**Software Design Document**

**AAU - Gold Team**

**Version: 1.0**

**Date: 4/25/2017**

**Revision History**

**Document Owner: Casey Boatman, Justin Andras**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| **04/25/2017** | **1.0** | **Document creation** | **Casey Boatman, Justin Andras** |
| **04/25/2017** | **1.0** | **Final Document Approval** | **Philip Parker, PM** |

**1 Introduction**

**1.1 Purpose**

**1.2 Scope**

**1.3 Definitions, acronyms, and abbreviations**

**2 References**

**3 Decomposition description**

**3.1 Module decomposition**

**3.2 Concurrent process decomposition**

**3.3 Data Decomposition**

**4 Dependency description**

**4.1 Intermodule Dependencies**

**4.2 Data Dependencies**

**5 Interface description**

**5.1 Login To University Website**

**5.2 Course Search**

**5.3 Directory Search**

**6 Detailed design**

**6.1 Module Detailed Design**

**6.1.1 Student Design**

**6.1.2 User Interface Design**

**6.1.3 Professor Design**

**6.2 Data Detailed Design**

**6.2.1 Students Detail**

**6.2.2 Students Enrollment Detail**

**6.2.3 Professors Detail**

**6.2.4 Courses Offered Detail**

**6.2.5 Degrees Detail**

**6.2.6 Departments Detail**

**6.2.7 Semesters Detail**

**6.2.8 Courses Detail**

**6.2.9 Course Categories Detail**

**6.2.10 Professor Employment Detail**

1. **Introduction**
   1. **Purpose**

This SDD provides the design details of the Gold Team’s semester project for the Above Average University Portal. The expected audience are the students in the Software Engineering course, Professor Zhang, and the Gold Team that developed and maintained the project.

* 1. **Scope**

This project was built for educational purposes in our Software Engineering course, it will not be for sale. AAU is a fictional university. The architecture was chosen as a learning opportunity. We did not want to go the direction of implementing this in a way that any of us have in previous courses (such as using JSPs like the majority of us did in Database Systems).

* 1. **Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| AAU | Above Average University |
| Gold Team | Team of developers working on this project. Includes:  Phil Parker  Ryan Peters  Jacob Hertl  Aaron Tonkovich  Casey Boatman  Justin Andras  Joshua Potrawski  Jonathan Nowak |
| Repo | Repository. Type of class in our project. |
| SDD | Software Design Document |

1. **References**

* Eric Braude, Michael Bernstein *Software Engineering: Modern Approaches*, John Wiley & Sons, 2011

1. **Decomposition description**

**3.1 Module decomposition**

Since our project only has one “package”, this SDD will break it down into types of classes

* 1. **Entry Point (.java file)**

The AauEntryPoint file is the heart of the project. Without this main file, nothing else will work. This is where everything for the project is setup, including things such as setting environment variables, instantiating the repo’s, created the JSONParser and sql2o objects, and more. This also defines all of the pages that are to be/can be navigated to.

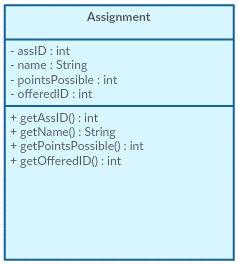
* 1. **Models (.java files)**

The Model classes are what hold our data. Data is returned from our database using a sql2o object, and is placed into the corresponding model. Section 3.3 of this document shows each model’s database counterpart.

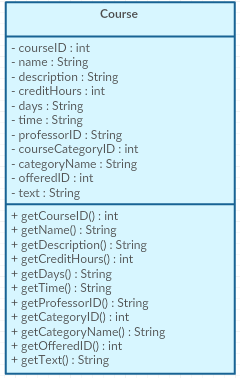
* + 1. **Announcement**



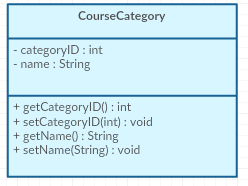
* + 1. **Assignment**



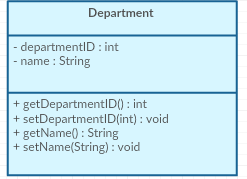
* + 1. **Course**



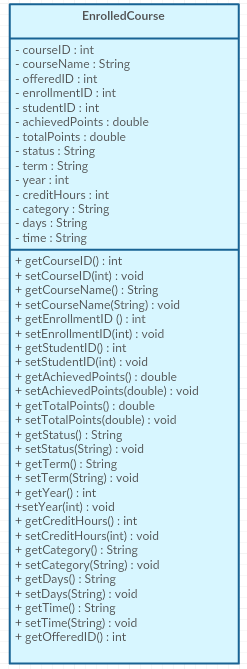
* + 1. **Course Category**



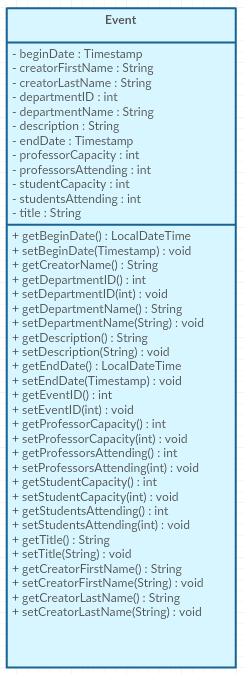
* + 1. **Department**



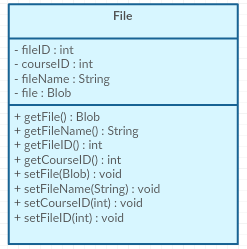
* + 1. **EnrolledCourse**



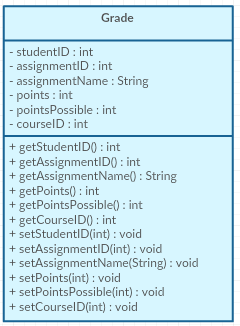
* + 1. **Event**



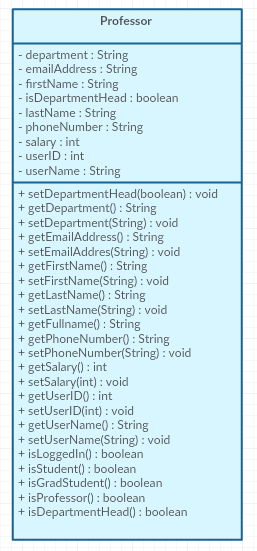
* + 1. **File**



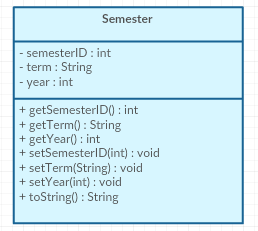
* + 1. **Grade**



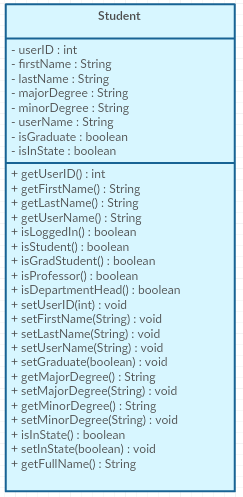
* + 1. **Professor**



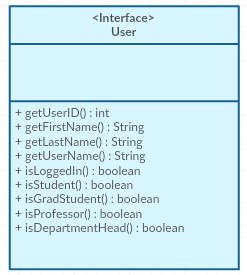
* + 1. **Semester**



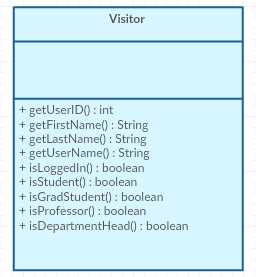
* + 1. **Student**



* + 1. **User**



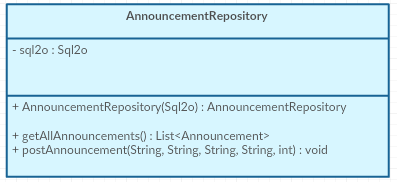
* + 1. **Visitor**



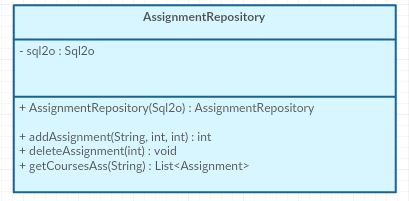
* 1. **Repos (.java files)**

The Repository classes are what fetch data from the database that are then put into the model classes. Repositories house multiple get methods to grab the data different ways.

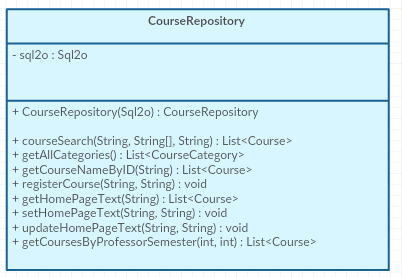
* + 1. **AnnouncementRepository**



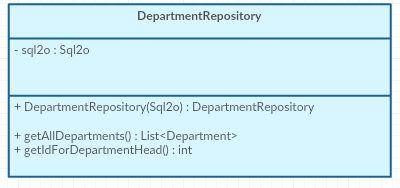
* + 1. **AssignmentRepository**



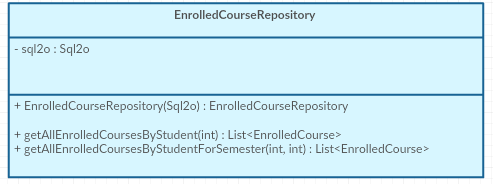
* + 1. **CourseRepository**



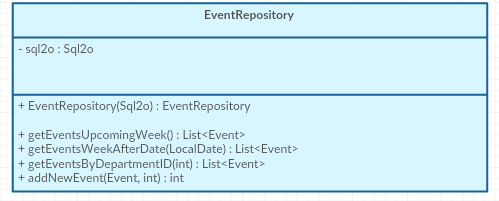
* + 1. **DepartmentRepository**



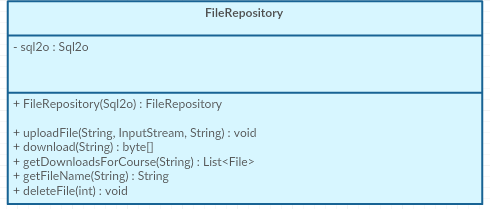
* + 1. **EnrolledCourseRepository**



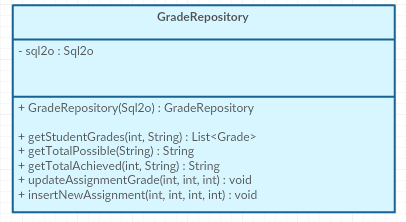
* + 1. **EventRepository**



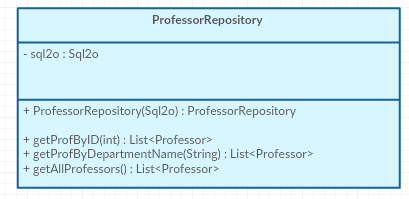
* + 1. **FileRepository**



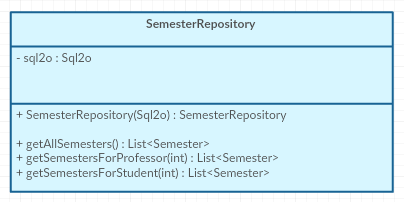
* + 1. **GradeRepository**



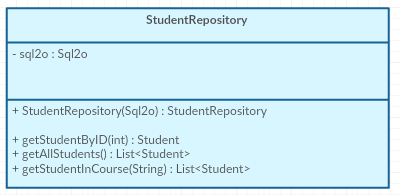
* + 1. **ProfessorRepository**



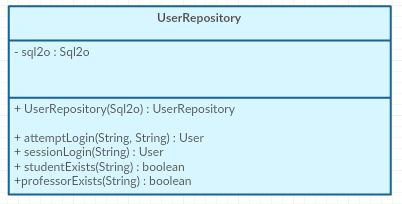
* + 1. **SemesterRepository**



* + 1. **StudentRepository**



* + 1. **UserRepository**



* 1. **Templates (.ftl files)**
     1. **Base**
     2. **Course-details**
     3. **Course-search**
     4. **directorySearch**
     5. **faculty**
     6. **homepage**
     7. **studentPortal**

**3.2 Concurrent process decomposition**We have a backend process that spawns a new thread for each http request to render output data.

**3.3 Data Decomposition**

We use repositories that depend on the model classes they export.

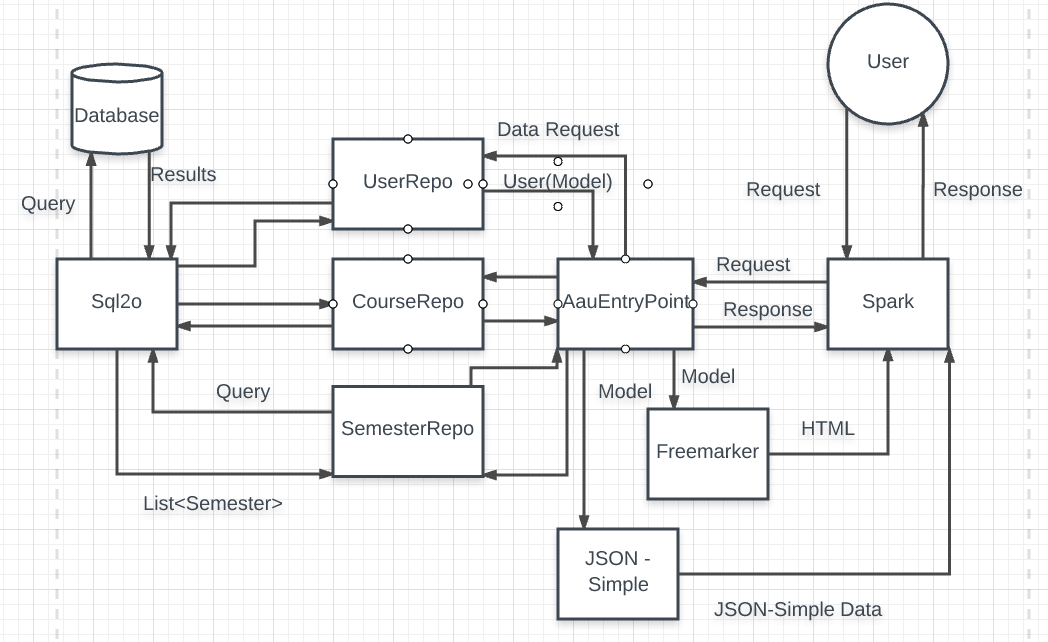
|  |  |
| --- | --- |
| **Model** | **Database Counterpart** |
| Announcement | Faculty\_Announcements |
| Assignment | Assignments |
| Course | Courses |
| CourseCategory | Course\_Categories |
| Department | Departments |
| EnrolledCourse | Student\_Enrollment |
| Event | Events |
| File | Course\_Files |
| Grade | Grades |
| Professor | Professors |
| Semester | Semesters |
| Student | Students |
| User | Professors & Students |
| Visitor | Just used to replace null references |

1. **Dependency description**

**4.1 Intermodule Dependencies**

**Our project only has one package, so it is only dependent among each breakdown of classes which was explained above**

**4.2 Data Dependencies**



1. **Interface description**

In this section, each user interface is explained along with some screenshots.

**5.1 Login To University Website**

The university login interface is explained here.

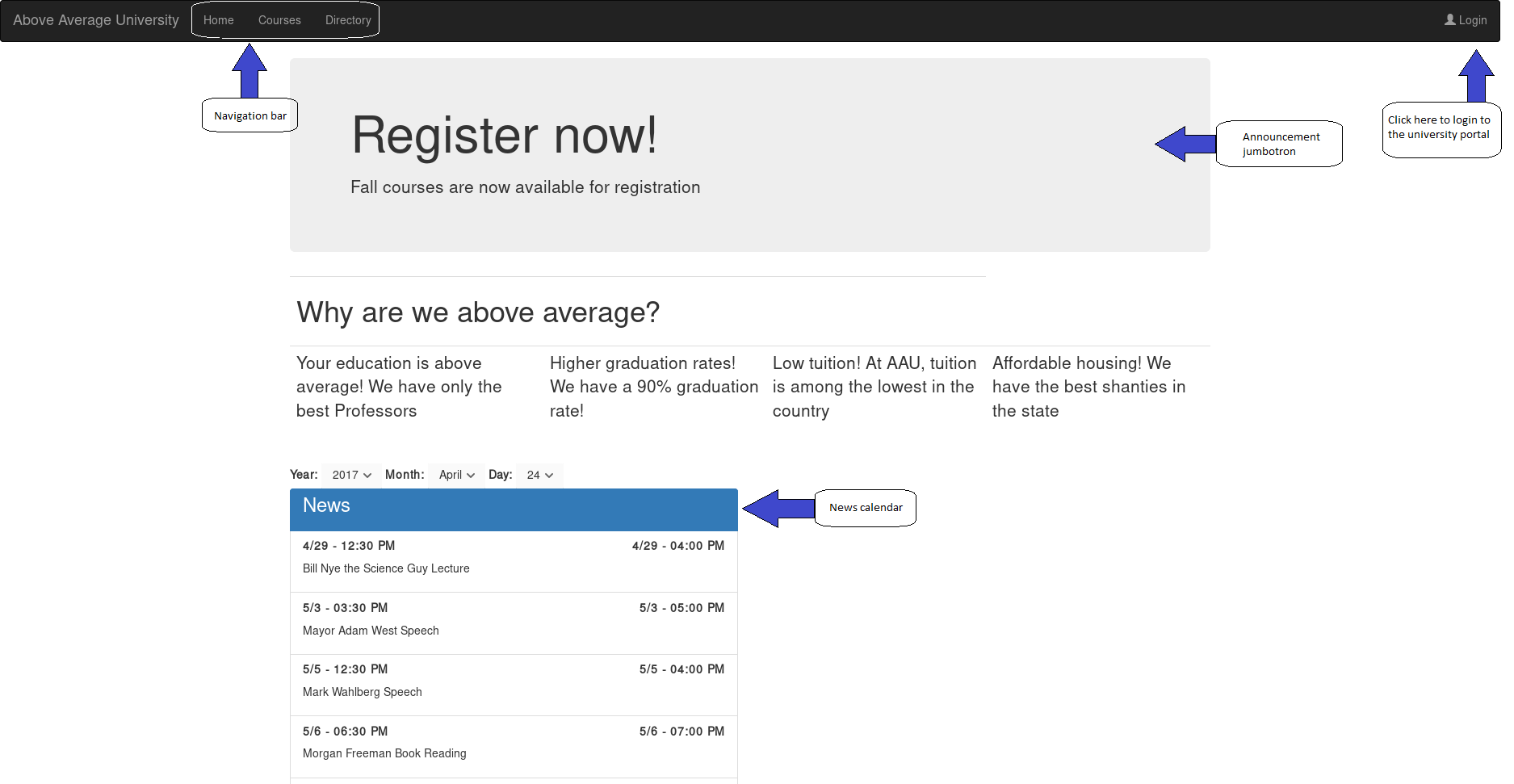


Figure 1: University homme page

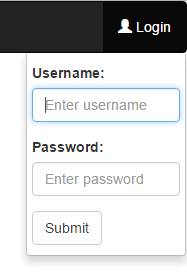


Figure 2: Login dropdown

The top right of the website home page contains a link and dropdown to login to the university portal. The user is prompted for a username and password. If the credentials are incorrect the user will be notified and not logged into the system. If the login credentials are correct the user will be directed to the User Portal page. The layout of the page will vary depending on the type of user.

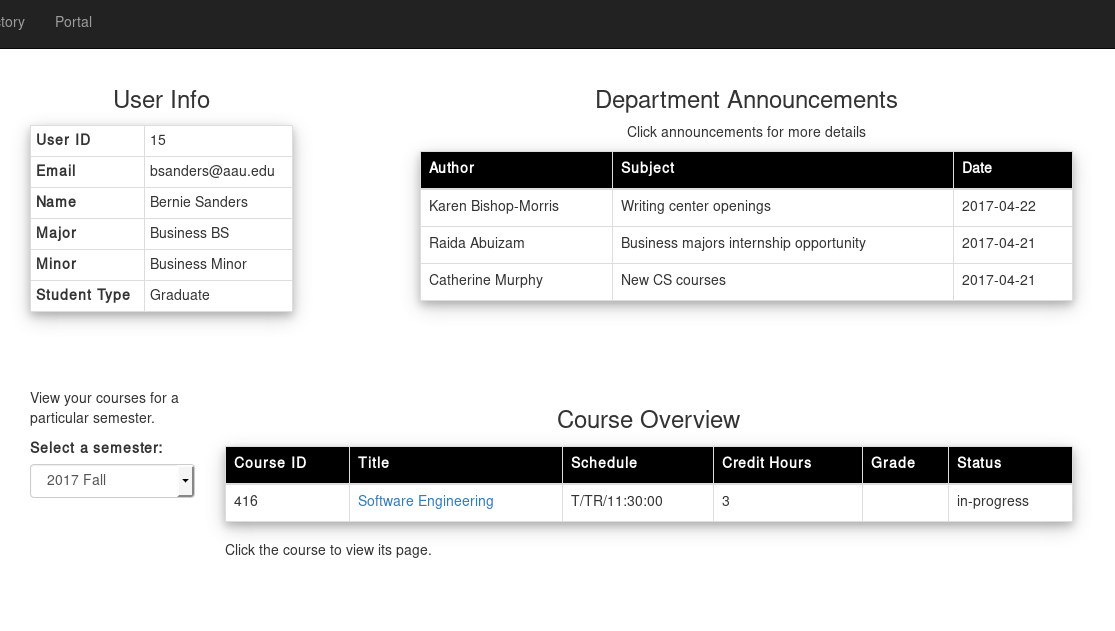


Figure 3: Student landing page

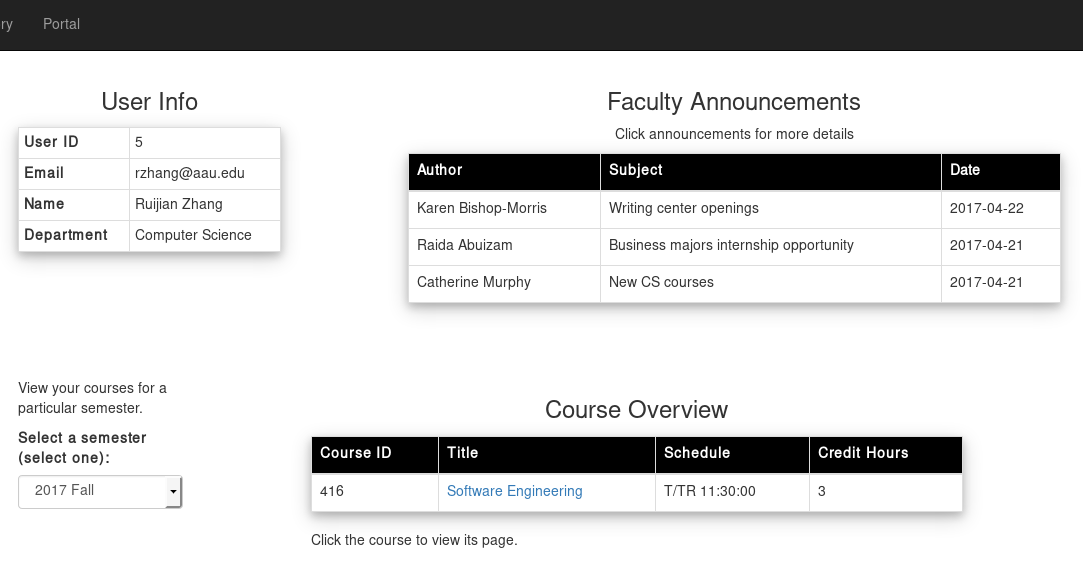


Figure 4: Faculty landing page

**5.2 Course Search**

The course search feature is described here.

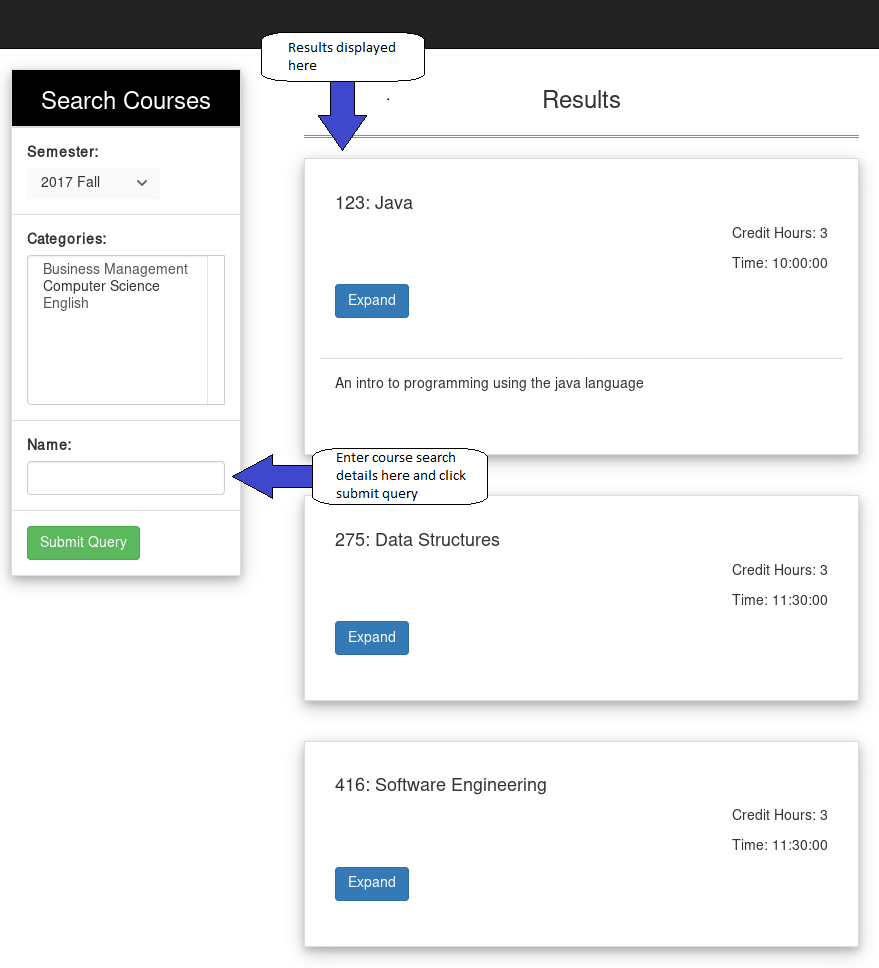


Figure 5: Course search page

By clicking the ***Course*** button in the top navigation bar a user can search for a specific course. The user will be asked to input the desired info into the ***Name*** field then click ***Submit Query***. A user may enter the course name or number in the search bar. The results matching the input data will be shown on the right half of the screen

**5.3 Directory Search**

The directory search feature is described here.

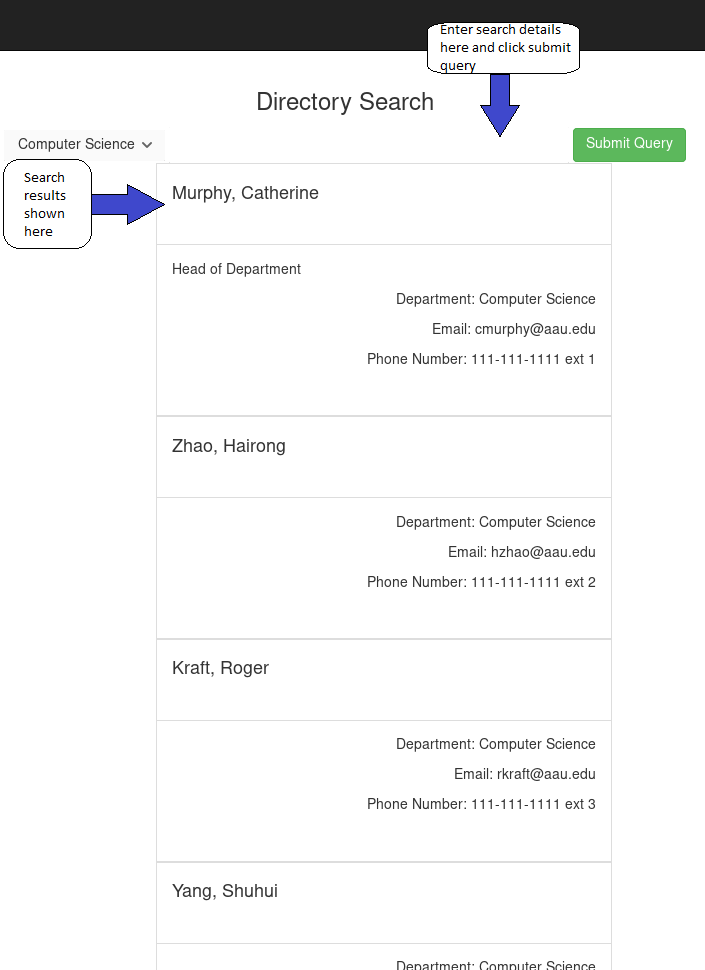
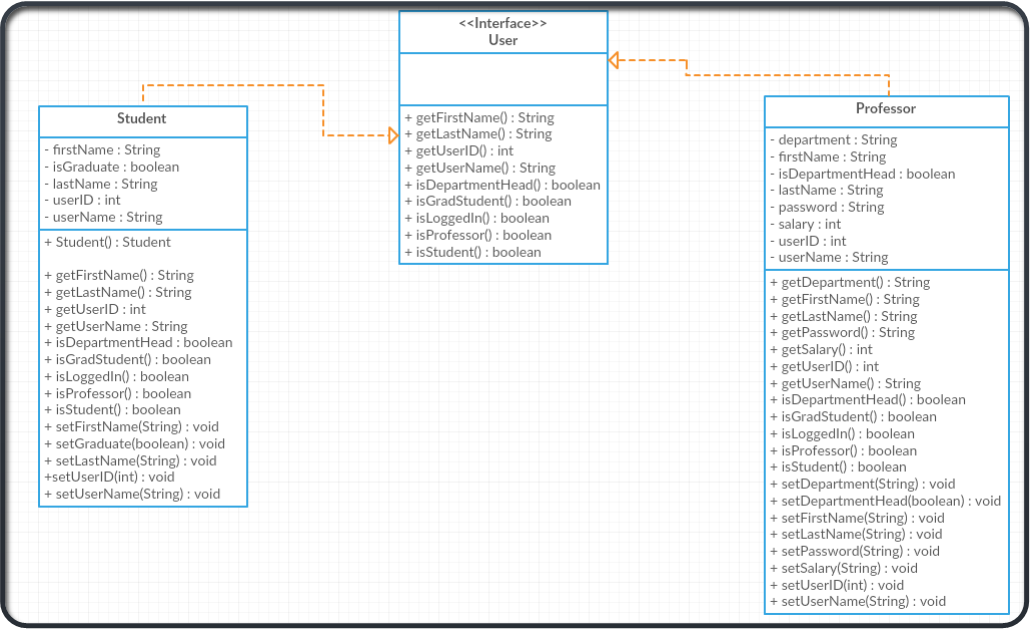


Figure 6: Directory search page

By clicking the ***Directory*** button in the top navigation bar a user can search for a specific faculty member. The user will be asked to input the desired info into the ***Name*** field then click ***Submit Query***. A user may enter the first or last name in the search bar. The results matching the input data will be shown under the search bar. The results will display the member’s department, email, and phone number.

1. **Detailed design**

**6.1 Module Detailed Design**



**6.1.1 Student Design**

* ***getFirstName( )*** - Returns student’s first name
* ***getLastName( )*** - Returns student’s last name
* ***getUserID ( )*** - Returns the student’s user ID
* ***getUserName ( )*** - Returns the student’s user name for university portal login
* ***isDepartmentHead ( )*** - Returns whether the student is a department head. This will always be false since a student can’t be a department head
* ***isGradStudent ( )*** - Returns whether the student is a Graduate student
* ***isLoggedIn ( )*** - Returns whether the student is logged into the university portal
* ***isProfessor ( )*** - Returns whether the student is a professor. This will always be false since a student can’t be a professor
* ***isStudent ( )*** - Returns whether the student is a student. This will always be true
* ***setFirstName (String)*** - Sets the student’s first name using the value defined as the input string
* ***setGraduate (boolean)*** - Sets whether the student is a graduate
* ***setLastName (String)*** - Sets the student’s last name using the value defined as the input string
* ***setUserID (int)*** - Sets the student’s user ID to the value defined as the input integer
* ***setUserName (String) -*** Sets the student’s user name using the value defined as the input string

**6.1.2 User Interface Design**

* ***getFirstName( )*** - Returns user’s first name
* ***getLastName( )*** - Returns user’s last name
* ***getUserID ( )*** - Returns the user’s user ID
* ***getUserName ( )*** - Returns the user’s user name for university portal login
* ***isDepartmentHead ( )*** - Returns whether the user is a department head
* ***isGradStudent ( )*** - Returns whether the user is a Graduate student
* ***isLoggedIn ( )*** - Returns whether the user is logged into the university portal
* ***isProfessor ( )*** - Returns whether the user is a professor
* ***isStudent ( )*** - Returns whether the user is a student.

**6.1.3 Professor Design**

* ***getDepartment ( )*** - Returns the professor’s assigned department
* ***getFirstName( )*** - Returns user’s first name
* ***getLastName( )*** - Returns user’s last name
* ***getPassword ( )*** - Returns the professor’s password
* ***getSalary ( )*** - Returns the professor’s salary
* ***getUserID ( )*** - Returns the professor’s user ID
* ***getUserName ( )*** - Returns the professor’s user name for university portal login
* ***isDepartmentHead ( )*** - Returns whether the professor is a department head
* ***isGradStudent ( )*** - Returns whether the professor is a Graduate student. This will always be false since a professor can’t be a Graduate student
* ***isLoggedIn ( )*** - Returns whether the professor is logged into the university portal
* ***isProfessor ( )*** - Returns whether the professor is a professor. This will always be true
* ***isStudent ( )*** - Returns whether the professor is a student. This will always be false
* ***setDepartment (String)*** - Sets the professor’s user name using the value defined as the input string
* ***setDepartmentHead (boolean)*** - Sets whether the professor is a department head
* ***setFirstName (String)*** - Sets the professor’s first name using the value defined as the input string
* ***setLastName (String)*** - Sets the professor’s last name using the value defined as the input string
* ***setPassword (String)*** - Sets the professor’s password using the value defined as the input string
* ***setSalary (String)*** - Sets the professor’s salary using the value defined as the input string
* ***setUserID (int)*** - Sets the professor’s user ID to the value defined as the input integer
* ***setUserName (String) -*** Sets the professor’s user name using the value defined as the input string

**6.2 Data Detailed Design**

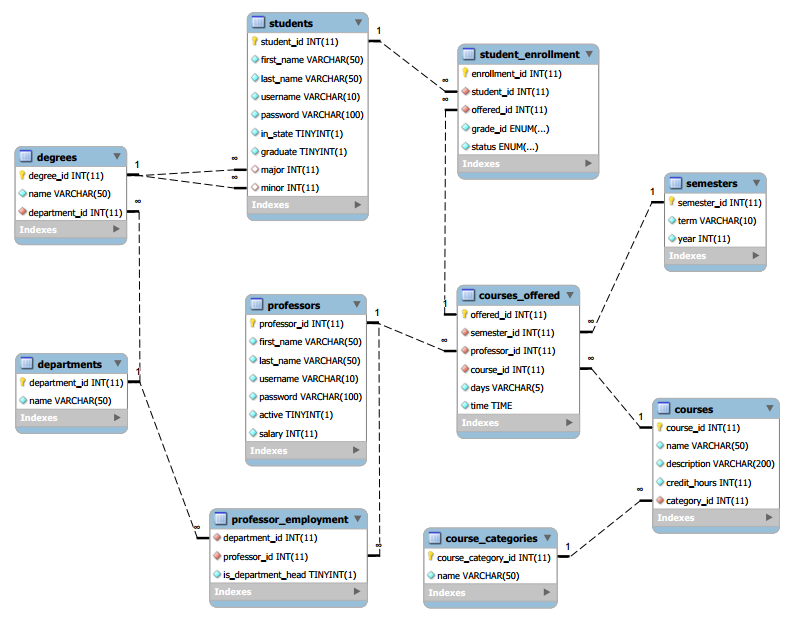


Figure 1: ER diagram

**6.2.1 Students Detail**

* ***studentId*** - Unique numeric identifier for each student
* ***first\_name*** - Student’s first name
* ***last\_name*** - Student’s last name
* ***username*** - Student’s user name used to login to the university portal
* ***password*** - Student’s password name used to login to the university portal
* ***in\_state*** - Integer identifying whether the student is an in-state student or not. This determines the amount of tuition the student pays to the university.
* ***graduate*** - Integer identifying whether the student is an undergraduate or graduate student.
* ***major*** - Integer determining the student’s major from a list of majors
* ***minor*** - Integer determining the student’s minor from a list of minors

**6.2.2 Students Enrollment Detail**

* ***enrollment\_id*** - A student’s enrollment identifier
* ***student\_id*** - Unique numeric identifier for each student
* ***offered\_id*** - Whether or not a course is offered a specific semester
* ***grade\_id*** - The student’s grade in a specific course
* ***status*** - Whether a student is enrolled in a specific course

**6.2.3 Professors Detail**

* ***professor\_id*** - Unique numeric identifier for each professor
* ***first\_name*** - Professor’s first name
* ***last\_name*** - Professor’s last name
* ***username*** - Professor’s user name used to login to the university portal
* ***password*** - Professor’s password name used to login to the university portal
* ***active*** - Whether or not the professor is actively teaching at the university
* ***salary*** - Professor’s salary at the university

**6.2.4 Courses Offered Detail**

* ***offered\_id*** - Whether or not a course is offered a specific semester
* ***semester\_id*** - The course’s identifier for the semester
* ***professor\_id*** - Unique numeric identifier for each professor
* ***course\_id*** - Unique numeric identifier for each course
* ***days*** - Days of the week the course is held
* ***time*** - Time the course is offered

**6.2.5 Degrees Detail**

* ***degree\_id*** - Unique numeric identifier for each type of degree
* ***name*** - Name of the degree
* ***department\_id*** - Unique numeric identifier for the department the degree is a part of

**6.2.6 Departments Detail**

* ***department\_id*** - Unique numeric identifier for the department the degree is a part of
* ***name*** - Name of the department

**6.2.7 Semesters Detail**

* ***semester\_id*** - Unique numeric identifier for the semester
* ***term*** - Which term the semester is throughout the year
* ***year*** - Calendar year

**6.2.8 Courses Detail**

* ***course\_id*** - Unique numeric identifier for each course
* ***name*** - Name of the course
* ***description*** - Description of the course
* ***credit\_hours*** - Course credit hours
* ***category\_id*** - The category of the course

**6.2.9 Course Categories Detail**

* ***category\_id*** - The category of the course
* ***name*** - Name of the course category

**6.2.10 Professor Employment Detail**

* ***department\_id*** - Unique numeric identifier for the department the degree is a part of
* ***professor\_id*** - Unique numeric identifier for each professor
* ***is\_department\_head*** - Whether or not the professor is the head of a department