ByteMeCollege

Dec 1, 2016

San Jose State University

https://github.com/brandonabajelo/ByteMeCollege.git

Project manager	
Project dates	Sep 1, 2016 - Nov 21, 2016
Completion	77%
Tasks	26
Resources	4

A web application where students can review courses.

Tasks

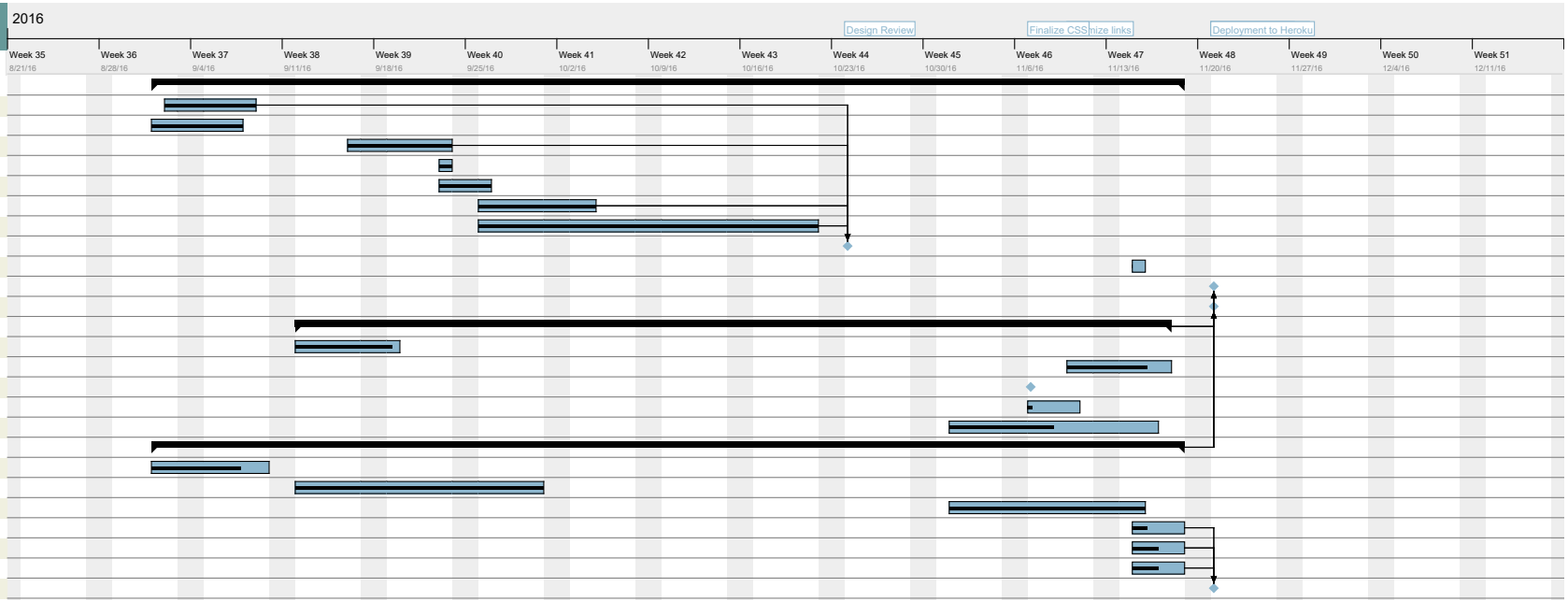
Name	Begin date	End date
Architectural design	9/1/16	11/18/16
Create draft of architecture	9/2/16	9/8/16
Functional navigation bar	9/1/16	9/7/16
Functional Specifications	9/16/16	9/23/16
Marketing Brochure	9/23/16	9/23/16
Concept Presentation	9/23/16	9/26/16
Conceptual design	9/26/16	10/4/16
Model-View-Controller architecture and data models	9/26/16	10/21/16
Design Review	10/24/16	10/24/16
Code Review	11/15/16	11/15/16
Application Demo	11/21/16	11/21/16
Deployment to Heroku	11/21/16	11/21/16
Front-end development	9/12/16	11/17/16
Optimize navigation bar	9/12/16	9/19/16
Optimize links	11/10/16	11/17/16
Finalize CSS	11/7/16	11/7/16
Potential JavaScript	11/7/16	11/10/16
About Us	11/1/16	11/16/16
Back-end development	9/1/16	11/18/16
Ruby on Rails	9/1/16	9/9/16
Database Design	9/12/16	9/30/16
User account database	11/1/16	11/15/16
Fill major database	11/15/16	11/18/16
Fill student review database	11/15/16	11/18/16
Fill department database	11/15/16	11/18/16
Fill courses database	11/21/16	11/21/16

Resources

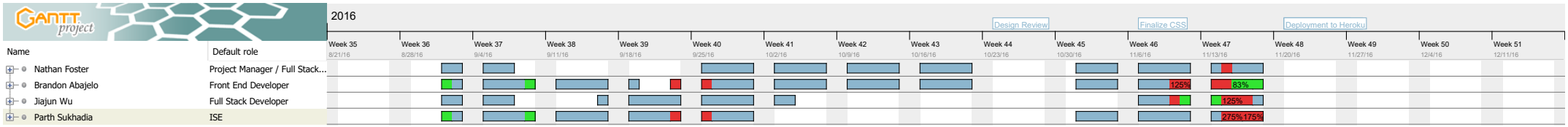
Name	Default role
Nathan Foster	Project Manager / Full Stack Developer
Brandon Abajelo	Front End Developer
Jiajun Wu	Full Stack Developer
Parth Sukhadia	ISE

Gantt Chart

Name	Begin date	End date
Architectural design	9/1/16	11/18/16
Create draft of architecture	9/2/16	9/8/16
Functional navigation bar	9/1/16	9/7/16
Functional Specifications	9/16/16	9/23/16
Marketing Brochure	9/23/16	9/23/16
Concept Presentation	9/23/16	9/26/16
Conceptual design	9/26/16	10/4/16
Model-View-Controller architecture a...	9/26/16	10/21/16
Design Review	10/24/16	10/24/16
Code Review	11/15/16	11/15/16
Application Demo	11/21/16	11/21/16
Deployment to Heroku	11/21/16	11/21/16
Front-end development	9/12/16	11/17/16
Optimize navigation bar	9/12/16	9/19/16
Optimize links	11/10/16	11/17/16
Finalize CSS	11/7/16	11/7/16
Potential JavaScript	11/7/16	11/10/16
About Us	11/1/16	11/16/16
Back-end development	9/1/16	11/18/16
Ruby on Rails	9/1/16	9/9/16
Database Design	9/12/16	9/30/16
User account database	11/1/16	11/15/16
Fill major database	11/15/16	11/18/16
Fill student review database	11/15/16	11/18/16
Fill department database	11/15/16	11/18/16
Fill courses database	11/21/16	11/21/16



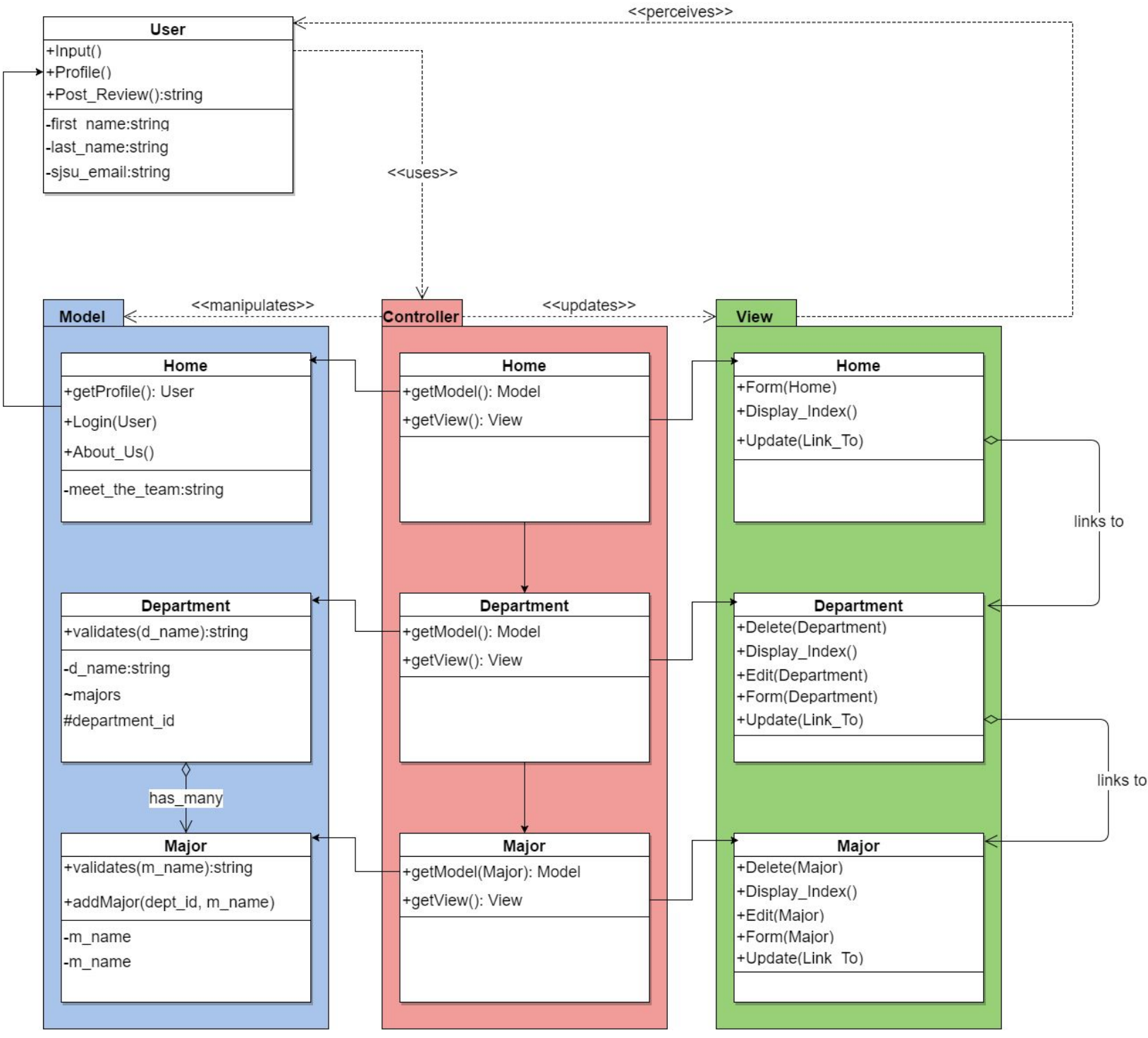
Resources Chart



byteMeCollege

MVC Architecture

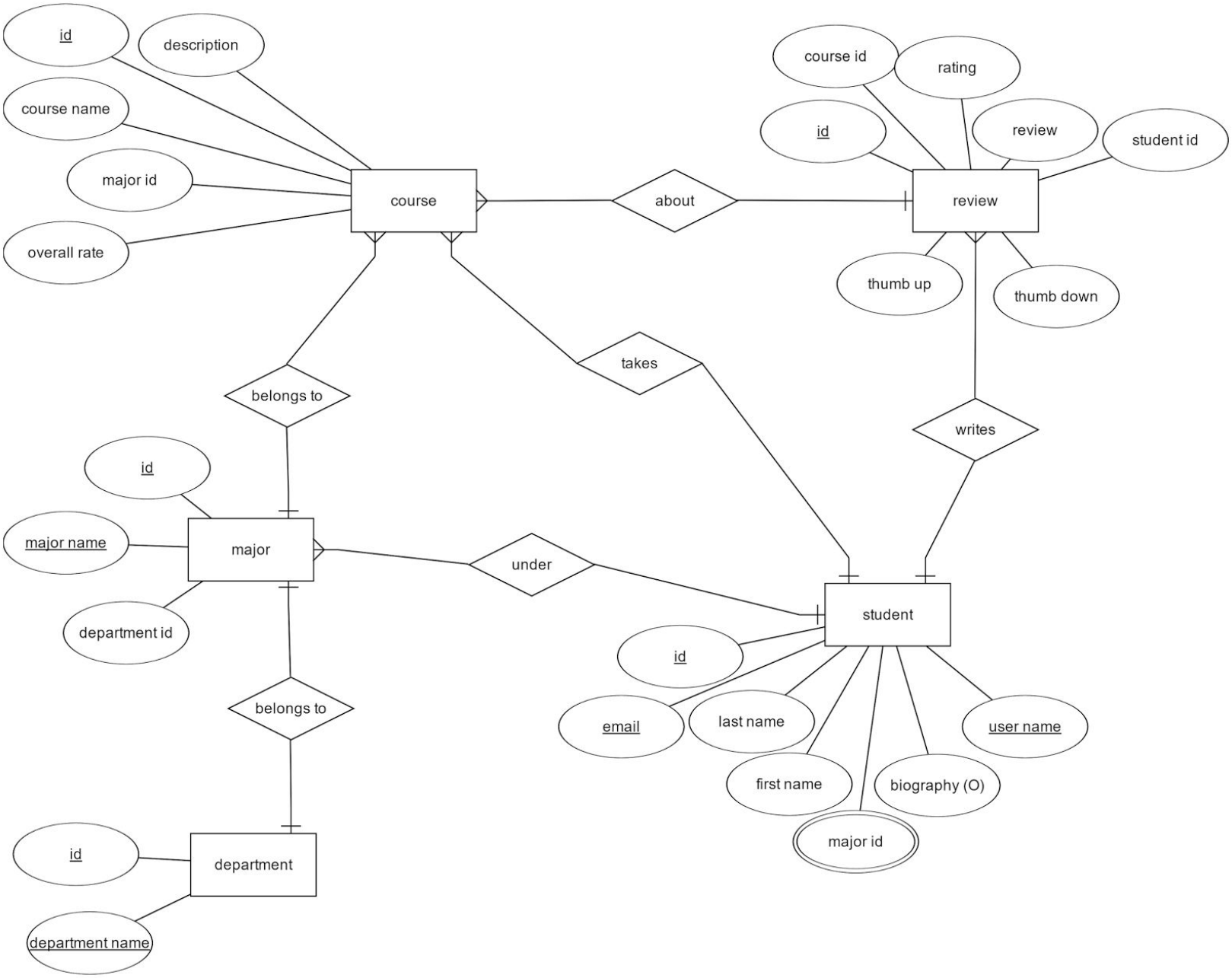
UML Package Class Diagram



Package Name	Package Class Name	Description
Model	Home	Contains user profile data
Model	Department	Contains department data and is related to the Major model
Model	Major	Contains major data and is related to the Department model
Controller	Home	Directs user to profile / profile creation and other linked pages
Controller	Department	Accesses the Department model data to be displayed
Controller	Major	Accesses the Major model data to be displayed
View	Home	Introduces website's purpose and directs user to profile and other pages
View	Department	Displays all the departments
View	Major	Displays all the majors

Database Architecture

Entity Relationship Diagram



Relational Schema Diagram

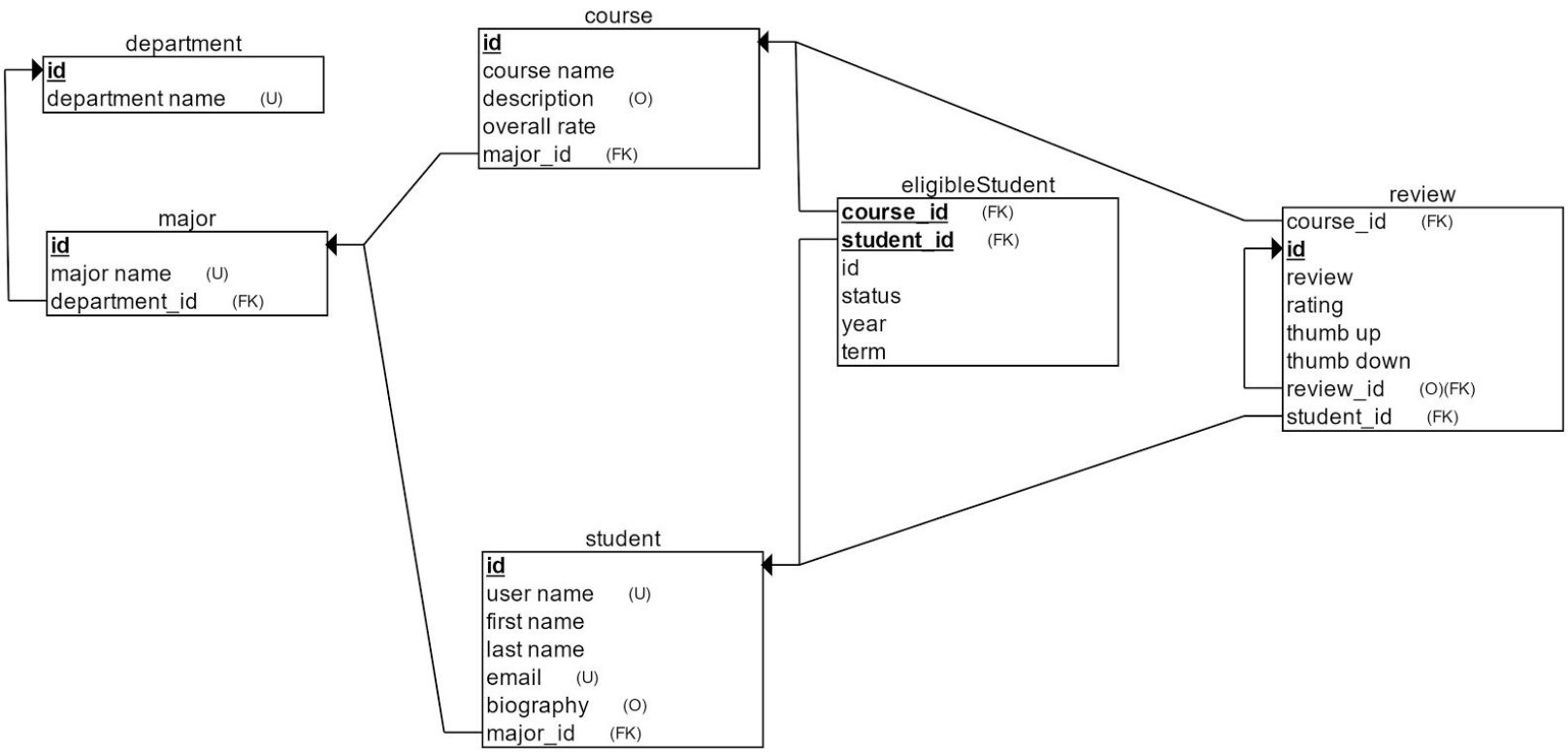
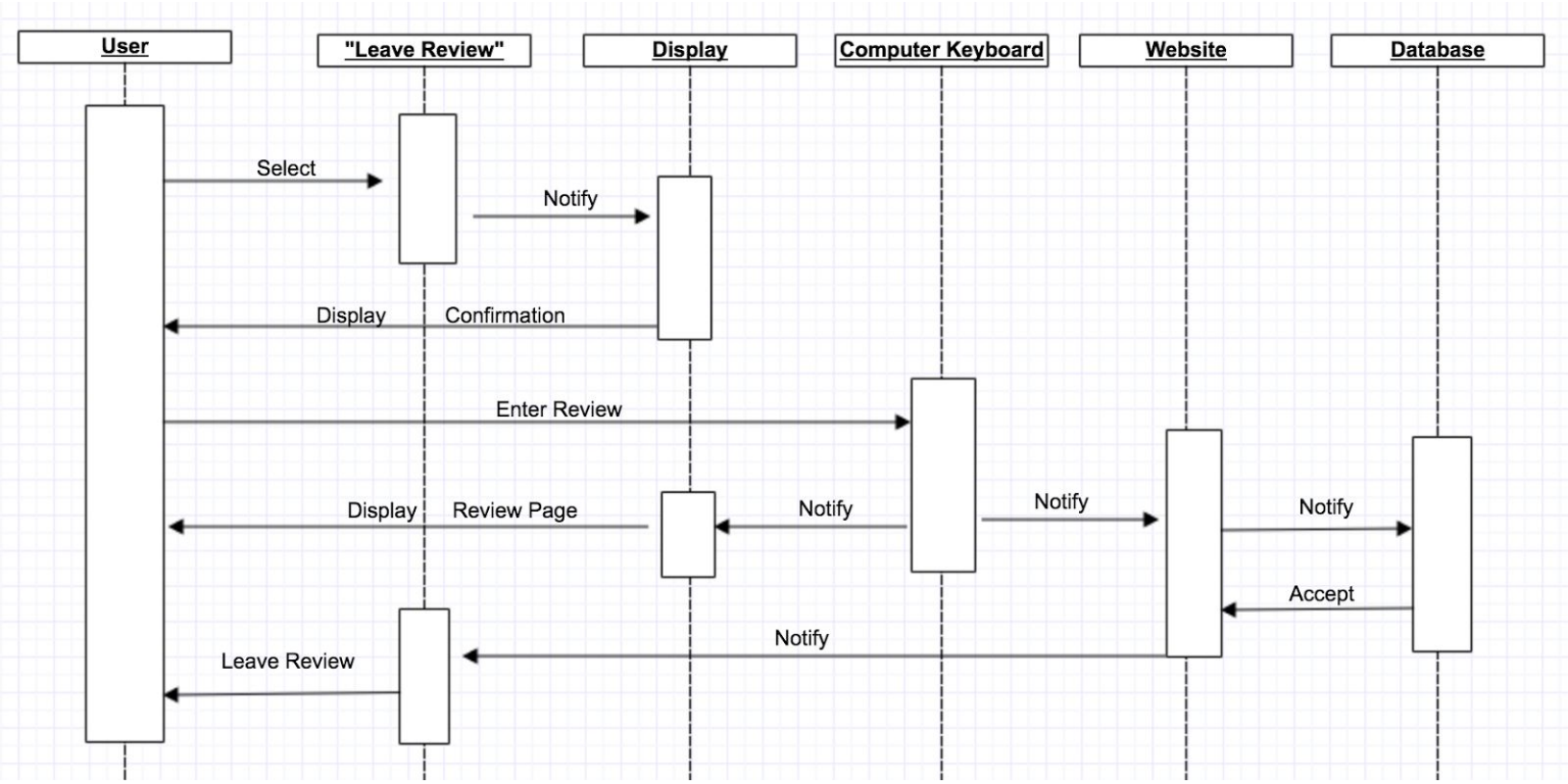


Table Name	Description
department	This table stores the different academic departments of a university
major	Each major under the corresponding academic department will be stored in this table
course	Each course relates to a particular major
review	This table corresponds to each review left by a user
student	Information regarding students is in this table
eligibleStudent	The purpose of this table is to deem a student eligible to leave a review or not. An eligible student will have either taken said course or currently be enrolled in it.

UML Sequence Diagram



Functional Specification

1. Product name: ByteMeCollege.com

2. Problem statement:

College students do not have enough information about the courses they take before enrolling. They need a platform to decide relevant courses to their academic interest from offered options. While choosing a university to attend, students don't have a one-stop-shop to compare similar programs offered by different academic institutions.

3. Product objectives

The main objective of ByteMeCollege.com is to provide user verified reviews of courses offered by different academic institutions (currently focusing on SJSU). We hope to help students by providing constructive information to help them choose the courses that are most relevant to their academic goals and interests. Students can also do real world comparisons of similar courses offered by various institutions to help them choose the right academic programs and most fitting career path for their educational goals.

4. Functional requirements

1. Administrators must enter departments, majors, courses.
2. Must students with SJSU email sign up for account.
3. Anybody must be able to read course reviews.
4. Registered members must be able to modify content.
5. Registered members must be able to leave a review.
6. Registered members shall request to add a course.
7. Registered members must be not able to modify review after 24 hours of posting.

5. Nonfunctional requirements

1. The server must be continuously running.
2. Website interface must be easy to use and navigate.
3. Reviews must be submitted by respective institution's students.

4. System shall send confirmation email to new users within 5 min of account creation.

6. Use cases

Use Case Description 1

Use Case name:	About us
Product name:	ByteMeCollege
Team:	ByteMeCollege
Date:	September 22, 2016

1. Goal

Provide user with ample amount of information about the website's purpose.

2. Summary

A user must quickly understand the purpose of the website and who the administrators are.

3. Actors

Actor 1: The user

Actor 2: Website's homepage

etc.

4. Preconditions

- The website server is running.
- The user has navigated to the website's homepage.
- The user navigates to the "about us" section on the website's homepage.

5. Trigger

The web application's homepage controller detects the user's HTTP request for access.

6. Primary Sequence

Step	Action
1	Homepage controller detects a user's request for access.
2	Homepage controller calls the homepage view.
3	Homepage index.html.erb file is rendered to the users.
4.	User scrolls down to the "about us" section of the rendered view.

7. Primary Post Conditions

- User is informed of the website's purpose.
- User is informed of the website's services.
- User is informed of who are the administrators of the website.

8. Alternate Sequences

Alternate Trigger

The application's homepage controller detects the user's HTTP request for access but then terminates.

Step	Action

1	An alternate controller detects a user's request for access.
2	The alternate controller calls its corresponding view.
3	The called view is rendered to the user.

Alternate Post Conditions

- The user quickly aborts access to the website.
- The user is ignorant of the website's purpose, services, and administrators.

Alternate Trigger

The web application's homepage controller detects the user's HTTP request outside its domain.

Step	Action
1	A controller detects a user's HTTP request for access to an external site.
2	The controller is no longer supporting the user's requests.

Alternate Post Conditions

- The user is no longer connected to the web application.
- The user is ignorant of the website's purpose, services, and administrators.

9. Nonfunctional Requirements

- The view pages must be rendered in less than 3 seconds.
- The web application shall run on Google Chrome, Internet Explorer, Microsoft Edge, Firefox, and Safari on any platform.
- The web application must be optimized for mobile devices.
- There must be a navigation bar at the top of every page view.

10. Glossary

Homepage controller = is routed as the root of the web application.

Alternate controller = any other controller other than the homepage controller (application, departments, or majors)

HTTP = Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems. It is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol to exchange or transfer hypertext.

Use Case Description 2

Use Case name:	Request to add a class
Product name:	ByteMeCollege
Team:	ByteMeCollege
Date:	September 22, 2016

1. Goal

User requests to add a class that isn't listed

2. Summary

A user who has logged in can send one or more requests to add a course to be reviewed that is not already listed.

3. Actors

Actor 1: User
Actor 2: System
etc.

4. Preconditions

·The server has been connected
·The user has navigated to the web page
·The user has logged in with their unique username and password
·The user has searched for a course and was unsuccessful in finding said course
etc.

5. Trigger

The user selects the “request to add course” option

6. Primary Sequence

Step	Action
1	The user clicks to request to add a course
2	The system checks whether or not the user has logged in
3	If the user is not logged in, the system displays message to log in
4	If the user is logged in, the system will check to see if course already exists

7. Primary Postconditions

·The user gets redirected to homepage

·

etc.

8. Alternate Sequences

Alternate Trigger

User submits a request for a course that does not exist

Step	Action
1	The user clicks to request to add a course
2	The system checks whether or not the user has logged in
3. *4.*	If the user is not logged in, the system displays message to log in If the user is logged in, the system will check to see if course already exists

Alternate Postconditions

·Admin returns a message stating that course does not exist

·User gets redirected to homepage

Alternate Trigger

User submits a request for a course that already exists

Step	Action
1	The user clicks to request to add a course
2	The system checks whether or not the user has logged in

3	If the user is not logged in, the system displays message to log in
4	If the user is logged in, the system will check to see if course already exists

Alternate Postconditions

·Admin return a message stating course already exists
·User gets redirected to homepage

9. Nonfunctional Requirements

·Friendly user interface
·Email to be sent within 5 min to user that request has been completed

10. Glossary

User = person who wants to submit a request to add a course
System = ByteMeCollege.com: a college course reviewer web application
Admin = Team who created ByteMeCollege.com

Use Case Description 3

Use Case name:	User does a class search
Product name:	ByteMeCollege.com

Team:	ByteMeCollege
Date:	9/23/2016

1. Goal

To locate a course on the website to read reviews.

2. Summary

User attempts to locate a course he/she intends to get information in search bar and reads reviews.

3. Actors

Actor 1: User
Actor 2: ByteMeCollege.com server

4. Preconditions

- Computer has been started up
- Computer is connected to internet.
- User is on ByteMeCollege.com page in an internet browser

5. Trigger

User clicks on the search bar

6. Primary Sequence

Step	Action
1	ypes course code or course related keywords in search bar.
2	er clicks on search button next to the search bar or hits enter. omputer communicates with server to search for results relevant to the query.
3	Computer shows relevant searches on the screen.
4	User clicks on the search result that matches with the course he intends to find information on.
5	Computer opens the course page and shows it on the screen.

7. Primary Postconditions

- User is able to read the reviews on the page
- User is able to rate reviews if he/she wishes.

8. Alternate Sequences

Alternate Trigger

Customer clicks on search button or hits enter.

Step	Action
1	Computer communicates with server to search for results relevant to the query but does not find a match
2	Computer shows “no match found” on screen asks user to modify query or submit a request to add desired course.

Alternate Postconditions

- User either modifies the query or submits a request to add course.

Alternate Trigger

Step	Action
1	
2	
etc.	

Alternate Postconditions

.
.
etc.

9. Nonfunctional Requirements

- System responds to each customer input within 3 seconds
- System displays content in English

10. Glossary

User: Person who is visiting ByteMeCollege.com site
Course: An educational class taught at the college or university

ByteMeCollege.com: A website that provides reviews on various college courses
System: Server on which ByteMeCollege.com is running on
Computer: A device that lets user interact with the system.

Use Case Description 4

Use Case name:	Adding course review
Product name:	ByteMeCollege
Team:	ByteMeCollege
Date:	September 23, 2016

1. Goal

Student add course review.

2. Summary

A logged in student is able to add only one review to multiple courses which are list on the system.

3. Actors

Actor 1: Students
Actor 2: ByteMeCollege website system

4. Preconditions

- ❑ A student must be logged in.
- ❑ The student has not added a review to a particular course.

5. Trigger

After the student finishes entering review, the student clicks “post” button.

6. Primary Sequence

Step	Action
1	The system lists previous review of the particular course.
2	If the student has not posted a review to the course, the system will display a posting text area for the student to write review and post it. It has a minimum number of character.
3	The student confirms and submits the review.
4	The system display all the reviews with the one just submitted.

7. Primary Postconditions

The student will be able to view the post, it also can be modify within 24 hours.

8. Alternate Sequences

Alternate Trigger

- ❑ The student clicks “post” button.

Step	Action
1	The student did not enter enough desired number of character
Alternate Postconditions	
	The system will not save the review that the student entered.

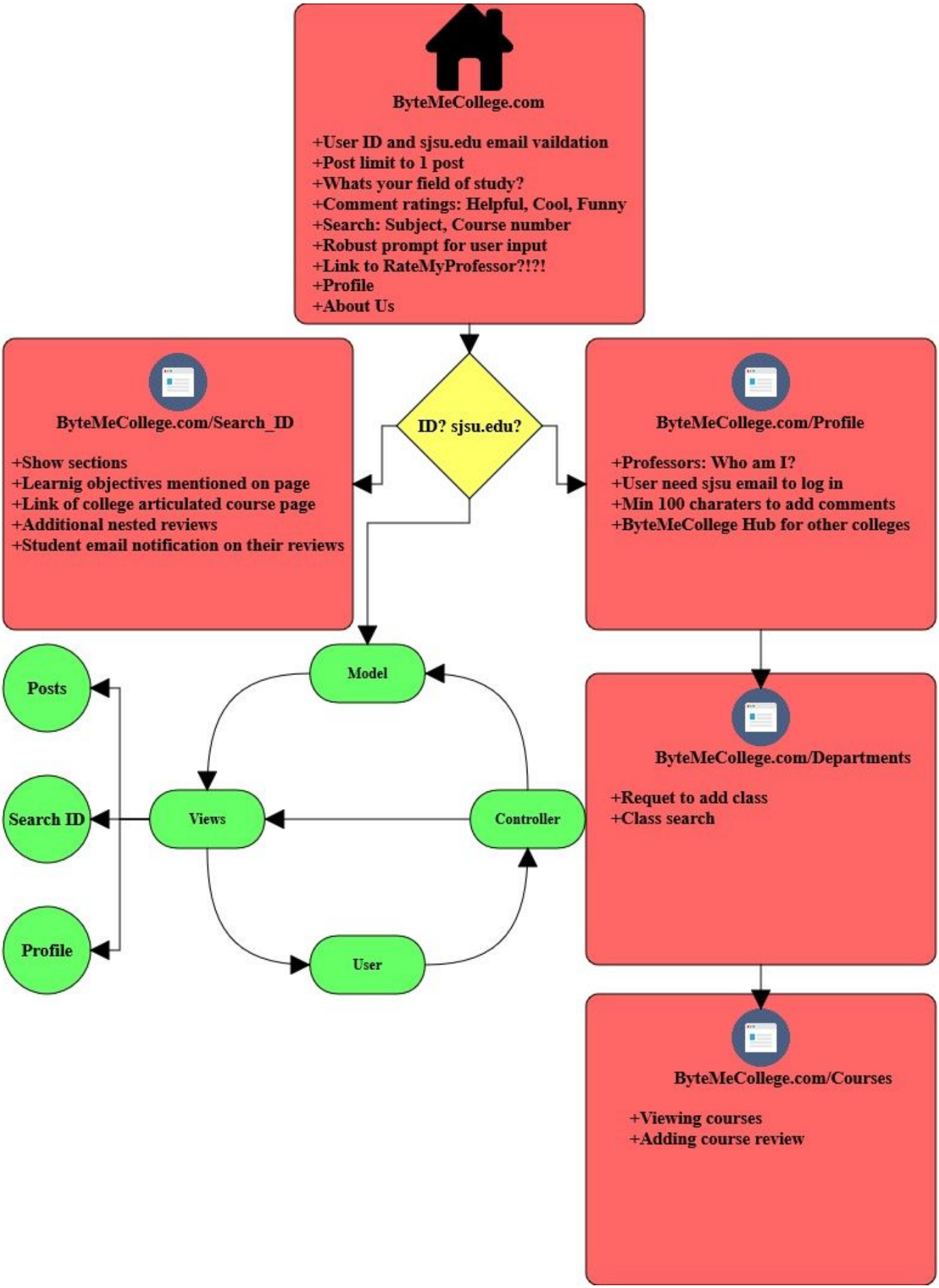
9. Nonfunctional Requirements

The system displays text in English.

- ❏ The system responds to the students immediately. May vary depends on the network.

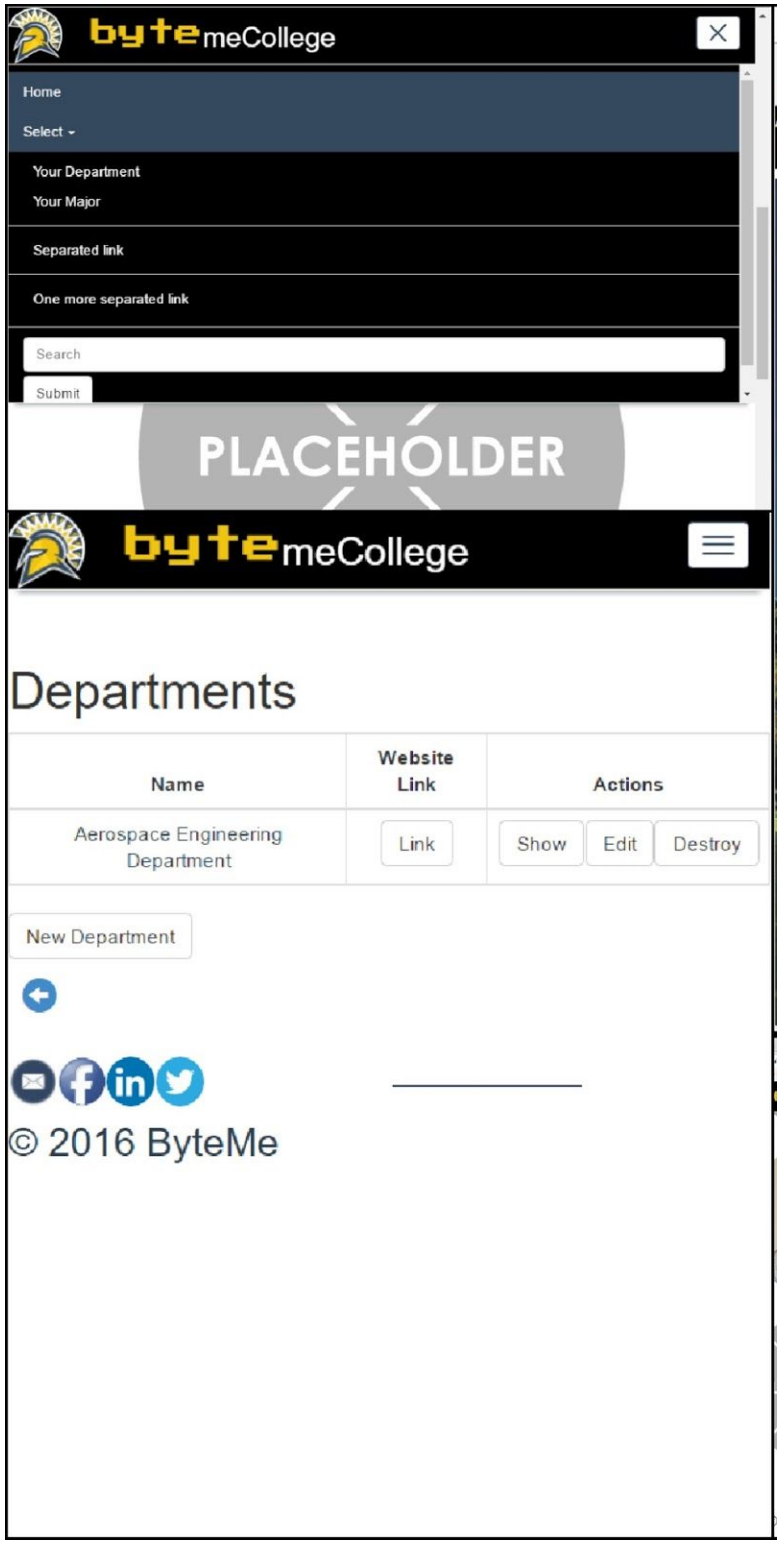
10. Glossary

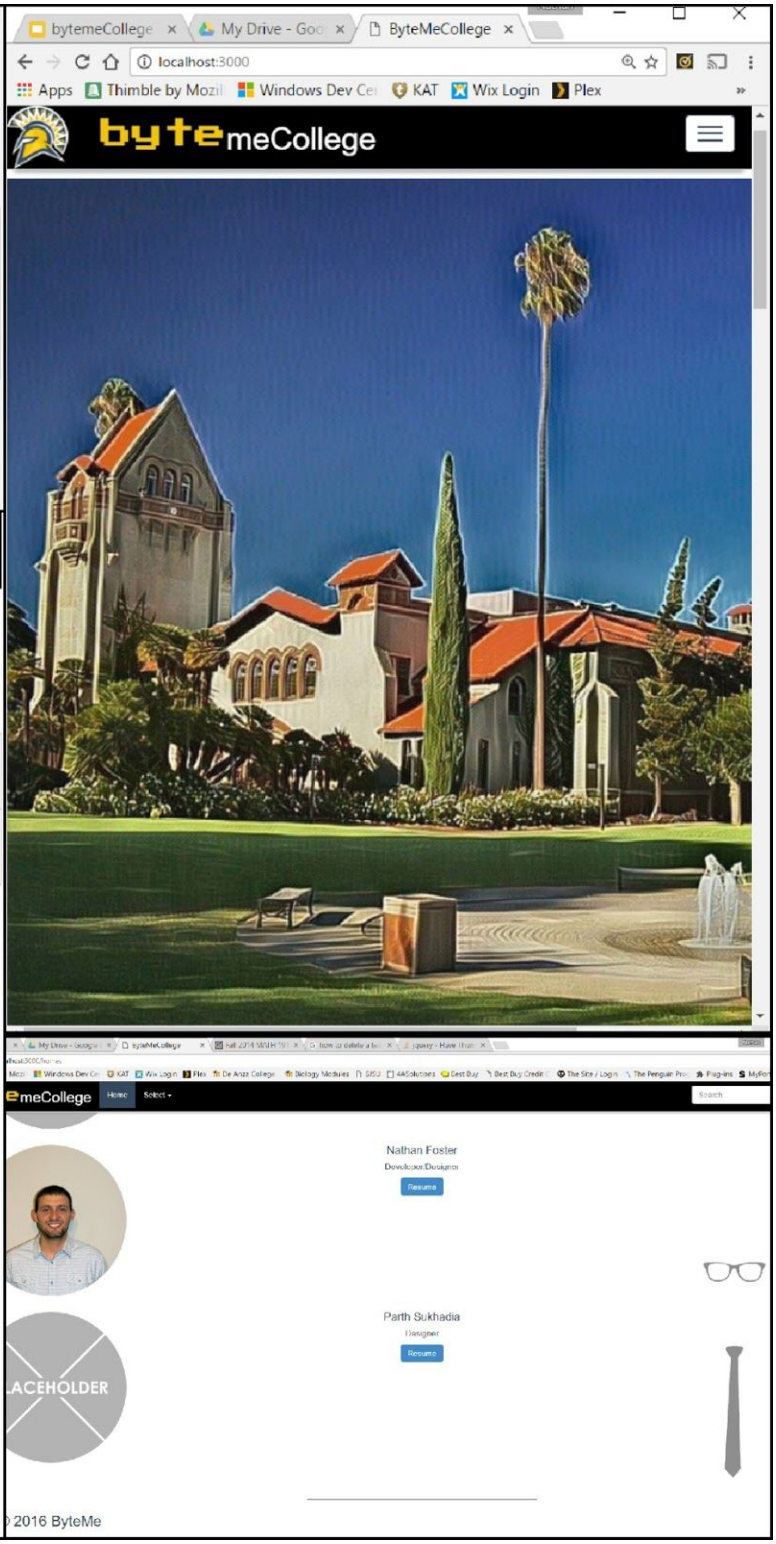
- ❏ Student = a student who already logged in to the system
- ❏ System = ByteMeCollege.com

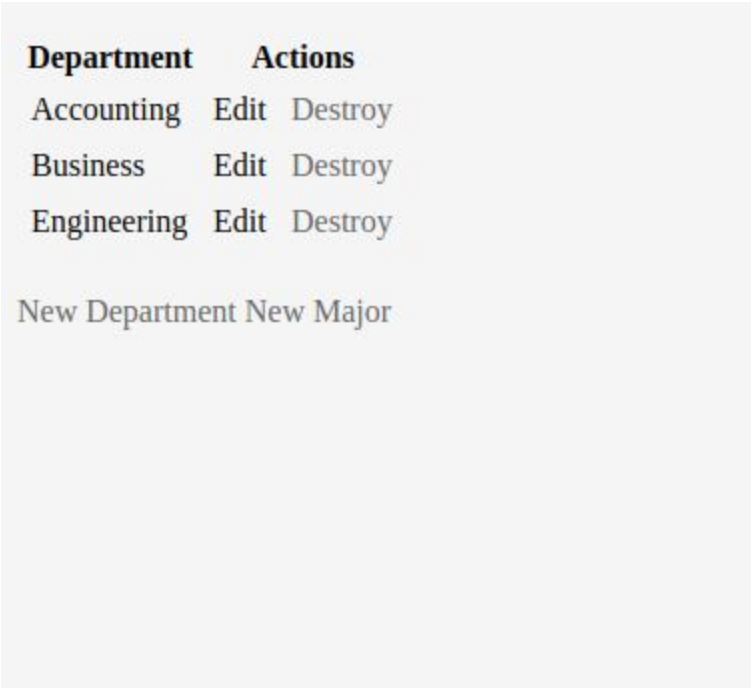

ByteMeCollege.com
+User ID and sjsu.edu email vaildation
+Post limit to 1 post
+Whats your field of study?
+Comment ratings: Helpful, Cool, Funny
+Search: Subject, Course number
+Robust prompt for user input
+Link to RateMyProfessor?!?!
+Profile
+About Us
ID? sjsu.edu?
ByteMeCollege.com/Search_ID
+Show sections
+Learnig objectives mentioned on page
+Link of college articulated course page
+Additional nested reviews
+Student email notification on their reviews
ByteMeCollege.com/Profile
+Professors: Who am I?
+User need sjsu email to log in
+Min 100 charaters to add comments
+ByteMeCollege Hub for other colleges
Model
Posts
Search ID
Profile
Views
Controller
User
ByteMeCollege.com/Departments
+Requet to add class
+Class search
ByteMeCollege.com/Courses
+Viewing courses
+Adding course review

Conceptual design

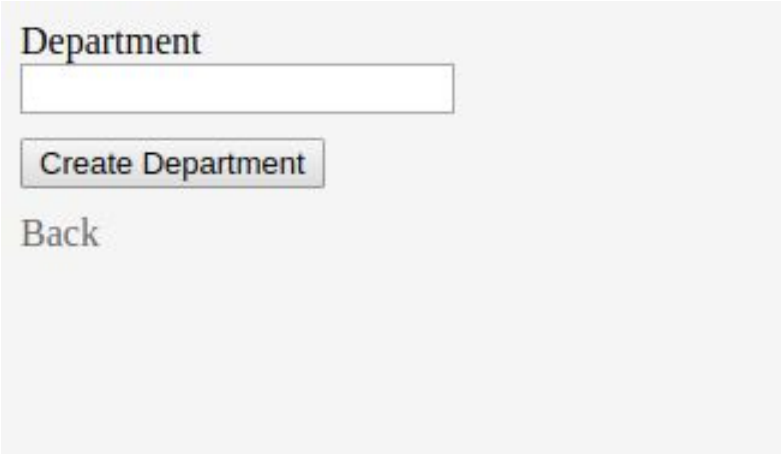
Screen Shots







- Admin can simply add and modify new departments or majors.
- Once a department or major is created or modified it is stored into the database.



Major Name

Software Engineering

Please select a department

Engineering

Update Major

Show | Back

Major Name

Please select a department

Accounting

Create Major

Back

Major	Department	Actions	
Accounting	Accounting	Edit	Destroy
ME	Engineering	Edit	Destroy
Software Engineering	Engineering	Edit	Destroy

New Department New Major

What are we?

The main objective of ByteMeCollege.com is to provide user verified reviews of courses offered by different academic institutions (currently focusing on SJSU). We hope to help students by providing constructive information to help them choose the courses that are most relevant to their academic goals and interests. Students can also do real world comparisons of similar courses offered by various institutions to help them choose the right academic programs and most fitting career path for their educational goals.

Why this Project?

College students do not have enough information about the courses they take before enrolling. They need a platform to decide relevant courses to their academic interest from offered options. While choosing a university to attend, students don't have a one-stop-shop to compare similar programs offered by different academic institutions.

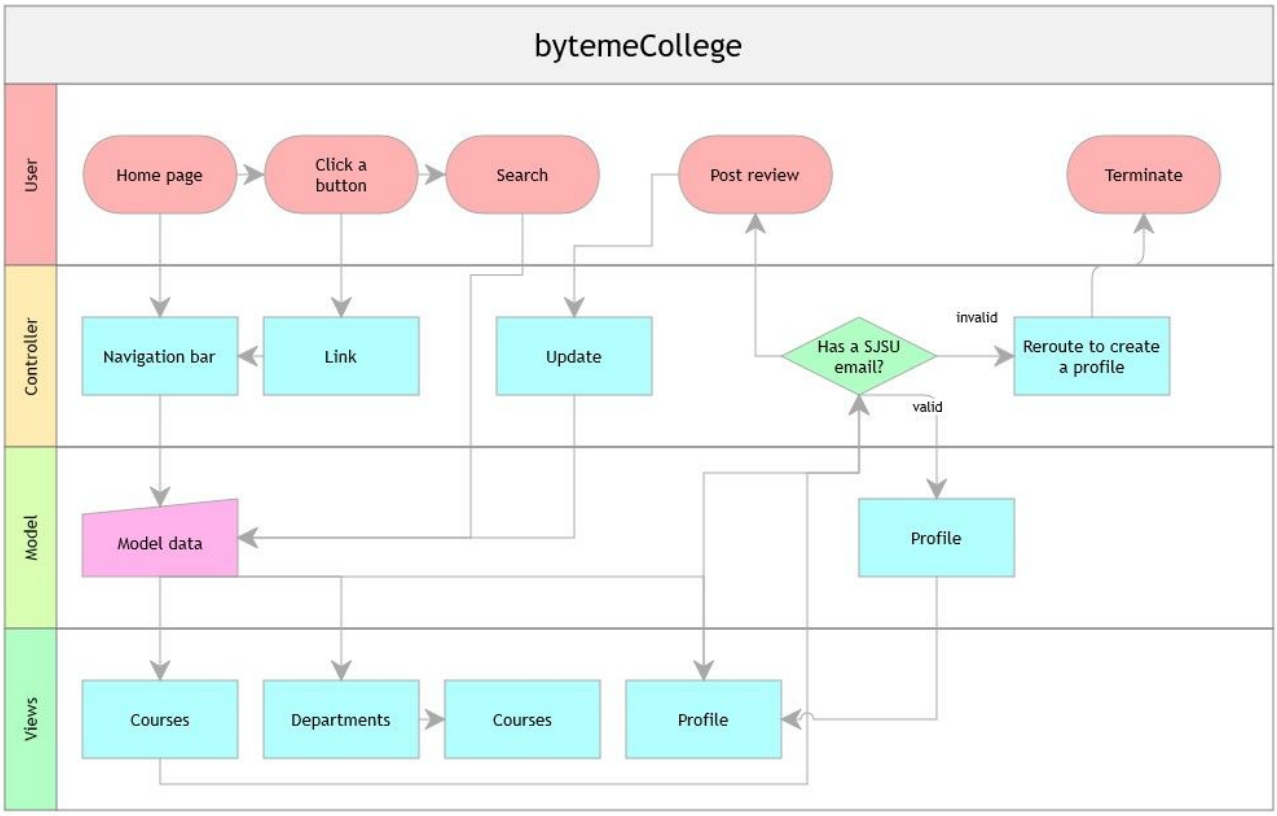
How is it different than RateMyProfessor?

- Classes are rated instead of professors
- Users are encouraged to provide constructive reviews of the course content, class structure

and what they got out of it instead of how hard the professor is.

- Only students with registered university email addresses can create accounts and leave reviews.
- User can compare courses between colleges while making decision on what college to go to.

Diagram



Test Cases
ByteMeCollege

Test Case Name:	Create an account
Purpose:	Test that the user is able to create an account through the website
Manual Operations:	● Fill out an account creation form consisting of inputs of a user name, user password, first name, last name, sjsu email, and declared major ● Click the submit button
Expected Results:	The account creation controller will successfully append the user's inputs into the database
Acceptance Criteria:	Checks the availability of all the input fields of the user and informs the user if the inputs are valid

Test Case Name:	Validate Log In
Purpose:	Test the user who is trying to log in is an exist data info in the database.
Manual Operations:	● Enter the username and the password into corresponding input text field ○ Username: sjsustudent ○ Password: student ● Press the Login button
Expected Results:	If both username and password match a data instance inside the system, the user will be directed to the user panel.
Acceptance Criteria:	● It shows a message that indicates the user login successfully
Manual Operations:	● Enter the username and the password into corresponding input text field ○ Username: donotexist ○ Password: notapsw ● Press the Login button
Expected Results:	Message ERROR: Username and password do not exist
Acceptance Criteria:	● The error message appears. ● The user is able to re-enter the username and password
Manual Operations:	● Enter the username and the password into corresponding input text field ○ Username: 1=1 ○ Password: 1=1 ● Press the Login button
Expected Results:	Be sure the system is able to prevent the “sql injection” Message ERROR: Username and password do not exist
Acceptance Criteria:	● The error message appears. ● The user is able to re-enter the username and password

Test Case Name:	Add/Update a Department
Purpose:	Display as many departments as possible for users to use as search criterias for finding courses
Manual Operations:	Enter department name into department database ● No special characters !@# allowed
Expected Results:	The user's input will be added to the database
Acceptance Criteria:	A valid university department will be added to the database
Manual Operations:	Enter invalid department name
Expected Results:	ERROR MESSAGE: invalid department
Acceptance Criteria:	Error message appears User is able to re-enter department name

Test Case Name:	Add/Update a Major
Purpose:	Display as many majors as possible for users to use as search criterias for finding courses
Manual Operations:	Enter major name into the database ● No special characters !@# allowed
Expected Results:	The user's input will be added to the database
Acceptance Criteria:	A valid major will be added to the database
Manual Operations:	Enter invalid majors into the database
Expected Results:	ERROR MESSAGE: Invalid major
Acceptance Criteria:	Error message appears User is able to re-enter major name

Test Case Name:	Add/Update a Course
Purpose:	Display all courses for a particular university
Manual Operations:	Enter name of course and corresponding course number ● Alphanumeric input accepted
Expected Results:	The user's input will be added to the database
Acceptance Criteria:	Valid course will be added to the database
Manual Operations:	Enter invalid course name/number
Expected Results:	ERROR MESSAGE: Invalid course name/number entered
Acceptance Criteria:	Error message appears User is able to re-enter course name and number

Test Case Name:	Add Course Review
Purpose:	Display reviews for various courses to be viewed by students
Manual Operations:	● Log in ● Navigate to a course and click post review button ● Enter course review with minimum 100 characters and maximum 350 words ● Press Submit button
Expected Results:	Course review will be added to site and its corresponding id will be added to database
Acceptance Criteria:	A system message shows and reload the page The review should be within the page.
Manual Operations:	● Enter course review either: ○ with less than 100 characters ○ and/or excesses 350 words ● Press Submit button
Expected Results:	Message ERROR: Review should not be less than 100 characters and/or excess 350 words.
Acceptance Criteria:	Error Message shows.

Test Case Name:	Update Course Review
Purpose:	To modify an existing comment
Manual Operations:	● Access user profile ● Navigate to user's posts ● Click the "update" button next to the perspective post.
Expected Results:	The post is modified within the database
Acceptance Criteria:	A system message indicates the user has successfully updated the review

Test Case Name:	Reply Course Review
Purpose:	Allow the user to reply to another user's post
Manual Operations:	● Log in ● Navigate to courses ● Navigate to the specific post and click the reply button
Expected Results:	The user's reply will be appended to the original review
Acceptance Criteria:	A system message shows and reload the page The review should be within the page.

Test Case Name:	Rate a Review
Purpose:	To encourage a social dynamic to the website
Manual Operations:	● Log in ● Navigate to courses ● Navigate to specific post and click on the wanted rating button ○ Can only click either the thumb up button or thumb down button once
Expected Results:	The chosen rating will be appended to the database
Acceptance Criteria:	The listed number of the corresponding action will be added one

Test Case Name:	Calculate Course Rating
Purpose:	Display the average course rating
Manual Operations:	Click on the wanted rating button
Expected Results:	Each user can only click and post the rating button of a specific course once
Acceptance Criteria:	The webpage will be displaying an average rating of the course that is included the rating just posted

byteMeCollege Home Select Login

Log in

Email

Password

Remember me

Log in

Sign up

Forgot your password?

Sign up

First name

Last name

Email

Password *(6 characters minimum)*

Password confirmation

Biography

Sign up

Log in

byteMeCollege

Home | Select | Logout

- Your Department
- Your Course
- SJSU
- About Us

Signed in successfully.

Departments

Name	Website Link
Aerospace Engineering	Link
Aerospace Studies Department	Link
African Studies Program	Link
African-American Studies Department	Link
American Studies Program	Link
Anthropology Department	Link
Art and Art History Department	Link
Asian Studies Program	Link
Athletics (Intercollegiate)	Link
Aviation	Link
Behavioral Sciences Program	Link
Biological Sciences Department	Link

Courses

Department	Course name	Overall rating	Description	Units	Action
Computer Engineering	CMPE 102		Assembly programming; assembly-C interface; CPU and memory organization; addressing modes; arithmetic, logic and branch instructions; arrays, pointers, subroutines, stack and procedure calls; software interrupts; multiplication, division and floating point arithmetic. Prerequisite: CMPE 050 or CS 046B (with grade of "C-" or better) Sophomore or upper division standing. Allowed Declared Majors: Computer Engineering, Software Engineering.	3	Reviews
Computer Engineering	CMPE 120		Introduction to computer organization and architecture, system buses, internal memory and external memory, input/output, central processing unit CPU, instruction sets, CPU structure and function, RISC, control unit. Prerequisite: CMPE 050 or CS 046B (with a grade of "C-" or better). Computer Engineering and Software Engineering Majors Only.	3	Reviews
Computer Engineering	CMPE 131		Why software engineering? What is software engineering? Software development lifecycle activities: project planning and management requirements analysis, requirement specification. Software design, software testing, verification, validation, and documentation. Software quality assurance and review techniques, software maintenance, team-based projects. Prerequisite: For CMPE Major: CMPE126 with a grade of "C-" or better. Allowed Declared Majors: Any Engineering For SE Majors: CS 046B with a grade of "C-" or better.	3	Reviews
Computer Engineering	CMPE 133		Software Architecture, Software Technical Metrics, evaluating products, processes, and resources, improving predictions, products, processes, and resources. Advanced topics such as: Formal Methods, Software Reuse, Reengineering, Client/Server Software Engineering, Computer-Aided Software Engineering, Team-Based Projects. Prerequisite: CMPE 131 (with a grade of "C-" or better). Computer Engineering or Software Engineering Majors Only. Misc/Lab: Lecture 2 hours/lab 3 hours.	3	Reviews
Computer Engineering	CMPE 148		Comparative evaluation of network architecture, layering model, standards, protocol examples for ISO and TCP/IP layers. Network applications, transport layer protocols, Internet routing, data link and physical transmissions. Applications in world wide web, file transfer, electronic mail, peer-to-peer and other areas. Prerequisite: For CMPE major: CMPE 124 and CMPE 126; for SE major: CMPE 120 and CS 146; Computer Engineering or Software Engineering Majors Only.	3	Reviews
Computer Engineering	CMPE 165		Integrated approach to managing development within small teams; including mission statement, synthesis of design concepts, tradeoff studies, risk assessment and the interactions encountered in the optimal design, development, manufacture and test of systems. Prerequisite: CMPE 133. Computer Engineering or Software Engineering Majors Only.	3	Reviews
Computer	CMPE		Introduction to enterprise software systems. Covers network operating systems, DBMS, transaction monitors, groupware, distributed objects, system tandards such as J2EE, CORBA, SQL, JDBC, and HTTP; and emerging software technologies. Prerequisite: CMPE 142 or	3	Reviews

Reviews

Course	Author	Review	Thumbs up	Thumbs down	Action	
CMPE 131	nathan.foster@sjsu.edu	Professor Mak is the best CMPE 131 instructor.	1	0	Reply	Show

Add a new Review

byteMeCollege Home Select Logout

New

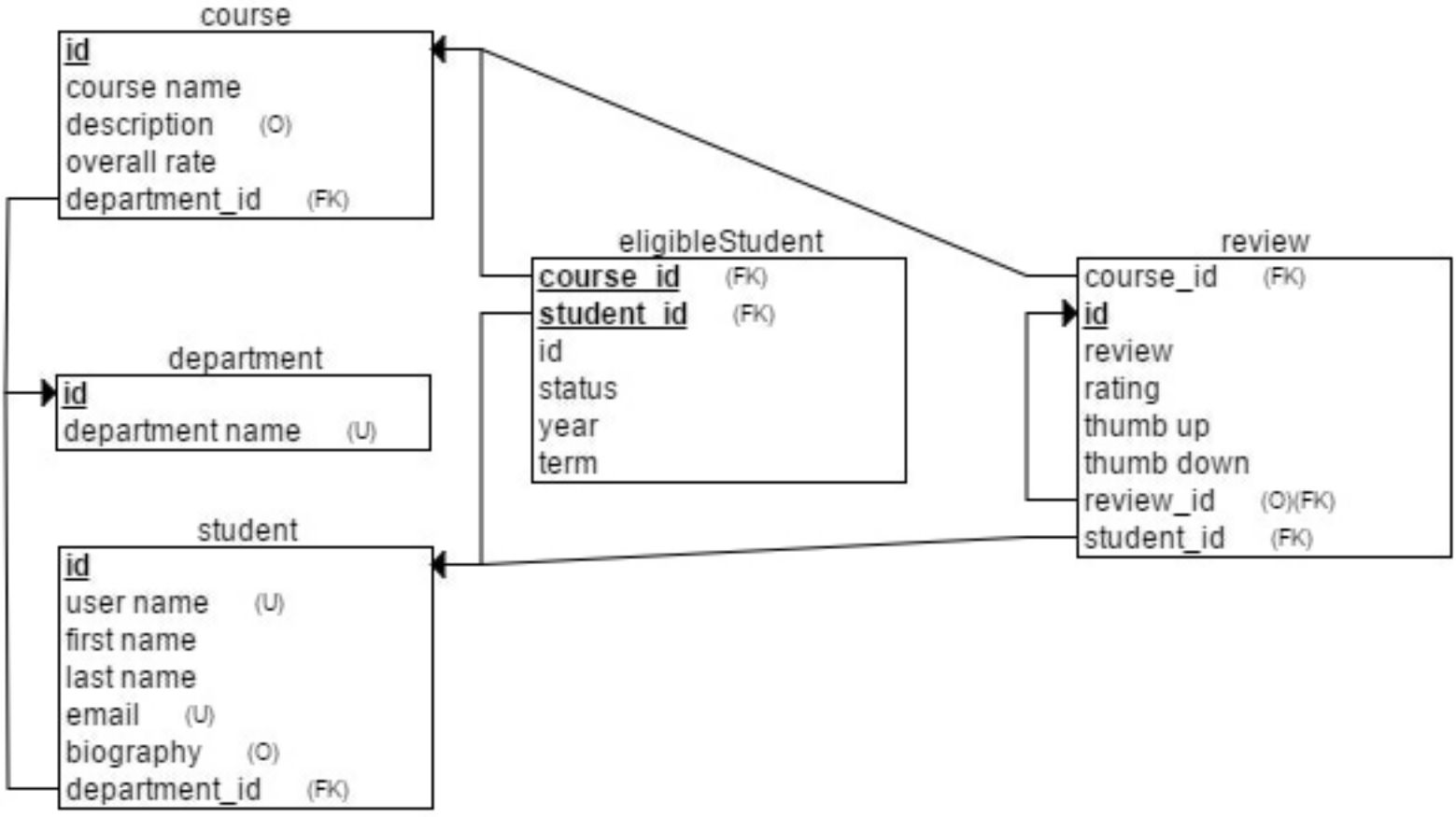
Review

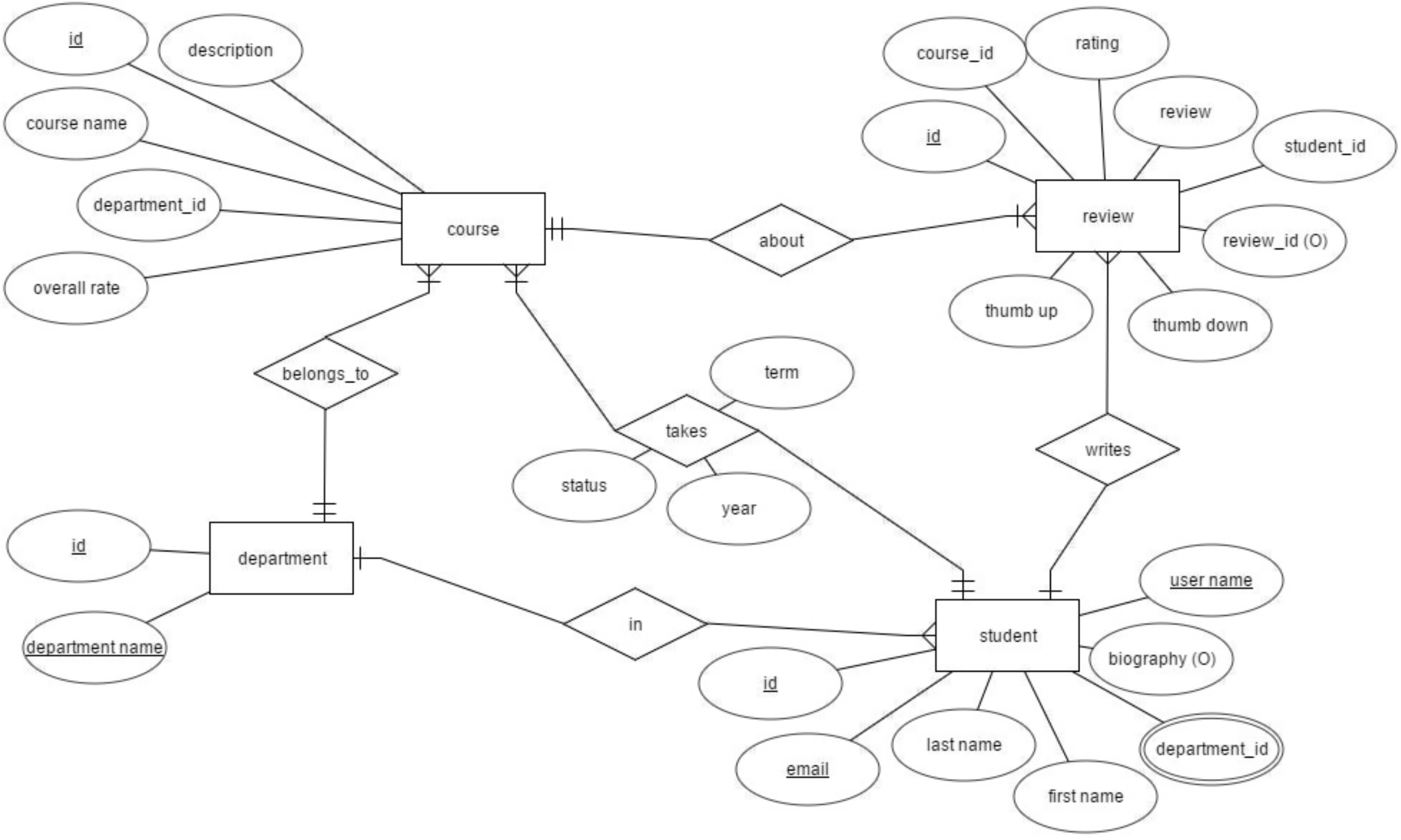
Review

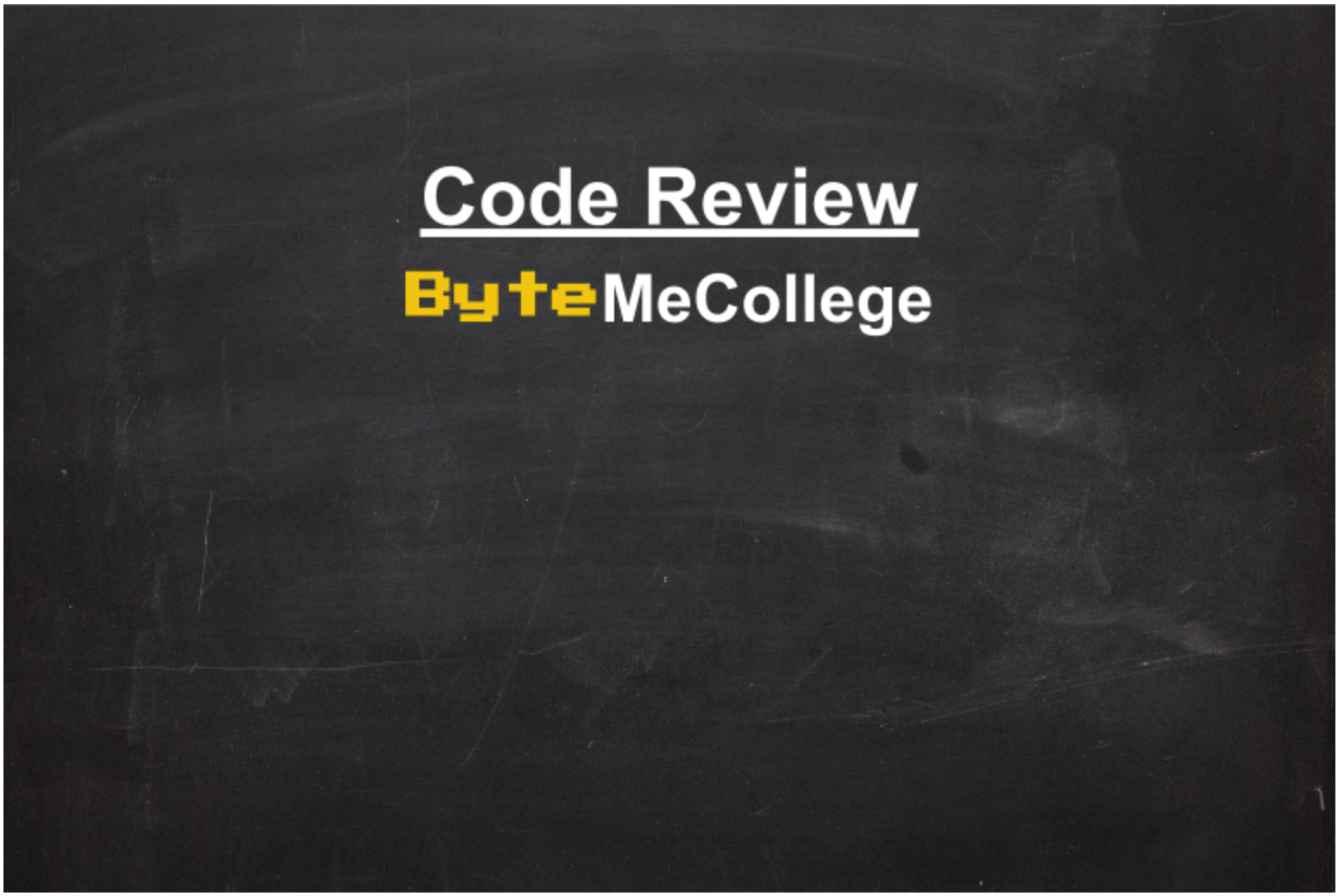
Yeah!

Create Review

Signed out successfully.


course
id
course name
description (O)
overall rate
department_id (FK)
department
id
department name (U)
student
id
user name (U)
first name
last name
email (U)
biography (O)
department_id (FK)
eligibleStudent
course_id (FK)
student_id (FK)
id
status
year
term
review
course_id (FK)
id
review
rating
thumb up
thumb down
review_id (O)(FK)
student_id (FK)


id
description
course name
department_id
overall rate
course
about
course_id
rating
id
review
student_id
review
review_id (O)
thumb up
thumb down
belongs_to
term
takes
status
year
writes
id
department
department name
in
user name
student
biography (O)
id
email
last name
first name
department_id



Views

views -> layouts -> application.html.erb

```
<!DOCTYPE html>
<html>
  <head>
    <title>ByteMeCollege</title>
    <!-- Links -->
    <%= csrf_meta_tags %>
    <%= stylesheet_link_tag
"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" %>
```

```
    <%= stylesheet_link_tag    'application', media: 'all', 'data-turbolinks-track': 'reload' %>
    <%= javascript_include_tag 'application', 'data-turbolinks-track': 'reload' %>
    <!-- End of Links -->
  </head>

  <body>
<!-- Navigation Bar -->
  <nav class="navbar navbar-default navbar-fixed-top">
<%= image_tag("sjsu_spartan_dark.png", :class => "style-image") %>
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target="#navbar" aria-expanded="false" aria-controls="navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar top-bar"></span>
        <span class="icon-bar middle-bar"></span>
        <span class="icon-bar bottom-bar"></span>
      </button>
      <a class="navbar-brand" href="#"><span id="span-byte">byte</span>meCollege</a>
    </div>

    <!-- Collect the nav links, forms, and other content for toggling -->
    <div id="navbar" class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li class="active"><%= link_to 'Home', root_path, {:class => 'button'} %><span
class="sr-only">(current)</span></li>

        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
aria-haspopup="true" aria-expanded="false">Select <span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><%= link_to 'Your Department', departments_path %></li>
        <li><%= link_to 'Your Major', majors_path %></li>
            <li role="separator" class="divider"></li>
            <li><a href="http://www.sjsu.edu/" target="_blank">SJSU</a></li>
            <li role="separator" class="divider"></li>
            <li><%= link_to 'About Us', new_home_path, {:id => 'add_row', :class => ''} %></li>
          </ul>
```

```
        </li>
      </ul>
      <form class="navbar-form navbar-right">
        <div class="form-group">
          <input type="text" class="form-control" placeholder="Search">
        </div>
        <button type="submit" class="btn btn-default">Submit</button>
      </form>
      </ul>
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>
 <!-- End of Navigation Bar -->
    <%= yield %>
    <hr></hr>
<br>
    <!-- Share buttons from simplesharebuttons.com -->
<div class="center-block">
    <!-- Email -->
    <a href="mailto:?Subject=Simple Share
Buttons&amp;Body=I%20saw%20this%20and%20thought%20of%20you!%20
https://simplesharebuttons.com">
        <img
src="https://tse3.mm.bing.net/th?id=OIP.Ma585b6caf46788df62b0033f8391eedao0&w=171&h
=169&c=7&rs=1&qlt=90&o=4&pid=1.1" alt="Email" / class="icon img-circle center-block">
    </a>

    <!-- Facebook -->
    <a href="http://www.facebook.com/sharer.php?u=https://simplesharebuttons.com"
target="_blank">
        <img
src="http://icons.iconarchive.com/icons/yootheme/social-bookmark/512/social-facebook-button
-blue-icon.png" alt="Facebook" / class="icon img-circle center-block">
    </a>

    <!-- LinkedIn -->
    <a
href="http://www.linkedin.com/shareArticle?mini=true&amp;url=https://simplesharebuttons.com
" target="_blank">
```

```
        <img src="https://simplesharebuttons.com/images/somacro/linkedin.png" alt="LinkedIn" /
class="icon img-circle center-block">
    </a>

    <!-- Twitter -->
    <a
href="https://twitter.com/share?url=https://simplesharebuttons.com&amp;text=Simple%20Shar
e%20Buttons&amp;hashtags=simplesharebuttons" target="_blank">
        <img src="https://simplesharebuttons.com/images/somacro/twitter.png" alt="Twitter" /
class="icon img-circle center-block">
    </a>

</div>
<!-- End of share buttons -->
<br>
<br>
<!-- Copyright -->
<p class="pull-left">&copy; 2016 ByteMe<p>

</body>
</html>
```

views -> students-> _form.html.erb

```
  <form id="survey-form" class="main-form">

    <div class="survey-section mat-shadow">
      <h2 class="sub-header">Profile</h2>


      <table class="profile-table">
        <tbody>

          <!-- Validate user input -->
          <tr>
            <td colspan="3">
              <input class="input" type="hidden" name="_csrf"
value="csrfTokenGoesHereFromServer">
            </td>
          </tr>
```

```
<tr>
  <td colspan="2">
  <div class="form-group">
    <%= f.label :first_name %>
    <%= f.text_field :first_name, :class => "form-control" %>
    </div>
    </tr>

  <td colspan="2">
    <div class="form-group">
    <%= f.label :last_name %>
    <%= f.text_field :last_name, :class => "form-control" %>
    </div>

<tr>
  <td colspan="2">
  <div class="form-group">
  <%= f.label :user_name %>
  <%= f.text_field :user_name, :class => "form-control" %>
  </td>
  </div>
</tr>

 <tr>
  <td colspan="2">
  <div class="form-group">
   <%= f.label :password %>
  <%= f.text_field :password, :class => "form-control" %>
  </td>
  </div>
</tr>

 <tr>
  <td colspan="2">
  <div class="form-group">
  <%= f.label :email %>
  <%= f.text_field :email, :class => "form-control" %>
  </td>
```

```
    </div>
  </tr>

  <tr>
    <td colspan="2">
    <div class="form-group">
    <%= f.label :biography %>
    <%= f.text_area :biography, :class => "form-control" %>
    </td>
    </div>
  </tr>

  <!-- Submit -->
  <tr>
    <td colspan="2">
    <div class="actions">
    <%= f.submit %>
    </div>
    </td>
  </tr>

  <!-- Back -->
   <tr>
    <td colspan="2">
<%= link_to '', students_path, {:class => 'glyphicon glyphicon-circle-arrow-left'} %>
    </td>
  </tr>

 </tbody>
</table>
</div>
</form>
```

Style Sheets

stylesheets -> _variables.scss

```
@import url('https://fonts.googleapis.com/css?family=Press+Start+2P|Roboto+Slab');

/* Color Declarations */
```

```
$white: #FFFFFF;
$white-smoke: #F5F5F5;
$yellow: #f1c40f;
$gray: #7f8c8d;
$primary: #34495e;
$secondary: #2c3e50;
$lighter: #354a5f;

/* Declaring Variables */
$background: url("../public/wallpapers/wallpaper-5.jpg");
$transparent: transparent;

/* Font Properties */
$byte: 'Press Start 2P', cursive;
$font: 'Roboto Slap', sans-serif;
$thin: 100;
$light: 300;
$normal: 400;
$medium: 500;
$bold: 700;
```

Java Scripts

javaScript.js

```
/* When the user clicks on the button,
toggle between hiding and showing the dropdown content */
    $(document).ready(function(){
     var i=1;
    $("#add_row").click(function(){
     $('#addr'+i).html("<td>"+ (i+1) +"</td><td><input name='name"+i+"' type='text'
placeholder='Name' class='form-control input-md'  /> </td><td><input  name='mail"+i+"'
type='text' placeholder='Mail'  class='form-control input-md'></td><td><input
name='mobile"+i+"' type='text' placeholder='Mobile'  class='form-control input-md'></td>");

     $('#tab_logic').append('<tr id="addr'+(i+1)+'"></tr>');
     i++;
  });
    $("#delete_row").click(function(){
         if(i>1){
```

```
                $("#addr"+(i-1)).html('');
                i--;
                }
        });
});

/* Add slideDown animation to dropdown */
$(document).ready(function(){
    $(".dropdown").hover(
        function() {
            $('.dropdown-menu', this).not('.in .dropdown-menu').stop( true, true ).slideDown("fast");
            $(this).toggleClass('open');
        },
        function() {
            $('.dropdown-menu', this).not('.in .dropdown-menu').stop( true, true ).slideUp("fast");
            $(this).toggleClass('open');
        }
    );
});
```