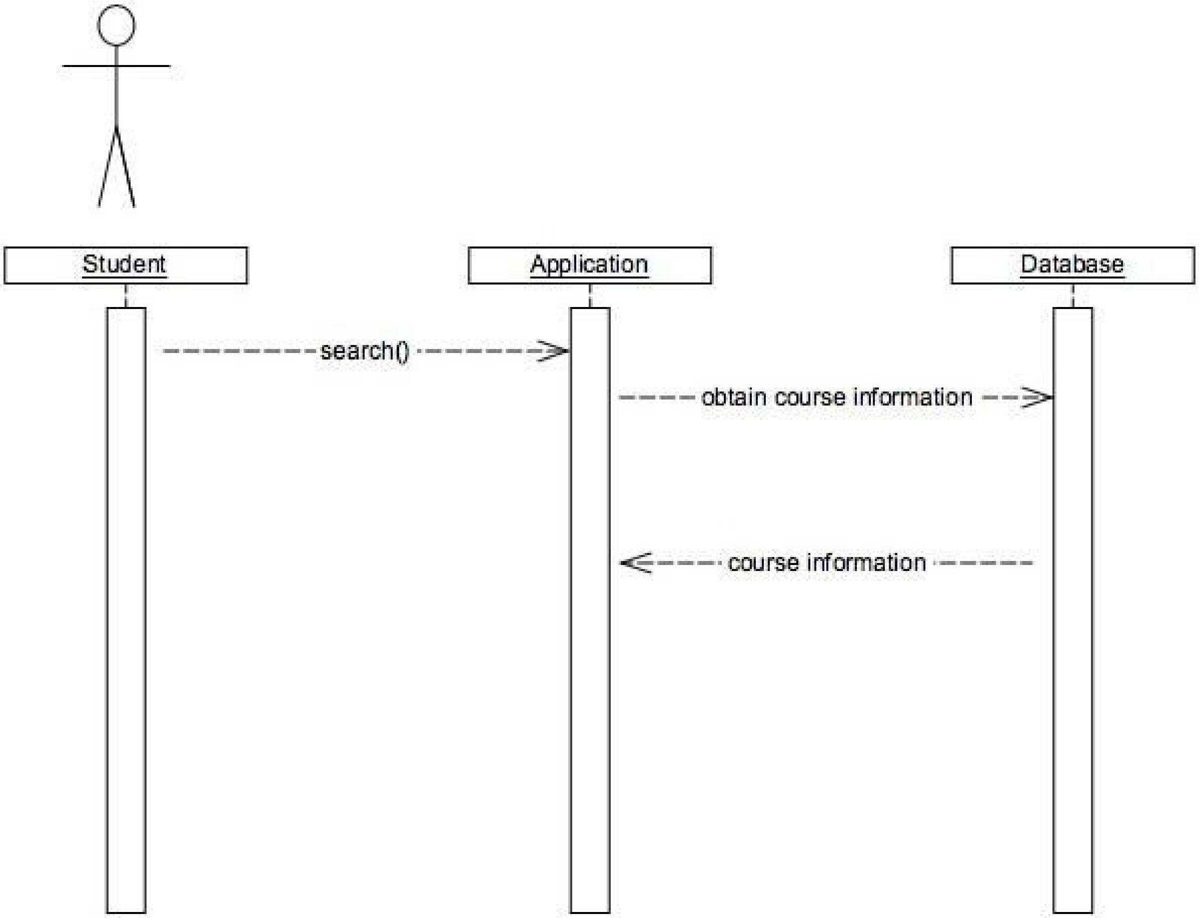
1.1.17. Student Sequence Diagrams:



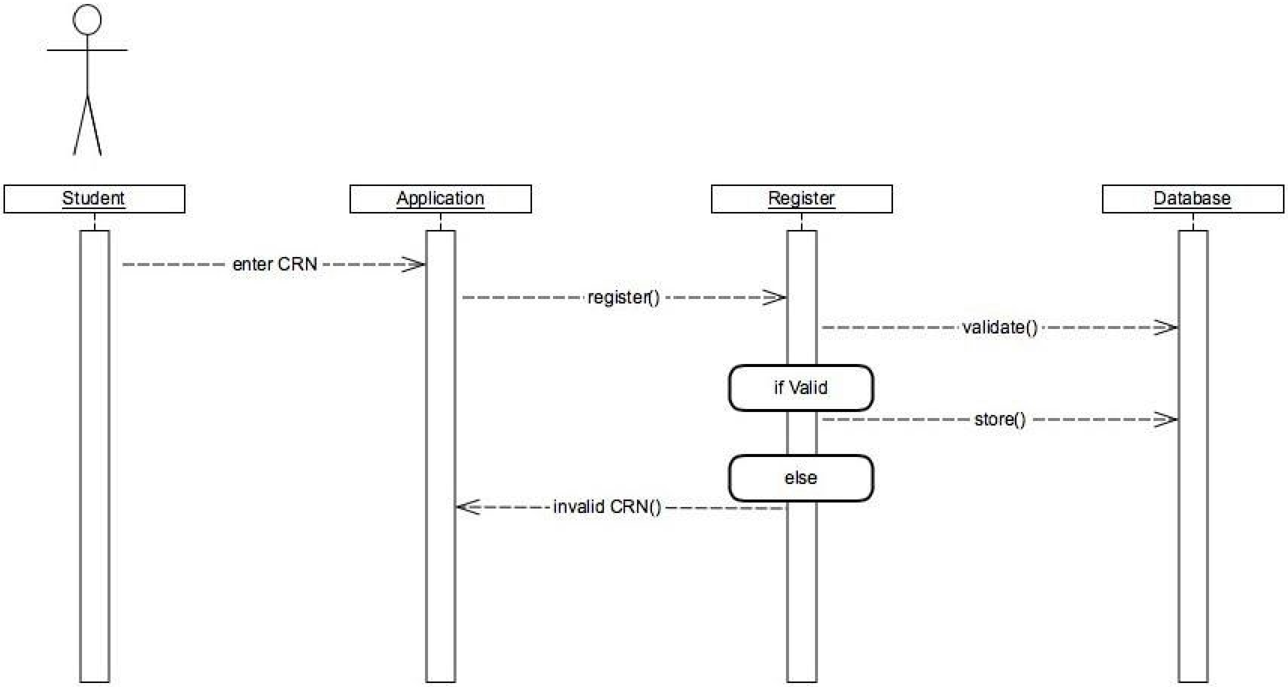
Login



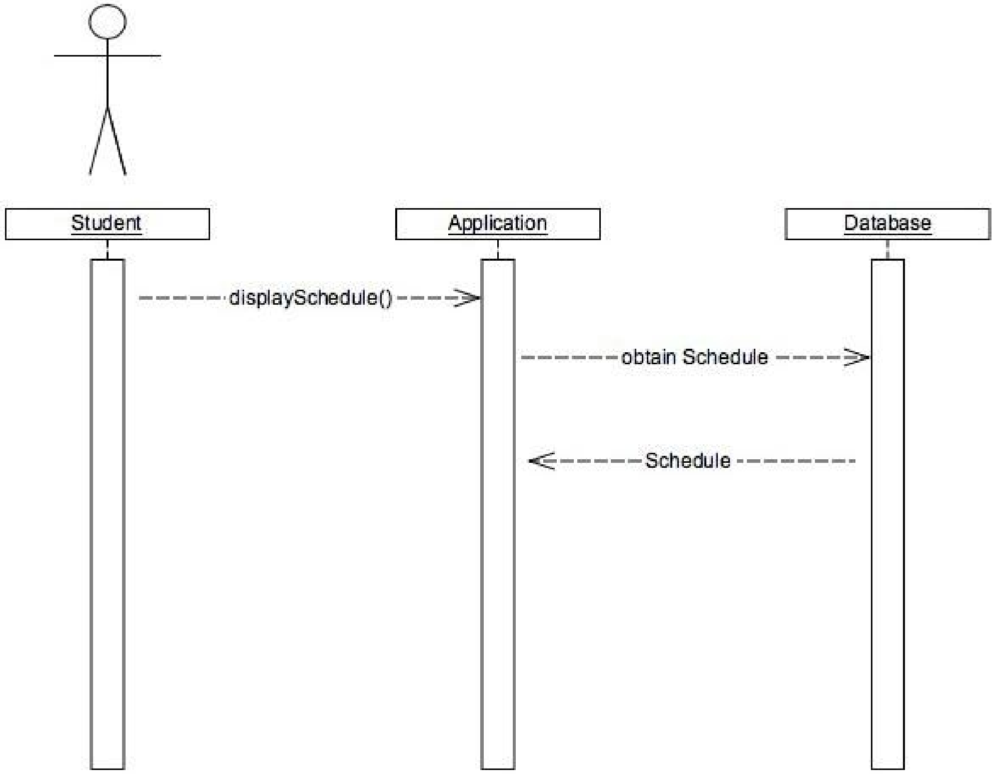
Logout



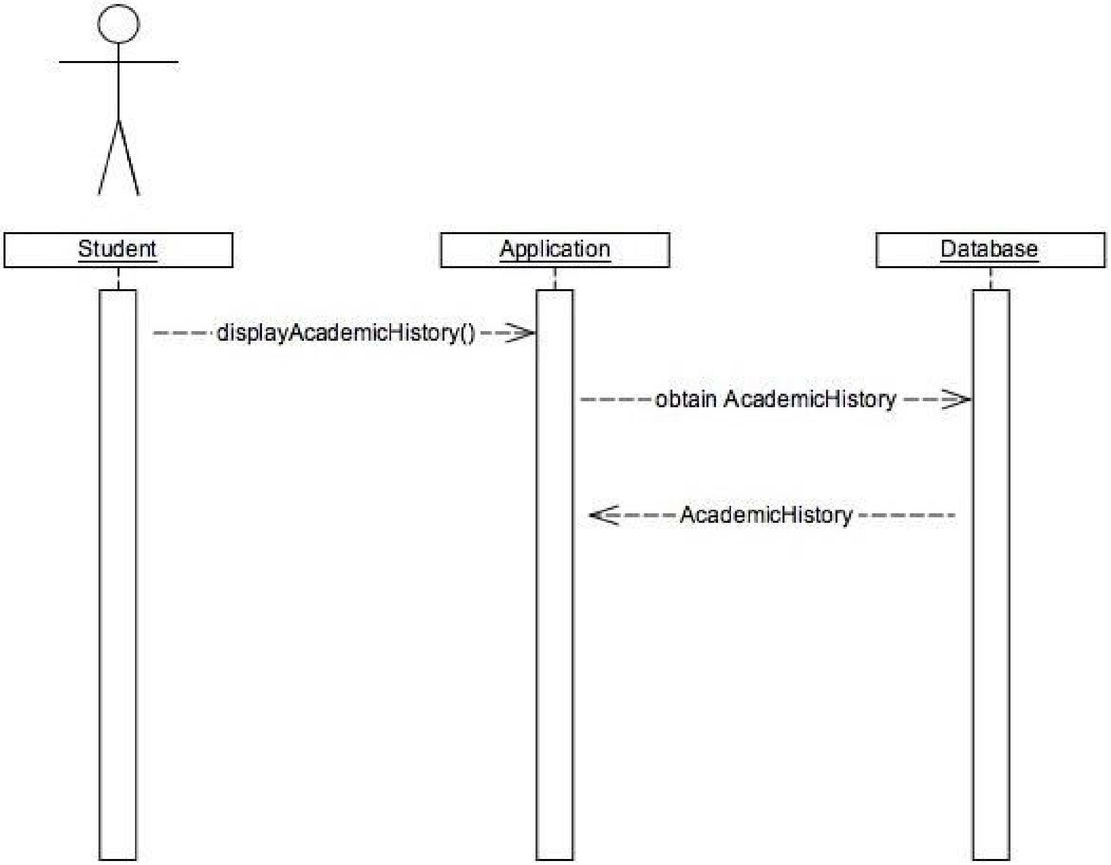
Course Search



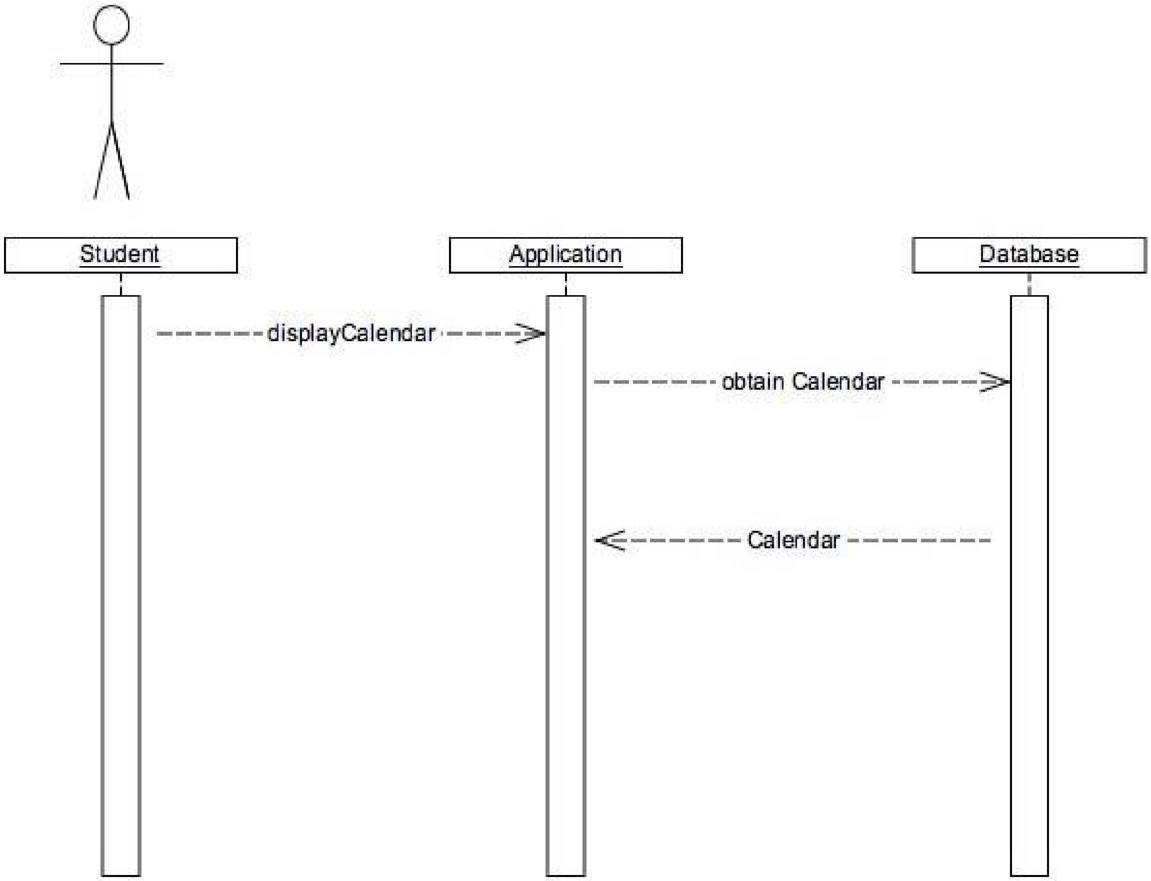
Course Add/Drop Registration



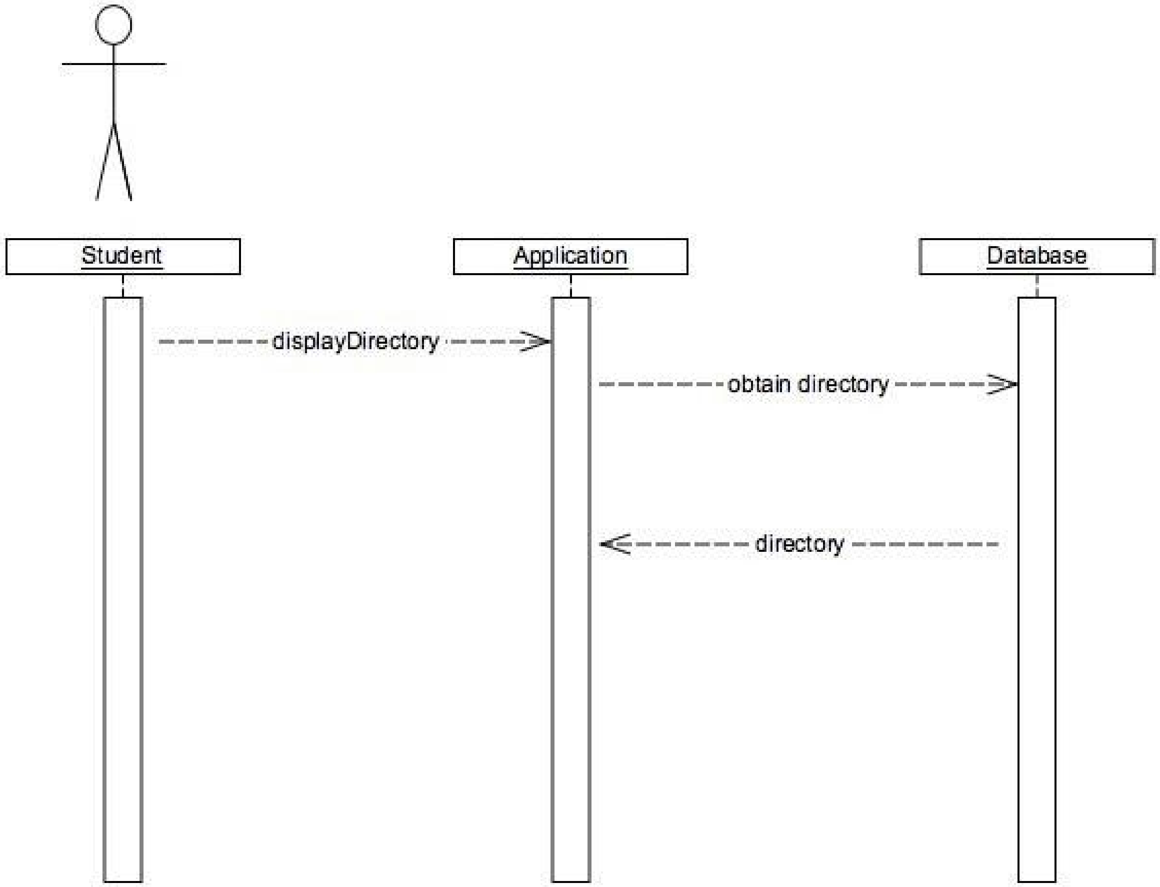
Schedule Display



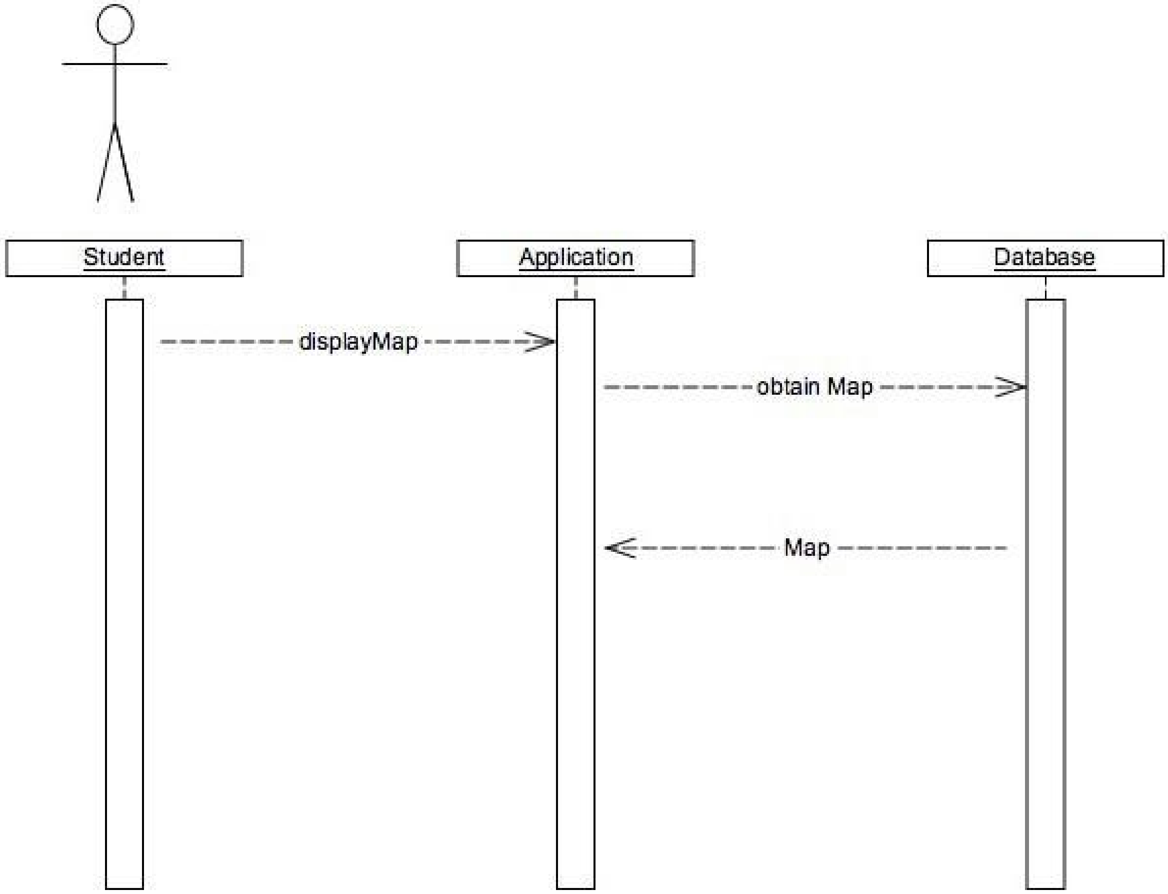
Academic History



Calendar Display

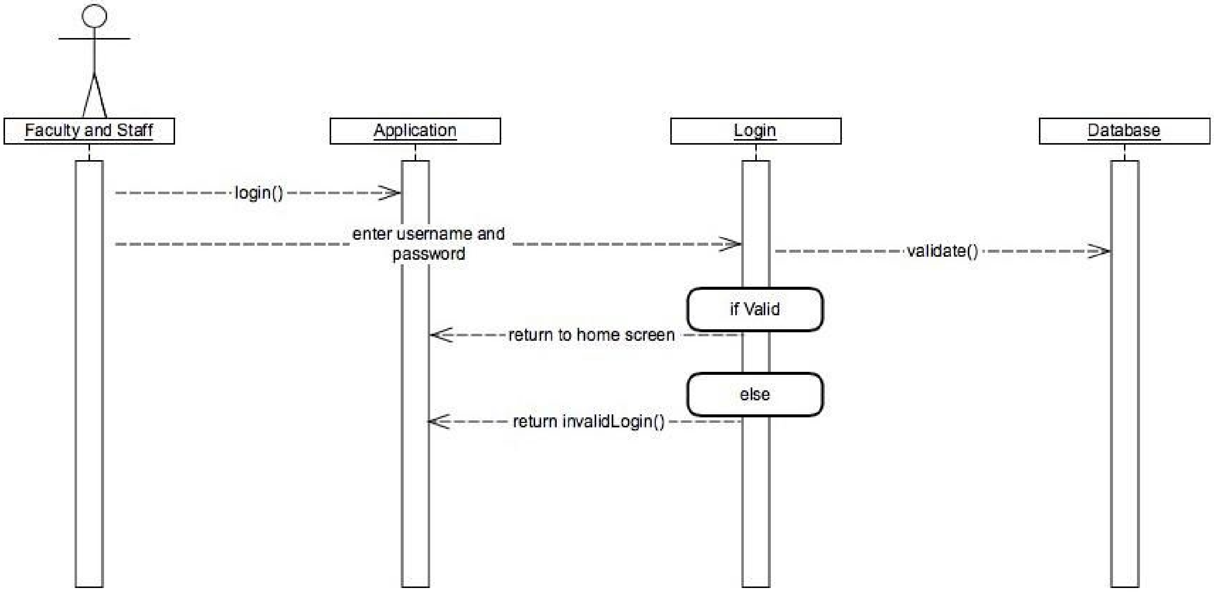


Faculty Directory



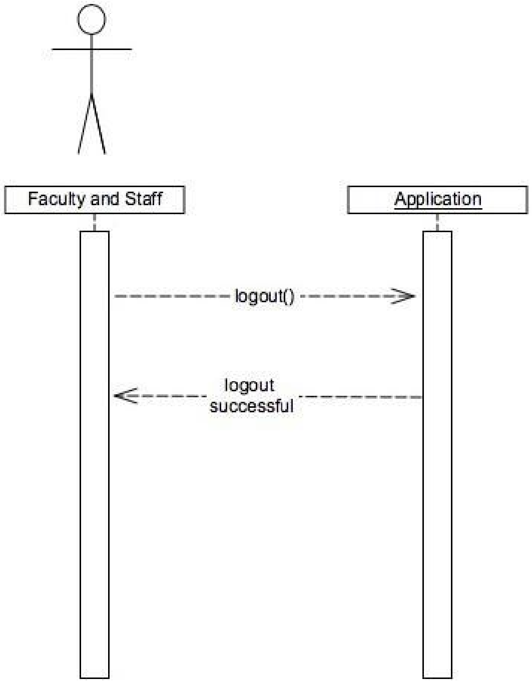
Map Display

1.1.18. Faculty and Staff Sequence Diagrams:



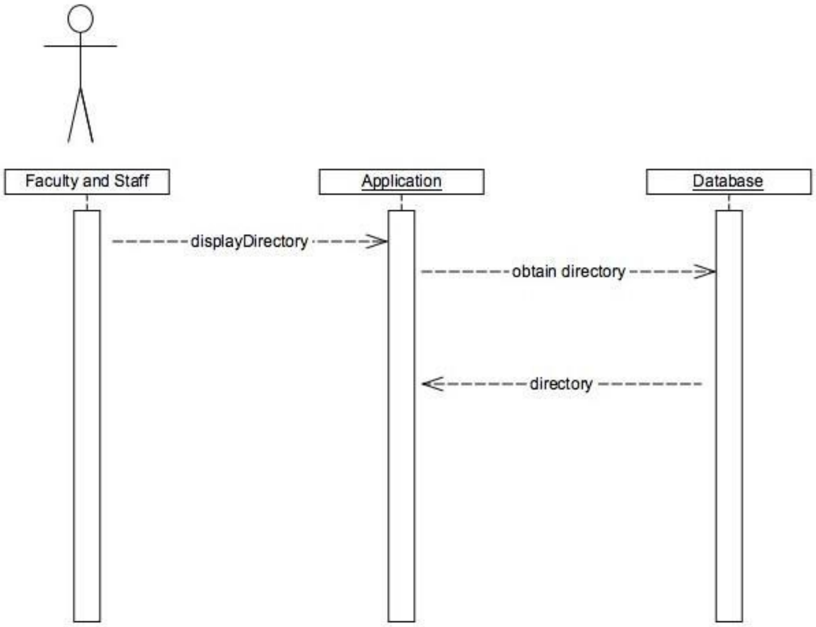
Login

Update - This will be a future addition.

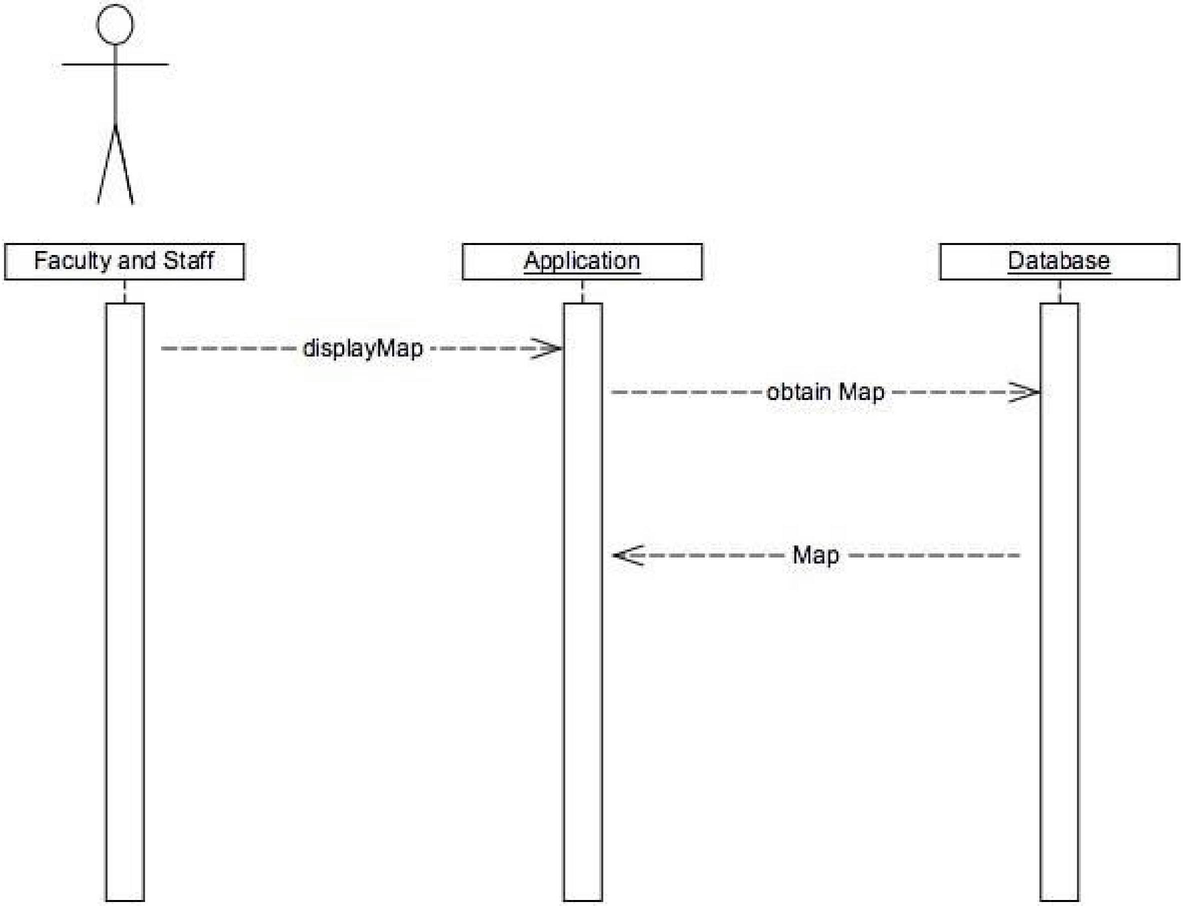


Logout

Update - This will be a future addition.

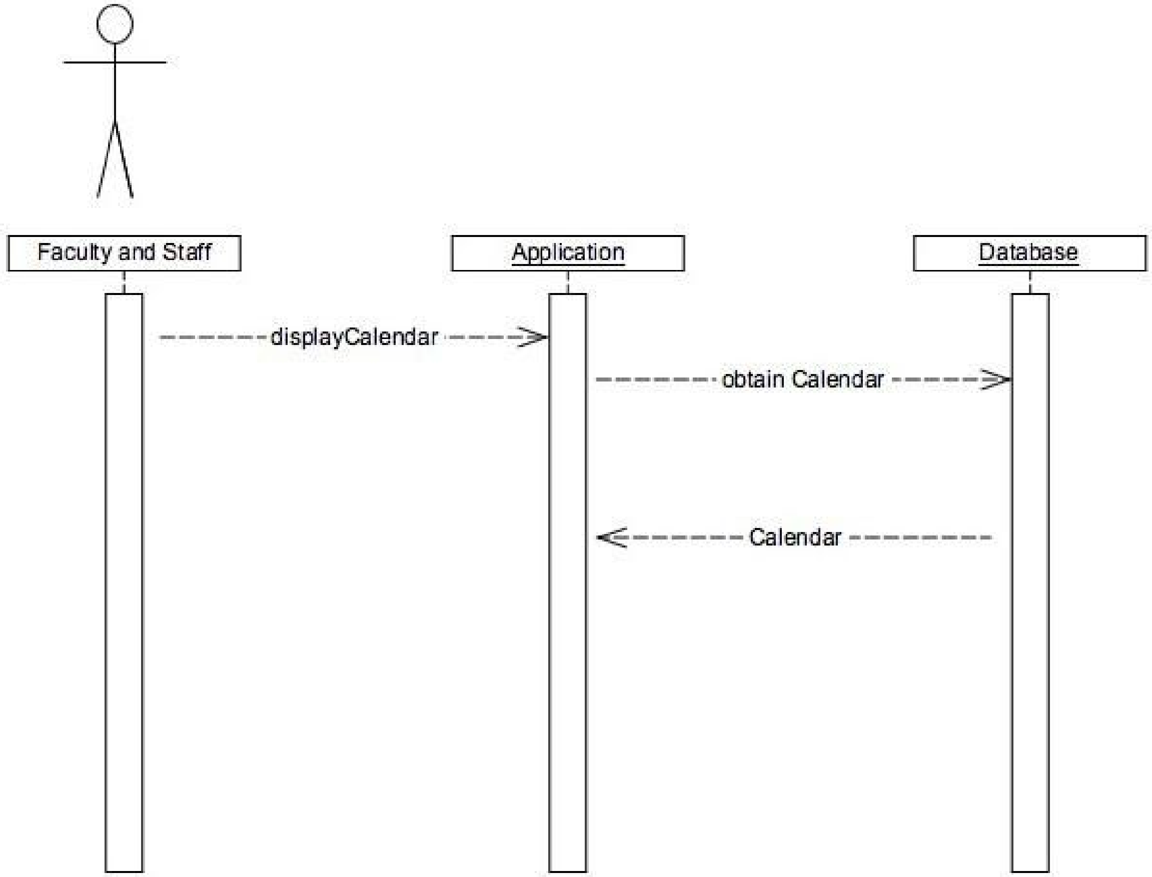


Directory Search for Faculty and Staff



Display Map

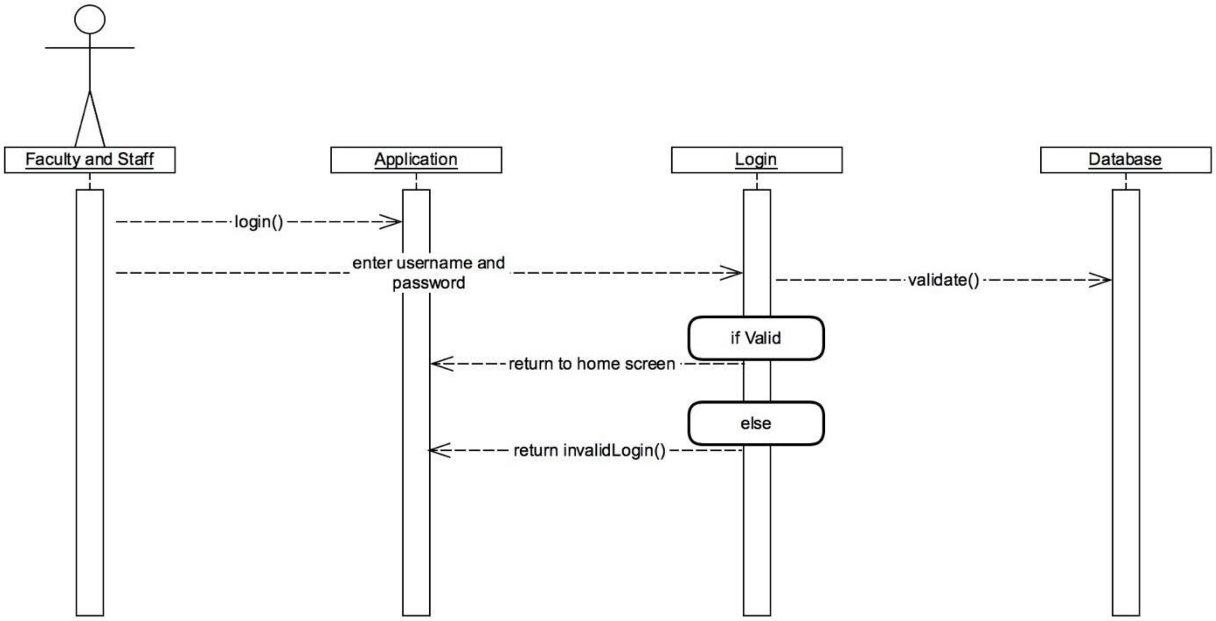
Update - This will be a future addition.



Display Calendar

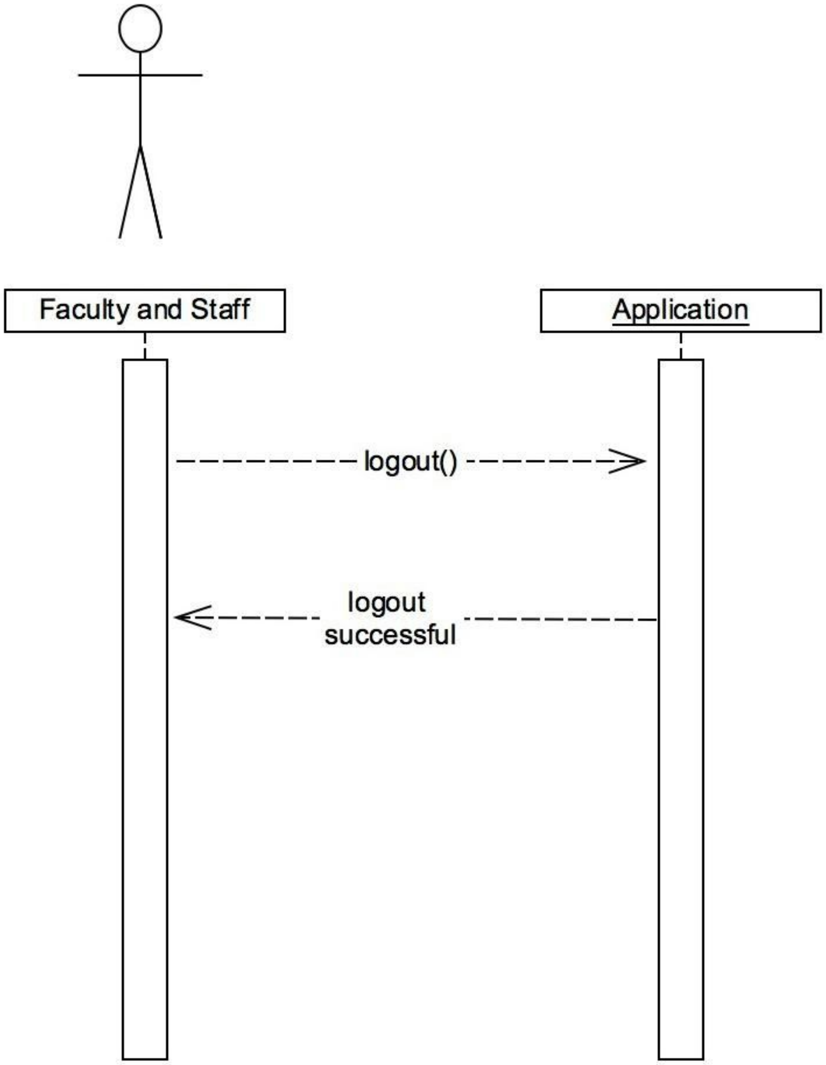
Update - This will be a future addition.

1.1.19. It Admin Sequence Diagrams:

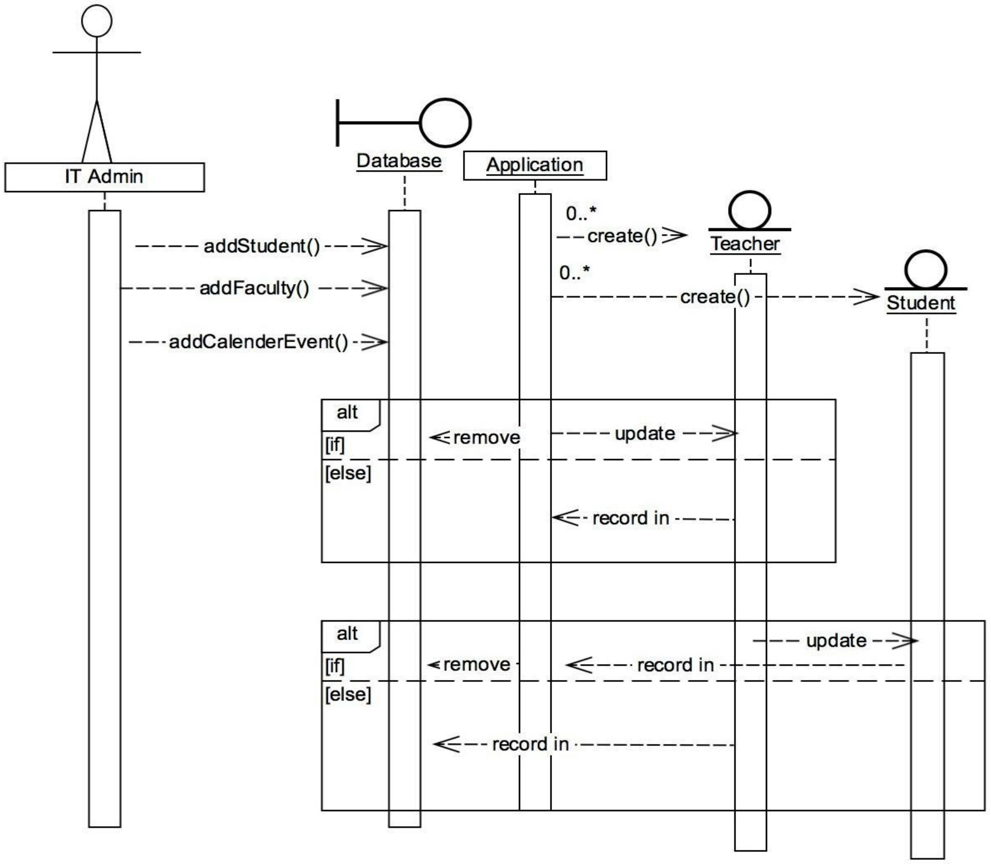


Login

Update - This will be a future addition.

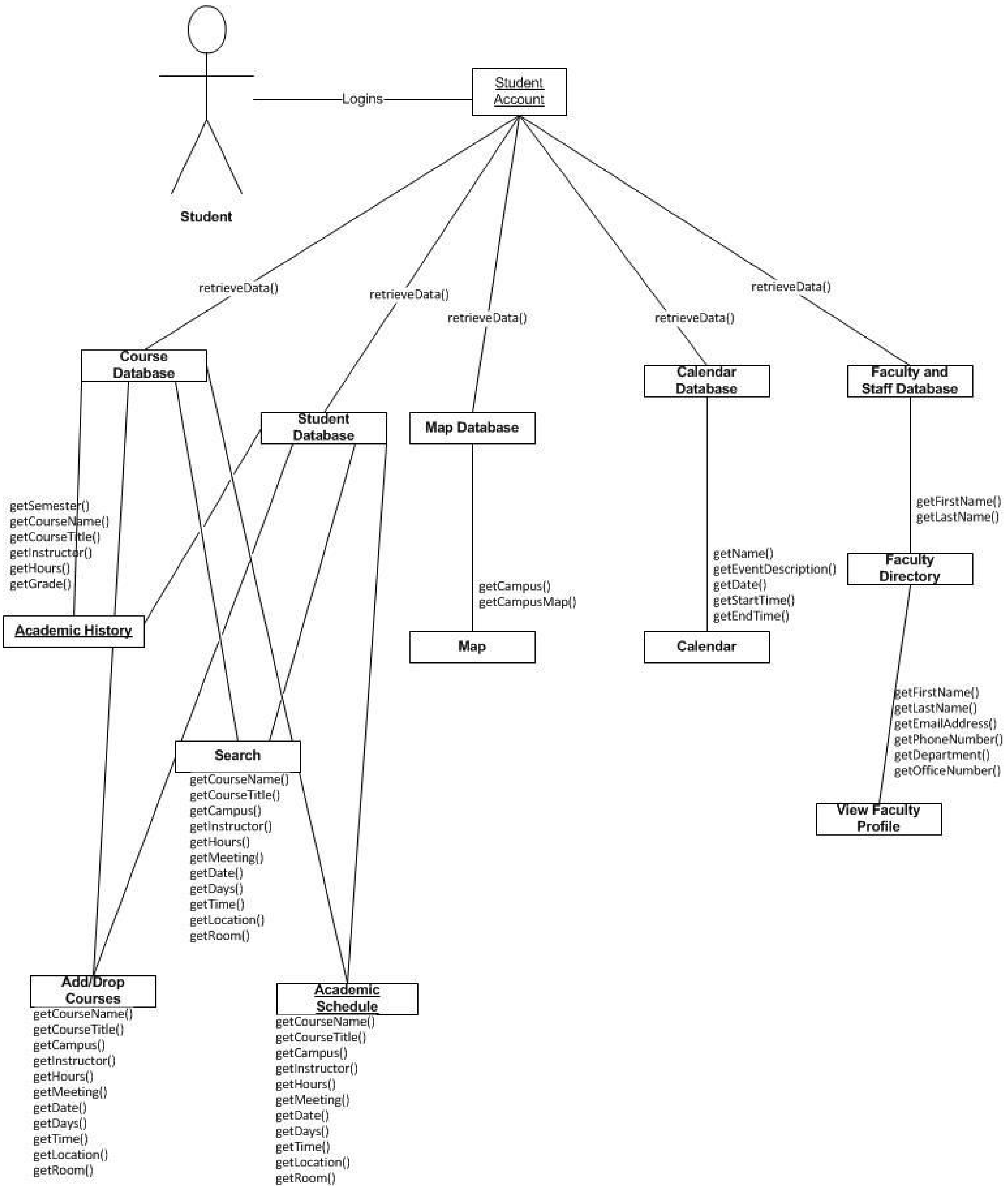

Faculty and Staff
Application
logout()
logout
successful

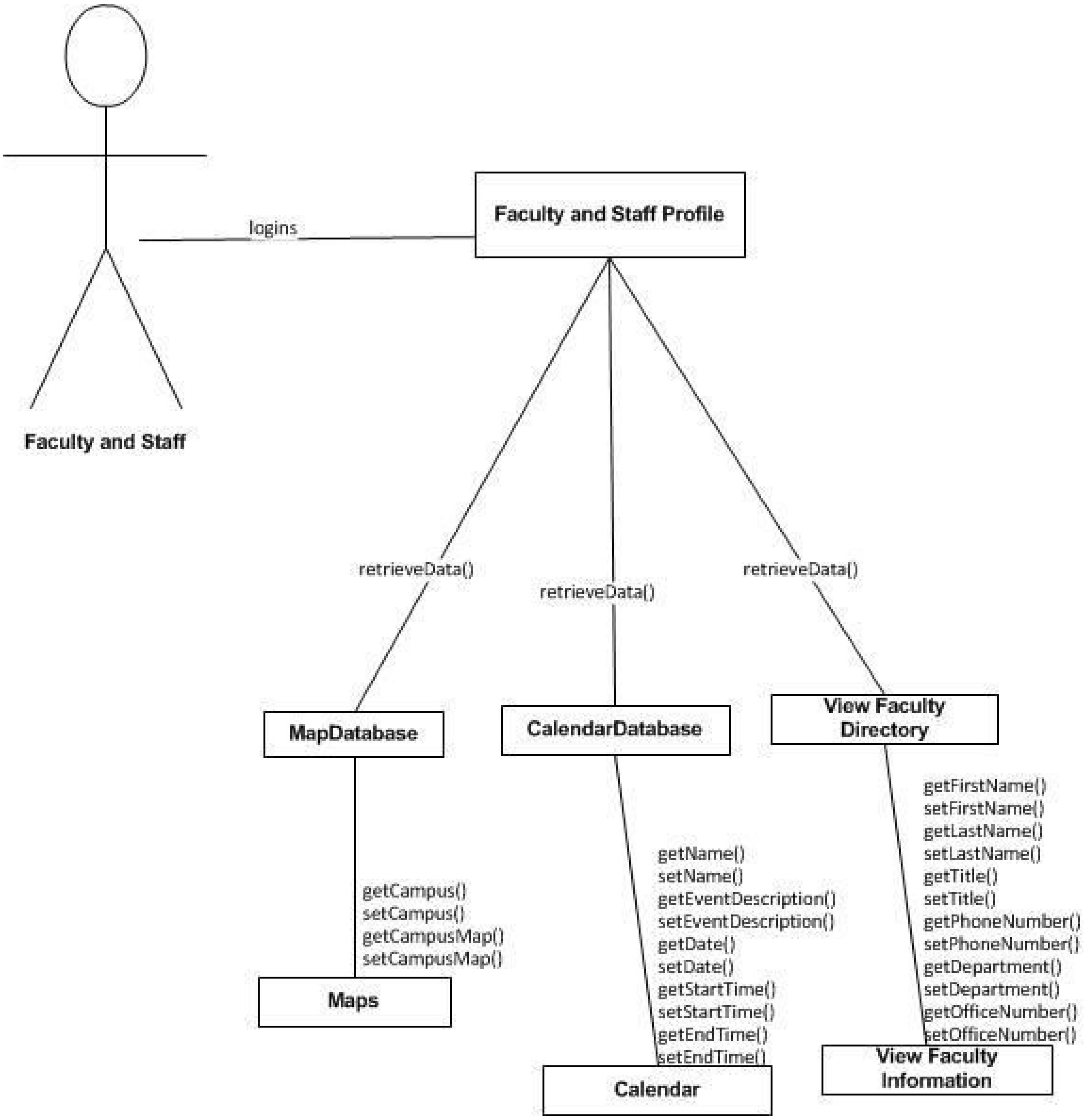
Logout

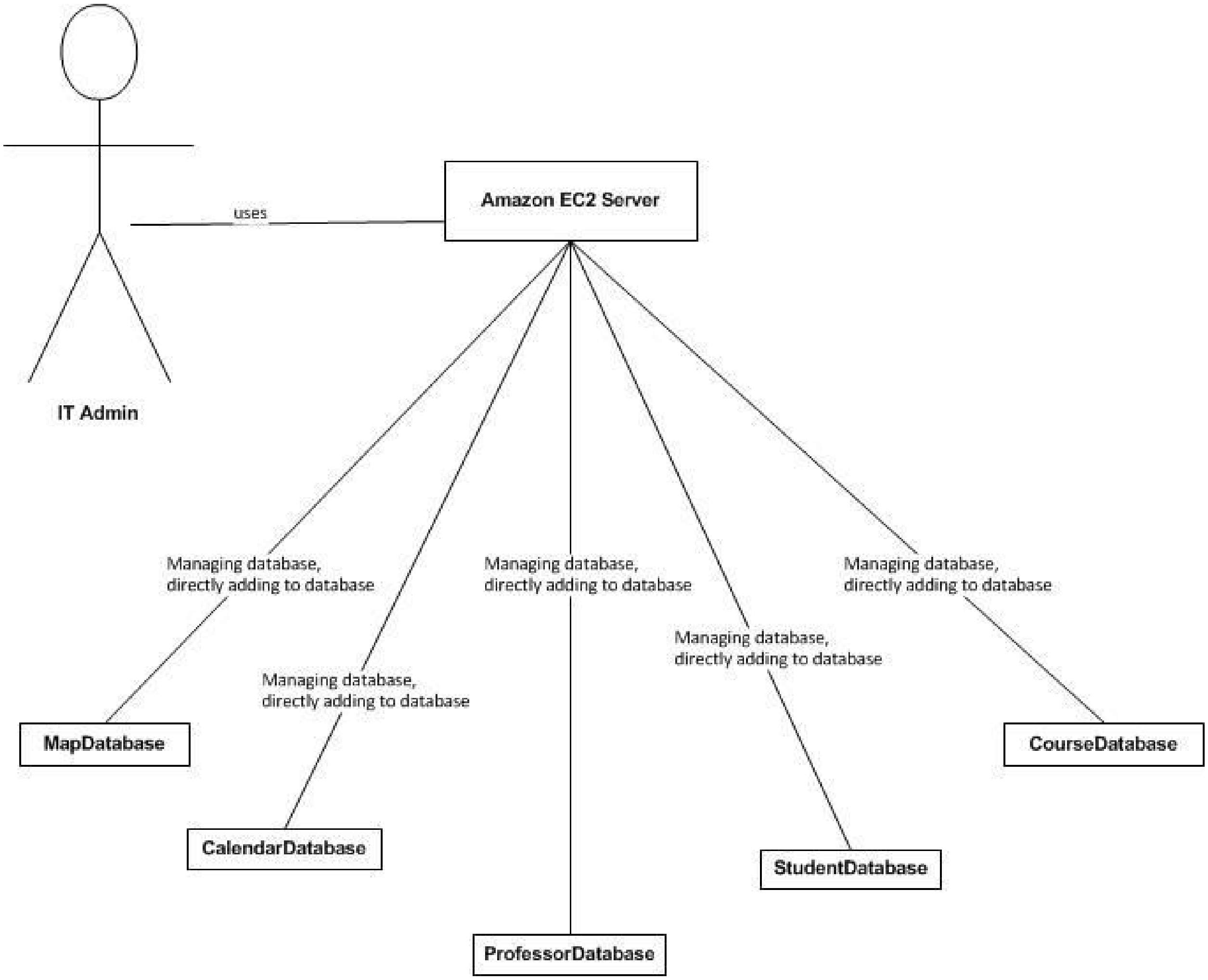


Database Management

Object Collaboration Diagrams

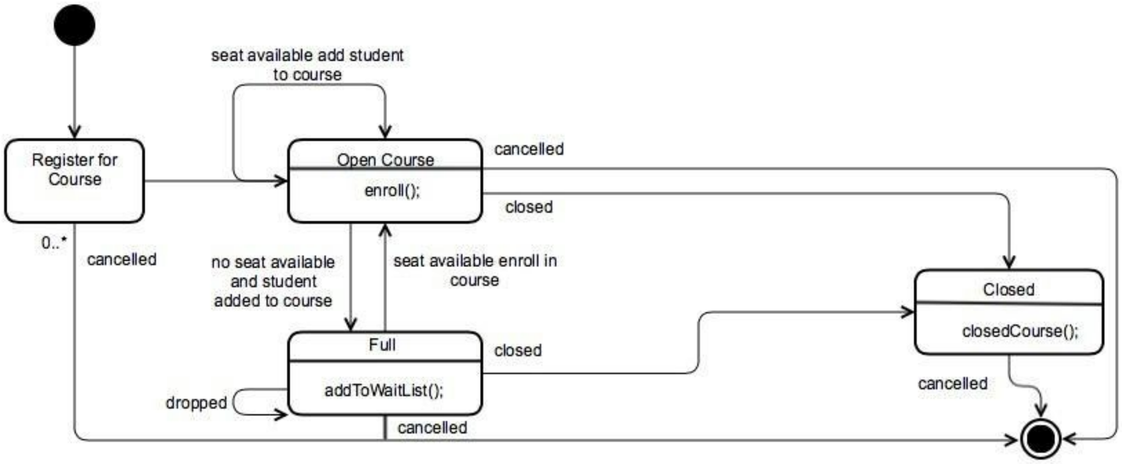

Student
Logins
Student Account
retrieveData()
retrieveData()
retrieveData()
retrieveData()
retrieveData()
Course Database
Student Database
Map Database
Calendar Database
Faculty and Staff Database
getSemester()
getCourseName()
getCourseTitle()
getInstructor()
getHours()
getGrade()
Academic History
getCampus()
getCampusMap()
Map
getName()
getEventDescription()
getDate()
getStartTime()
getEndTime()
Calendar
getFirstName()
getLastName()
Faculty Directory
getFirstName()
getLastName()
getEmailAddress()
getPhoneNumber()
getDepartment()
getOfficeNumber()
View Faculty Profile
Search
getCourseName()
getCourseTitle()
getCampus()
getInstructor()
getHours()
getMeeting()
getDate()
getDays()
getTime()
getLocation()
getRoom()
Add/Drop Courses
getCourseName()
getCourseTitle()
getCampus()
getInstructor()
getHours()
getMeeting()
getDate()
getDays()
getTime()
getLocation()
getRoom()
Academic Schedule
getCourseName()
getCourseTitle()
getCampus()
getInstructor()
getHours()
getMeeting()
getDate()
getDays()
getTime()
getLocation()
getRoom()


Faculty and Staff
logins
Faculty and Staff Profile
retrieveData()
retrieveData()
retrieveData()
MapDatabase
CalendarDatabase
View Faculty Directory
getCampus()
setCampus()
getCampusMap()
setCampusMap()
Maps
getName()
setName()
getEventDescription()
setEventDescription()
getDate()
setDate()
getStartTime()
setStartTime()
getEndTime()
setEndTime()
Calendar
getFirstName()
setFirstName()
getLastName()
setLastName()
getTitle()
setTitle()
getPhoneNumber()
setPhoneNumber()
getDepartment()
setDepartment()
getOfficeNumber()
setOfficeNumber()
View Faculty Information

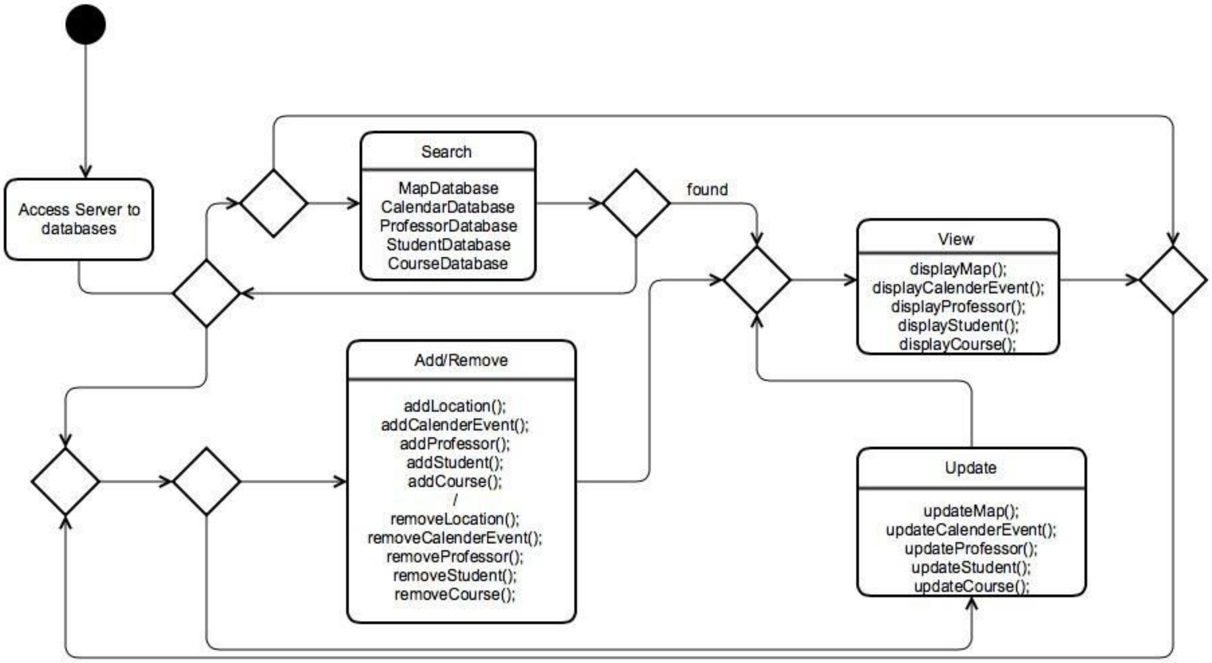

IT Admin
uses
Amazon EC2 Server
Managing database, directly adding to database
Managing database, directly adding to database
Managing database, directly adding to database
Managing database, directly adding to database
Managing database, directly adding to database
Managing database, directly adding to database
MapDatabase
CalendarDatabase
ProfessorDatabase
StudentDatabase
CourseDatabase

Update - Currently the only user is students. This application was designed for student use only. The only thing faculty and staff would be able to do is view other faculty information and use the map feature. The application was designed to be a mobile version of OWL express and OWL express is mainly for student use. In the future, we would add faculty and staff and add more features for those faculty and staff besides maps and faculty directory.

Object Behavior Diagrams



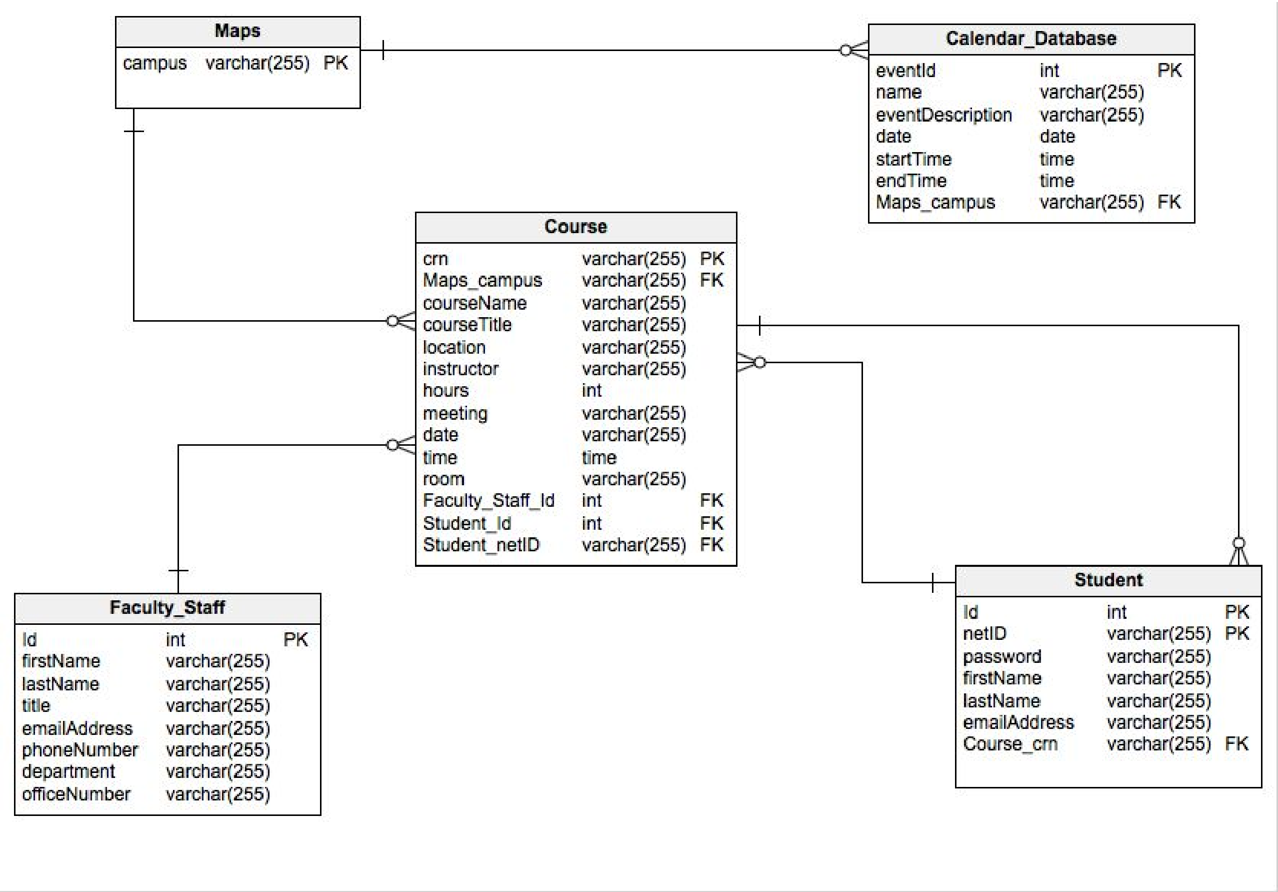
Student State Diagram



It Admin State Diagram

DATA DESIGN

Database Description



Maps		
campus	varchar(255)	PK

Calendar_Database		
eventId	int	PK
name	varchar(255)	
eventDescription	varchar(255)	
date	date	
startTime	time	
endTime	time	
Maps_campus	varchar(255)	FK

Course		
crn	varchar(255)	PK
Maps_campus	varchar(255)	FK
courseName	varchar(255)	
courseTitle	varchar(255)	
location	varchar(255)	
instructor	varchar(255)	
hours	int	
meeting	varchar(255)	
date	varchar(255)	
time	time	
room	varchar(255)	
Faculty_Staff_Id	int	FK
Student_Id	int	FK
Student_netID	varchar(255)	FK

Faculty_Staff		
Id	int	PK
firstName	varchar(255)	
lastName	varchar(255)	
title	varchar(255)	
emailAddress	varchar(255)	
phoneNumber	varchar(255)	
department	varchar(255)	
officeNumber	varchar(255)	

Student		
Id	int	PK
netID	varchar(255)	PK
password	varchar(255)	
firstName	varchar(255)	
lastName	varchar(255)	
emailAddress	varchar(255)	
Course_crn	varchar(255)	FK

Database Design

1.1.20. Map

```
-- tables
-- Table Maps
CREATE TABLE Maps (
  campus varchar(255)  NOT NULL,
  CONSTRAINT Maps_pk PRIMARY KEY (campus)
);
```

1.1.21. Calendar

```
-- tables
-- Table Maps
CREATE TABLE Maps (
  campus varchar(255)  NOT NULL,
  CONSTRAINT Maps_pk PRIMARY KEY (campus)
);
```

1.1.22. Course

```
-- tables
-- Table Course
CREATE TABLE Course (
  crn varchar(255)  NOT NULL,
  Maps_campus varchar(255)  NOT NULL,
  courseName varchar(255)  NOT NULL,
  courseTitle varchar(255)  NOT NULL,
  location varchar(255)  NOT NULL,
  instructor varchar(255)  NOT NULL,
  hours int  NOT NULL,
  meeting varchar(255)  NOT NULL,
  date varchar(255)  NOT NULL,
  time time  NOT NULL,
  room varchar(255)  NOT NULL,
  Faculty_Staff_Id int  NOT NULL,
  Student_Id int  NOT NULL,
  Student_netID varchar(255)  NOT NULL,
  CONSTRAINT Course_pk PRIMARY KEY (crn)
);
```

1.1.23. Student

```
-- tables
-- Table Student
CREATE TABLE Student (
  Id int  NOT NULL,
  netID varchar(255)  NOT NULL,
  password varchar(255)  NOT NULL,
  firstName varchar(255)  NOT NULL,
  lastName varchar(255)  NOT NULL,
  emailAddress varchar(255)  NOT NULL,
  Course_crn varchar(255)  NOT NULL,
  CONSTRAINT Student_pk PRIMARY KEY (Id,netID)
);
```

1.1.24. Faculty_Staff

```
-- tables
-- Table Student
CREATE TABLE Student (
  Id int  NOT NULL,
  netID varchar(255)  NOT NULL,
  password varchar(255)  NOT NULL,
  firstName varchar(255)  NOT NULL,
  lastName varchar(255)  NOT NULL,
  emailAddress varchar(255)  NOT NULL,
  Course_crn varchar(255)  NOT NULL,
  CONSTRAINT Student_pk PRIMARY KEY (Id,netID)
);
```

1.1.25. Student Database

The “studentDatabase” class will securely hold currents students information which includes, “firstName”, “lastName”

1.1.26. Course Database

The “courseDatabase” class will store information about academic schedule that can be retrieved by a logged in student.

1.1.27. Map Database

The “Map” class will store information about maps at the Kennesaw and Marietta campuses of Kennesaw State University. Student accessing the Map database will be able to view stored event description

1.1.28. Calendar Database

The “Calendar” class will store information about events that are going to be held on the Marietta and Kennesaw campuses of Kennesaw State University

Data Dictionary

1.1.29. Academic History

Class name: Academic History	
Brief description: Allows the student to look up previous semester’s academic history.	
Attributes (fields)	**Attribute Description**
String semester	Stores the semester which was selected.
String courseName	Stores the course CRN number taken.
String courseTitle	Stores the name of the course.
String instructor	Stores the instructor of that course.
int hours	Stores the number of hours from the course.
char grade	Stores the grade that was achieved from that course.
Methods (operations)	**Method Description**
public String getSemester()	Returns the semester selected by the student
public String getcourseName()	Return the course crn number of the student’s class.
public String getCourseTitle()	Returns the course title of the student’s class
public String getInstructor()	Returns the instructor of the student’s classes.
public int getHours()	Returns the hours of the course for the student’s class
public char getGrade()	Returns the grade of the course the student has taken.
public void isLoggedIn()	Computer will check the server if a secure connection is

	establish or not. If false, page will redirect you to login page.

1.1.30. Add/Drop Courses

Class name: Add/Drop Courses	
Brief description: Allows the student to add or drop their academic schedule of the current semester.	
Attributes (fields)	**Attribute Description**
String courseName	Stores the CRN of the student's class
String courseTitle	Stores the full course name of the class
String campus	Stores the campus where the class is located
String instructor	Stores the instructor that teaches the classs
int hours	Stores the number of hours of the class
String meeting	Stores whether the class is a lecture, online, or hybrid
String date	Stores the dates when the class will start and end
String days	Stores the days the class takes place.
String time	Stores the time the class takes place.
String location	Stores the building where the class takes place.
String room	Stores the room number where the class takes place.
Methods (operations)	**Method Description**
public String getCourseName()	Returns the CRN number of the course.
public String getCourseTitle()	Returns the course title of the student's classes.
public String getCampus()	Returns either if the campus is Kennesaw Or Marietta.
public String getInstructor()	Returns the instructor of the student's classes.
public int getHours()	Returns the hours for each of the student's classes.
public String getMeeting()	Returns whether the class is lecture, online, or hybrid.
public String getDate()	Returns the date when the class starts and ends.
public String getDays()	Returns the days the class takes place.
public String getTime()	Returns the time when the class takes place.

public String getLocation()	Returns the building of the class.
public String getRoom()	Returns the room number of the class.

1.1.31. Calendar

Class name: Calendar	
Brief description: A list of dates with the different events throughout Kennesaw State University.	
Attributes (fields)	**Attribute Description**
String name	Stores the name of the event.
String eventDescription	Stores the event's description.
String date	Stores the date of the event.
Date startTime	Stores the start time of the event.
Date endTime	Stores the end time of the event.
Methods (operations)	**Method Description**
public String getName()	Returns the name of the event.
public String getEventDescription()	Returns the event description.
public Date getDate()	Returns the date of the event.
public Date getStartTime()	Returns the start time of the event.
public Date getEndTime()	Returns the end time of the event.
public void createEventToGoogle()	Creates an event in the student's google account if they have of the event.
public bool isLoggedIn()	Computer will check the server if a secure connection is establish or not. If false, page will redirect you to login page.

1.1.32. Calendar Database

Class name: Calendar Database	
Brief description: Database which consists of information about events at KSU only accessed by an IT Admin.	
Attributes (fields)	**Attribute Description**
String name	Stores the name of the event.
String eventDescription	Stores the description about the event.

Date date	Stores the month/day/year of which the event is.
Date startTime	Stores the start time of the event.
Date endTime	Stores the end time of the event.
Methods (operations)	**Method Description**
public String getName()	Returns the name of the event.
public void setName(String name)	Sets the name of the event in the database.
ublic String getEventDescription()	Returns the event description of the event.
lic String setEventDescription(String eventDescription)	Sets the event description in the database.
public Date getDate()	Returns the event date of the event.
public Date setDate(Date date)	Sets the date of the event in the database.
public Date getStartTime()	Returns the start time of the event.
lic Date setStartTime(Date startTime)	Sets the start time of the event in the database.
public Date getEndTime()	Returns the end time of the event.
blic Date setEndTime(Date endTime)	Sets the end time of the event in the database.

1.1.33. Course Database

Class name: Course Database	
Brief description: Allows the student to look up their academic schedule of the current semester.	
Attributes (fields)	**Attribute Description**
String courseName	Stores the CRN of the student’s class
String courseTitle	Stores the full course name of the class
String campus	Stores the campus where the class is located
String instructor	Stores the instructor that teaches the classs
int hours	Stores the number of hours of the class
String meeting	Stores whether the class is a lecture, online, or hybrid
String date	Stores the dates when the class will start and end

String days	Stores the days the class takes place.
String time	Stores the time the class takes place.
String location	Stores the building where the class takes place.
String room	Stores the room number where the class takes place.
Methods (operations)	**Method Description**
public String getCourseName()	Returns the CRN number of the course.
public String getCourseTitle()	Returns the course title of the student’s classes.
public String getCampus()	Returns either if the campus is Kennesaw Or Marietta.
public String getInstructor()	Returns the instructor of the student’s classes.
pulbic int getHours()	Returns the hours for each of the student’s classes.
public String getMeeting()	Returns whether the class is lecture, online, or hybrid.
public String getDate()	Returns the date when the class starts and ends.
public String getDays()	Returns the days the class takes place.
public String getTime()	Returns the time when the class takes place.
public String getLocation()	Returns the building of the class.
public String getRoom()	Returns the room number of the class.

1.1.34. Faculty Database

Class name: Faculty Database	
Brief description: Database which consists of information about the professors only accessed by an IT Admin.	
Attributes (fields)	**Attribute Description**
String firstName	Stores the first name of the professor.
String lastName	Stores the last name of the professor.
String title	Stores the professor’s title.
String emailAddress	Stores the professor’s email address.
String phoneNumber	Stores the professor’s phone number.
String department	Stores the department of the professor.

String officeNumber	Stores the professor’s office number.
Methods (operations)	**Method Description**
public String getFirstName()	Returns the first name of the professor in the database.
public String getLastName()	Returns the last name of the professor in the database.
public String getTitle()	Returns the title of the professor in the database.
public String getPhoneNumber()	Returns the phone number of the professor in the database.
public String getDepartment()	Returns the department of the professor in the database.
public String getOfficeNumber()	Returns the office number of the professor in the database.

1.1.35. Faculty Directory

Class name: Faculty Directory	
Brief description: A directory of the list of professors attending Kennesaw State Univeristy.	
Attributes (fields)	**Attribute Description**
String firstName	Stores the first name of the professor.
String lastName	Stores the last name of the professor.
Methods (operations)	**Method Description**
public String getFirstName()	Returns the first name of the professor.
public String getLastName()	Returns the last name of the professor.
ic void openFacultyInformation(firstName, lastName)	Opens the professor profile of the selected professor.
public bool isLoggedIn()	Computer will check the server if a secure connection is establish or not. If false, page will redirect you to login page.

1.1.36. Faculty Information

Class name: Faculty Information	
Brief description: Information regarding about the faculty and staff members.	
Attributes (fields)	**Attribute Description**
String firstName	Stores the professor’s first name.

String lastName	Stores the professor’s last name.
String title	Stores the professor’s title.
String emailAddress	Stores the professor’s email address.
String phoneNumber	Stores the professor’s phone number.
String department	Stores the department the professor works at.
String officeNumber	Stores the office number of the professor.
Methods (operations)	**Method Description**
public String getFirstName()	Returns the first name of the professor.
public String getLastName()	Returns the last name of the professor.
public String getEmailAddress()	Returns the email address of the professor.
public String getPhoneNumber()	Returns the phone number to contact the professor.
public String getDepartment()	Returns the department the professor is currently in.
public String getOfficeNumber()	Returns the office number of the professor.
public void returnToDirectory()	the student back to professor directory to look up more professor or return to the main menu.
public bool isLoggedIn()	Computer will check the server if a secure connection is establish or not. If false, page will redirect you to login page.

1.1.37. Login

Class name: Login	
Brief description: Allows the user to gain access to their owl express account by entering their username and password.	
Attributes (fields)	**Attribute Description**
String username	Stores the student’s username as a string.
String password	Stores the user’s password for validation.
Methods (operations)	**Method Description**
public String getUserName()	Returns the student’s username.
public void setUsername(String userName)	Sets the student’s username for login.

public String getPassword()	Returns the student’s password.
public void setPassword(String password)	Sets the student’s password for login.
public bool login()	User attempts to log into their account using their username and password.
public bool validate()	Server checks the username and password to see if they match in a database.

1.1.38. Logout

Class name: Logout	
Brief description: Closes access to a user’s account information and permissions.	
Attributes (fields)	**Attribute Description**
Date loginDateTime	Stores the student’s login date time in the format h/m/s.
Date currentDateTime	Stores the student’s current date time in the format h/m/s.
Methods (operations)	**Method Description**
public void isLoggedIn()	Computer will check the server if a secure connection is establish or not. If false, page will redirect you to login page.
public void logout()	Can only be used if is LoggedIn returns true.
ate checkInactivityTime(Date loginDateTime, Date currentDateTime)	ill take in the parameter with the variables, loginDateTime and currentDateTime. If the t is idle from the application after so much time, for security reasons the user will be automatically logged out.

1.1.39. Map

Class name: Map	
Brief description: Visual graphics of both the Kennesaw and Marietta campus which shows all the buildings and any public places around the campuses.	
Attributes (fields)	**Attribute Description**
String campus	Stores which campus the student selects: Kennesaw or Marrietta.
Methods (operations)	**Method Description**
public String getCampus()	Returns the campus selected by the student.
public void getCampusMap(String campus)	Returns a map of the campus selected by the student.

public bool isLoggedIn()	Computer will check the server if a secure connection is establish or not. If false, page will redirect you to login page.

1.1.40. Map Database

Class name: Map Database	
Brief description: A database which holds all the information regarding the maps interface.	
Attributes (fields)	**Attribute Description**
String campus	Stores which campus the student selects: Kennesaw or Marietta.
Methods (operations)	**Method Description**
public String getCampus()	Returns the campus selected by the student.
ublic void setCampus(String campus)	Sets the campus in the database.
lic void getCampusMap(String campus)	Returns a map of the campus selected by the student.
lic void setCampusMap(String campus)	Sets the map of the campus in the database.

1.1.41. Schedule

Class name: Schedule	
Brief description: Allows the student to look up their academic schedule of the current semester.	
Attributes (fields)	**Attribute Description**
String courseName	Stores the CRN of the student’s class
String courseTitle	Stores the full course name of the class
String campus	Stores the campus where the class is located
String instructor	Stores the instructor that teaches the classs
int hours	Stores the number of hours of the class
String meeting	Stores whether the class is a lecture, online, or hybrid
String date	Stores the dates when the class will start and end
String days	Stores the days the class takes place.
String time	Stores the time the class takes place.

String location	Stores the building where the class takes place.
String room	Stores the room number where the class takes place.
Methods (operations)	**Method Description**
public String getCourseName()	Returns the CRN number of the course.
public String getCourseTitle()	Returns the course title of the student’s classes.
public String getCampus()	Returns either if the campus is Kennesaw Or Marietta.
public String getInstructor()	Returns the instructor of the student’s classes.
public int getHours()	Returns the hours for each of the student’s classes.
public String getMeeting()	Returns whether the class is lecture, online, or hybrid.
public String getDate()	Returns the date when the class starts and ends.
public String getDays()	Returns the days the class takes place.
public String getTime()	Returns the time when the class takes place.
public String getLocation()	Returns the building of the class.
public String getRoom()	Returns the room number of the class.
public void createScheduleOnPhone()	Creates an image of your schedule that can be accessed within your photos.
public bool isLoggedIn()	Computer will check the server if a secure connection is establish or not. If false, page will redirect you to login page.

1.1.42. Search

Class name: Search	
Brief description: Allows the user to search for a specific class through the course number.	
Attributes (fields)	**Attribute Description**
String courseName	Stores the CRN of the student’s class
Methods (operations)	**Method Description**
public String getCourseName()	Returns the CRN number of the course.
public String getCourseTitle()	Returns the course title of the student’s classes.
public String getCampus()	Returns either if the campus is Kennesaw Or Marietta.

public String getInstructor()	Returns the instructor of the student’s classes.
public int getHours()	Returns the hours for each of the student’s classes.
public String getMeeting()	Returns whether the class is lecture, online, or hybrid.
public String getDate()	Returns the date when the class starts and ends.
public String getDays()	Returns the days the class takes place.
public String getTime()	Returns the time when the class takes place.
public String getLocation()	Returns the building of the class.
public String getRoom()	Returns the room number of the class.

1.1.43. Semester

Class name: Semester	
ief description: Allows the student to select a previous semester for academic history, or a current semester for their academic schedule.	
Attributes (fields)	**Attribute Description**
String[] semester	An array of previous semester the student has been at KSU
String semesterSelected	Stores the semester selected as a string.
Methods (operations)	**Method Description**
public String[] semester	Returns an array of all the semester the students has been at KSU.
public String getSemesterSelected()	Returns the semester selected by the student.
c void submitSemester(String semesterSelected)	redirected to a page where it will display either their academic history or their academic schedule based on their current semester selected.

1.1.44. Student Database

Class name: Student Database	
Brief description: Information regarding about the students.	
Attributes (fields)	**Attribute Description**
String ksuID	Stores the student’s ksuID.
String netID	Stores the student’s netID.
String password	Stores the student’s password.

String firstName	Stores the student’s first name.
String lastName	Stores the student’s last name.
String email	Stores the student’s email address.
Methods (operations)	**Method Description**
public String getKsuId()	Returns the KSU id of the student.
public String getNetId()	Returns the Net id of the student.
public String getPassword()	Returns the password of the student.
public String getFirstName()	Returns the first name of the student. .
public String getLastName()	Returns the last name of the student..
public String getEmailAddress()	Returns the email address of the student. .

Non-Functional

A student log in is required for access to any information in the application. This requirement is not necessary when accessing the Kennesaw State website directly, however when using a mobile application as a safety precaution we require a student or administrator log in. This ensures consistency and limits application users to students and administration only.

Appendix

A

Academic 28
Academic History 2, 3, 12, 46
application 56

C

Calendar 2, 3, 6, 16, 20, 29, 36, 47, 48
Communication 2, 8
Course 25, 26
Course Database 2, 3, 20, 49

D

Data 3, 46
Database 2, 3, 20, 40, 45, 48, 50, 53
dd/Drop Courses 2
Directory 30, 34

F

Faculty 30, 32, 34
Faculty and Staff 30, 32, 34
Faculty Information 2, 3, 18, 51

H

Hardware 2, 8

I

Interfaces 2, 6, 9

K

Kennesaw 56
KSU 1, 4, 6, 7, 8, 19, 48, 56

L

Login 2, 3, 11, 18, 23, 32, 38, 52
Logout 2, 3, 11, 24, 33, 39, 52

M

Map 2, 3, 17, 20, 31, 35, 53

O

Object 2, 3, 9, 40, 44

S

Schedule 2, 3, 6, 13, 27, 54
Search 2, 3, 15, 25, 34, 55
Semester 2, 3, 12, 55
Sequence 3, 23, 32, 37
Software 1, 2, 4, 8
Student 23, 44, 56
Student Database 2, 3, 19, 20, 56
System 2, 6, 9, 11

U

user 6, 7, 8, 11, 16, 17, 20, 52, 53, 55