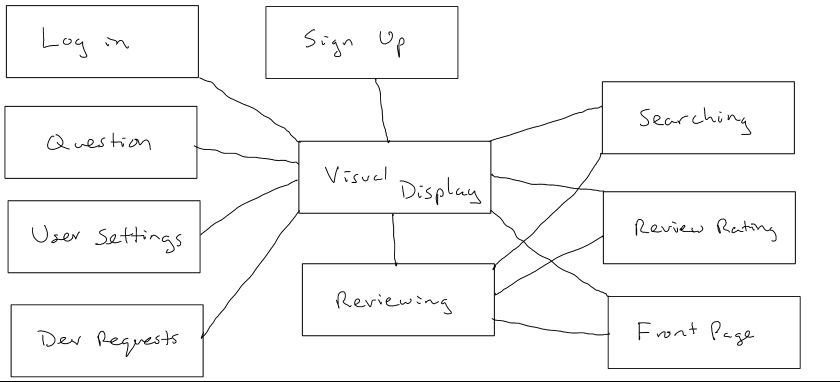
**Part A: Components Diagram**

We followed bottom up testing where we tested our individual submodules before committing them into the main branch of the repo to put it all together into a working website.



* Sign up Component - Allows users to create an account on the website
  1. input: email, password
  2. output: creates an account for the user
  3. dependent components: Visual Display Component
* Log in Component - allows the users to sign into their account on the website
  1. input: email, password
  2. output: logs the user into the website
  3. dependent components: Visual Display Component
* Front Page Ranking Component
  1. input: courses with reviews
  2. output: displays the list of highest rated courses on the homepage
  3. dependent components: Review Component, Visual Display Component
* Searching Component - Allows users to find a class to review or see reviews
  1. input: course number, department, course name
  2. output: displays a list of relevant search results sorted by rating
  3. dependent components: Review Component, Visual Display Component
* Reviewing Component - Allows users to write a review for a specific class
  1. input: information relevant to a review (review content, professor name, etc…)
  2. output: the review is published to the course’s page
  3. dependent components: Visual Display Component
* Question Component - Allows users to ask and answer questions about a class
  1. input: questions about a class or responses to already published questions
  2. output: the question/answer is published to the course’s page
  3. dependent components: Visual Display Component
* Review Rating Component - Allows users to rate reviews based off how helpful or funny they are
  1. input: the user will click on a helpful or funny button
  2. output: a counter is incremented on the page and updated in the databases
  3. dependent components: Visual Display Component, Review Component
* Dev requests/contact devs Component - Allows users to contact the developers
  1. input: email, content
  2. output: the developer email account is sent an email with the content and email given
  3. dependent components: Visual Display Component
* User Settings Component - Allows users to change their password
  1. input: current password, new password
  2. output: the password for the user is updated in the database and an alert is displayed to the user notifying them that the password has been updated
  3. dependent components: Visual Display Component
* Visual Display Component - This is the overall display of the website
  1. input: no input required
  2. output: the website content is displayed
  3. dependent components: none

**Part B: Components Incremental and Regression Test Breakdown**

Project Subsections

* Sign up System
  1. No Confirmation Email/Confirmed Email tests - Fixed by adding confirmable fields to the user database and adding :confirmable to the user model in accordance with devised
* Log in System
  1. Login 003 - Error Message is behind the navbar when using an incorrect email or giving an incorrect password
* Front Page Ranking System
* Searching System
* Reviewing System
  1. Reviews 004 - Description was un-clickable - Fixed by deleting extra jquery
  2. Reviews 004 - Description visually changes back after change
  3. Reviews 003 - Don’t have to be logged in to change description
  4. Reviews 005 - Doesn’t show timestamp for reviews
* Question System
* Review Rating System
* Dev requests/contact devs System
  1. Contact Devs 001 - Email not automatically filled out when logged in
* User Settings System (change password)
* Visual Display

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Sign Up System | | |
| Incremental Testing | | | |
| Defect # | Description | Severity | How To Correct |
| 1 | The website doesn’t send an email for the user to confirm their email with the website. | 1 | Add “confirmable” field to the user database and add :confirmable to the user model in accordance with devise. This forces devise to send an email confirmation to any new account. |
| Regression Testing | | | |
| Defect # | Description | Severity | How To Correct |
| N/A | No defects found | N/A | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Log In System | | |
| Incremental Testing | | | |
| Defect # | Description | Severity | How To Correct |
| 2 | Login 003 - Error Message is behind the navbar when using an incorrect email or giving an incorrect password. | 2 | Configured the site HTML/CSS structure to support extra room for the navbar and give more space for the error to show when needed. layouts/\_navbar.html.erb & layouts/application.html.erb |
| Regression Testing | | | |
| Defect # | Description | Severity | How To Correct |
| N/A | No defects found | N/A | N/A |

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Reviews System | | |
| Incremental Testing | | | |
| Defect # | Description | Severity | How To Correct |
| 3 | Reviews 004 - Description for a course was un-clickable while logged in. | 2 | Delete extra jquery in /courses/show.html.erb that did not allow for that field to be clickable. |
| 4 | Reviews 003 - Don’t have to be logged in to change the description of a course | 3 | Add an “if logged in” statement (courses/show.html.erb), and just simply show the description without allowing editing if not logged in |
| 5 | Reviews 005 - Doesn’t show timestamp for reviews when submitted | 1 | Add  <p><%= review.created\_at %></p>  below the code for showing reviews, this displays the time correctly. |
| Regression Testing | | | |
| Defect # | Description | Severity | How To Correct |
| 1 | Reviews 004 - Description visually changes back after a user attempts to change it. | 1 | We changed how the description is updated/sent into the database with the javascript code so that the page does not need to be refreshed to show changes. |

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Dev requests/contact devs System | | |
| Incremental Testing | | | |
| Defect # | Description | Severity | How To Correct |
| 6 | Contact Devs 001 - Email not automatically filled out when logged in | 3 | Had the placeholder saved as the email but the value stored was not actually the email. Changed so that the value stored was the email of the user using current\_user.email. This was wrapped in an if statement that would check if the user is logged in so that if the user is not logged in we do not put anything. |
| Regression Testing | | | |
| Defect # | Description | Severity | How To Correct |
| N/A | No defects found | N/A | N/A |