

Release 1 Documentation

COSC 4P02

Version 1.0

Due: Sunday 7th Mar at 23:55

R8Scholar

Team Leader (Scrum Master):

Twino Puthiakunnel / mp13ko@brocku.ca / 5564182 / mp13ko

Product Owner:

Seth Shickluna / ss16wn@brocku.ca / 6217558 / SethShickluna

Erikas Klimusinas / ek15kg@brocku.ca / 5903547 / ErikasK

Grant Nike / gn17az@brocku.ca / 6349302 / GrantNike

James Sargent / js17sy@brocku.ca / 6380356 / shorinbonsai

Logan Bell / lb16tp@brocku.ca / 6047211 / loganbe11

Luciano Ugalde / lu16xx@brocku.ca / 6102545 / lu16xx

Munashe Masango / mm16rh@brocku.ca / 6204911 / munashemasango

Table of Contents

Table of Contents	1
Introduction:	3
Phases:	4
Home Page:	4
Signup:	6
Login:	8
Logout:	9
Profile Page:	10
Instructor Page:	11
Course Page:	12
Department Page:	13
Instructors listing page:	14
Courses listing page:	15
Departments listing page:	16
Release:	17
Iteratives:	17
Release 1 goals:	17
User stories:	17
Signup	17
Email Confirmation	17
Log In	18
Edit Profile	18
Create Review	18
View Instructors	18
View Departments	19
View Courses	19
Search	19
Sprints:	19
Sprint 1 (Feb 1st - Feb 14th):	19
Sprint 2 (Feb 15th - Feb 22th):	20
Sprint 3 (Feb 22nd - Feb 27th):	20
Sprint 4 (Mar 1st - Mar 5th):	21

Teamwork policy:	22
Meetings:	22
Meeting 1 - First Group Meeting - Jan 11th	22
Meeting 2 - Discuss Requirements Elicitation - Jan 18th	22
Meeting 3 - Share resources and information - Jan 22nd	22
Meeting 4 - Ask Professor Naser questions - Jan 25th	22
Meeting 5 - Finalize Requirements - Jan 29th	23
Meeting 6 - Sprint Planning 1 - Feb 1th	23
Meeting 7 - Sprint Refinement 1 - Feb 8th	23
Meeting 8 - Sprint Review Meeting 1 - Feb 14th	23
Meeting 9 - Sprint Planning 2 - Feb 15th	23
Meeting 10 - Sprint Refinement 2 - Feb 18th	23
Meeting 11 - Sprint Planning 3 (+ Review 2) - Feb 22th	23
Meeting 12 - Sprint Review 3 (+ FullStack 3) - Feb 27th	24
Meeting 13 - Sprint Planning 4 (Release 1) - March 1st	24
Meeting 14 - Sprint Review 4 (Release 1) - March 5th	24
Meeting 15 - Release Meeting - March 7th	24
Meeting 16 - Release Submission Meeting - March 7th	24
Project Proposal Meetings:	25
Project Proposal - Jan 11th to Jan 15th	25
Project Proposal Meeting Jan 15th	25
Requirements Document Meetings:	25
Requirements Document - Jan 18th to Feb 7th	25
Requirement Document Submission - Feb 7th	26
Full Stack Meetings:	26
FullStack Meeting 1- Feb 13th	27
FullStack Meeting 2- Feb 16th	27
FullStack Meeting 3 - March 3rd	27
Misc Meetings:	27
Diagram Group Meeting (Grant and Luciano) - Jan 24th	27

Introduction:

COSC 4P02 R8Scholar Release 1 Documentation:

Expectation of the system:

- Online platform to rate scholarly aspects of Brock University
- Anyone should be able to view all ratings on the site
- Verified brock students should be able to leave ratings on the site
 - Signup / Login
 - Verify Brock Email
- Needs to display a page for each department
 - Departments can list staff and courses
 - All of these are dynamically generated, based off database entries
 - Search feature on front page

Expectation of the team:

The team will be looking to take the project on using the SCRUM methodology, the reason being that SCRUM offers an agile development and the freedom to work in concentrated teams but with a common goal. The interleaved nature of updated requirements and incrementing of software features seems just right for this type of project.

The team will attend all weekly meetings, and collaborate to develop R8Scholar.

The timing for releases/iterations of the project will be followed according.

Phases:

Home Page:

Design: Our initial design for our homepage displays the top professor and courses for brock, as well as having a search bar and various tabs at the top in order to easily get around the site.

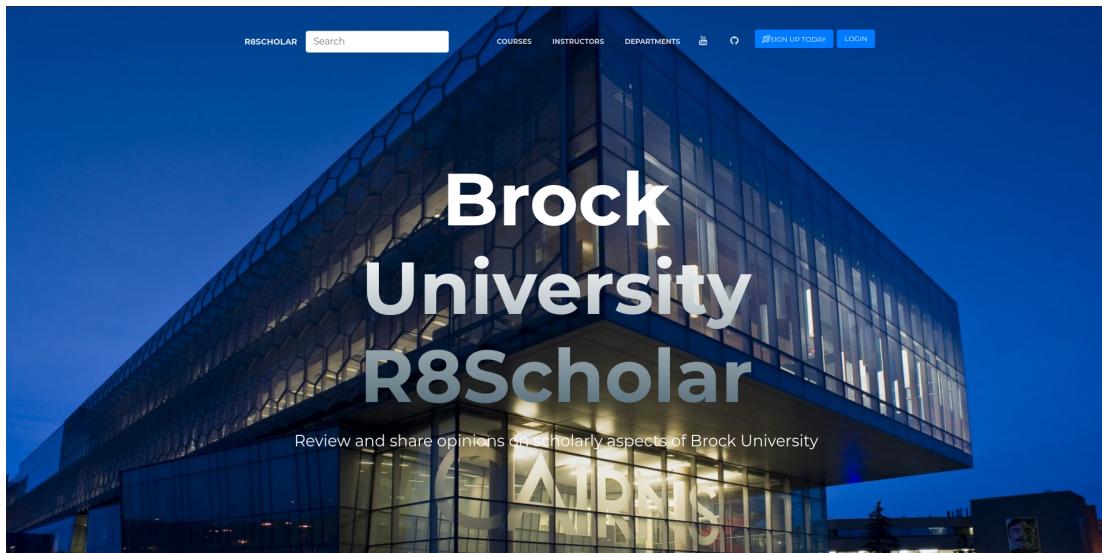
The screenshot shows the R8Scholar homepage with a red header bar containing the logo and navigation links for Login, Teachers, Courses, and About. Below the header, there are two main sections: "Top professors" and "Top courses". Each section contains five entries, each with a name, lastname, course, and rating (all listed as 5.0). At the bottom of each section is a red button labeled "View all proffesors" or "View all courses". Below these sections are two summary statistics: "Average professor rating 5.0" and "Average course rating 5.0".

Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0

Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0
Name Lastname	Computer Science	5.0

Implementation:

The home page consists of displaying the top 5 courses, instructors, and departments. The home page also has the navbar at the top which includes links to departments, instructors, courses, as well as registering and sign in options. The home page also has a search bar as well. Finally there is an article about us that will display our goals and our stance on privacy.



Top 5 Courses			Top 5 Instructors			Top 5 Departments		
Rank	Name	Rating	Rank	Name	Rating	Rank	Name	Rating
1	ABED 2F01		1	Dave Bockus		1	Aboriginal Adult Education	
2	ABED 2F02		2	Earl Foxwell		2	Aboriginal Teacher Education	
3	ABED 3F01		3	Basil Nanayakkara		3	Accounting	
4	ABED 3F02		4	Robson De Grande		4	Adult Education	
5	ABED 3F30		5	Brian Ross		5	Administration	

[VIEW ALL COURSES](#) [VIEW ALL INSTRUCTORS](#) [VIEW ALL DEPARTMENTS](#)

About Us

Our Goal

The goal of this website is to centralize the thoughts and opinions of the Brock University community. We aim to provide a comprehensive and complete experience for the users on the site, as well as providing visitors an overview of the University's academic structure. The importance of this project also is derived from the fact that web development technologies are always growing and expanding so having hands-on experience with these technologies allows us to become increasingly familiar with developer technologies that are used in the industry.

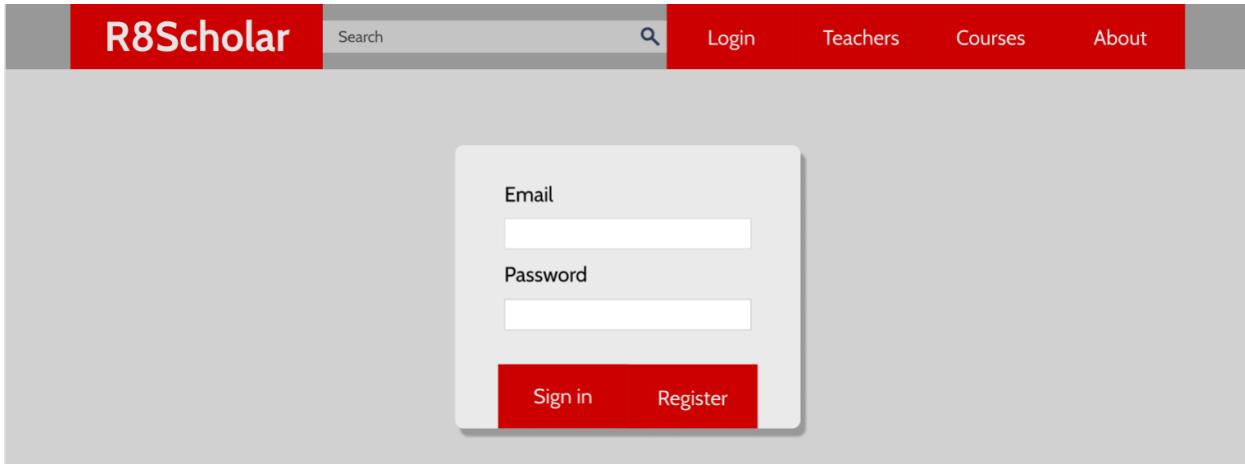
Your Privacy, Our Priority

The users of the website should be aware of the terms and conditions as well as any community guidelines in order to properly use the site as designed.

Signup:

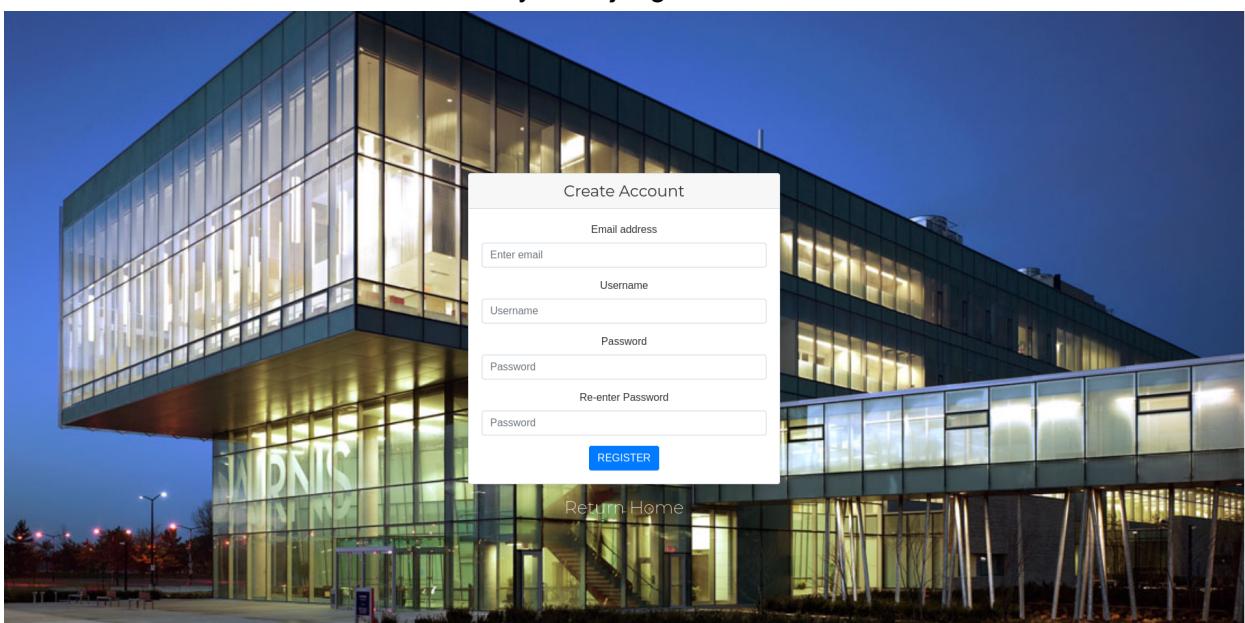
Design:

Our initial Design for the sign-up page has the user entering both an email and password in, with the options to both sign in and register a new account.



Implementation:

The signup page takes a brock email, a username, a password entered twice. The data is sent using a POST request from the React framework client side to be handled by the Django/Rest framework server side. The email, username, and password are validated on the backend by the built in validators in the Django/Rest framework, which confirm that the data matches the constraints defined in the database schema. For example the username and email must not already be registered to another user. The password is also validated by our own custom validators that ensure the password meets some minimum security requirements. The password must be at least 10 characters long, and contain at least a single uppercase letter, lowercase letter, and digit. The password is hashed using the SHA-256 algorithm and salted before it is stored in the database, this is handled by the Django/Rest framework.



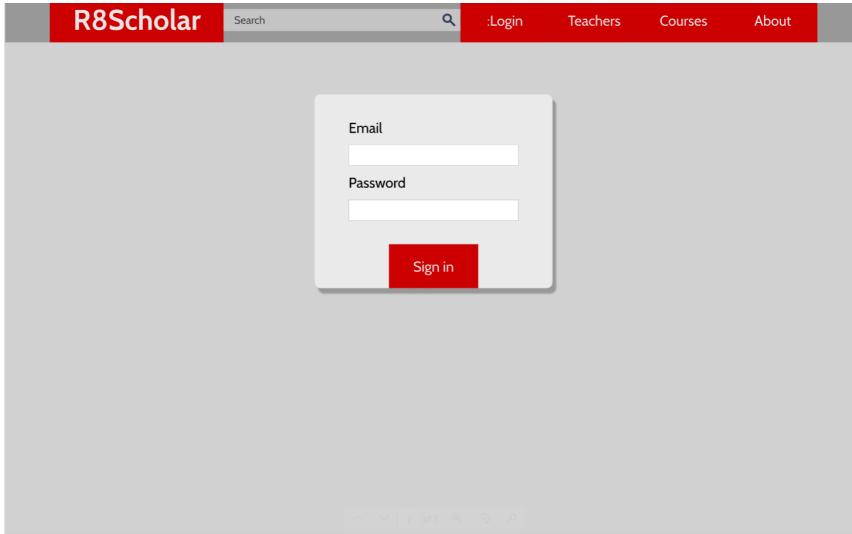
Test:

We have tested to ensure all data is validated correctly before a new user is created in the database. This was done by creating a test for each type of incorrect data; passwords under ten characters, passwords not containing a digit, uppercase letter, and lowercase letter, non-brock email addresses, and emails and usernames that are already registered to another user. We have also tested that a user is created correctly when valid data is entered. This was done using sample data created to simply meet the requirements for valid data, and then checking the database user table against the sample data.

Login:

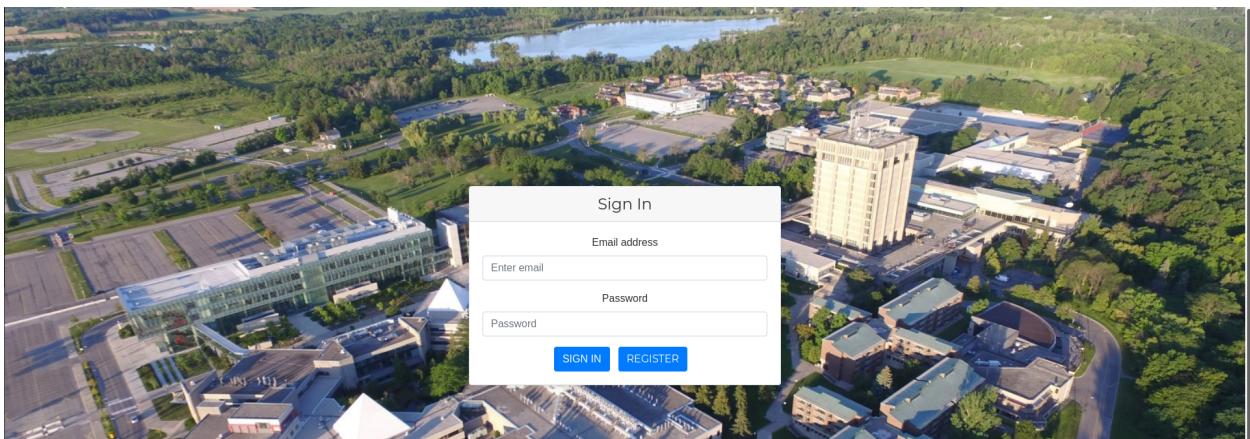
Design:

Initially we had two various designs for our Login page. The key difference between the two being one of the logins has its own individual page, while another has a forum box at the top corner of the navigation bar for those who are not currently logged in



Implementation:

The login page takes a user's email and a hash of their password, and checks them against the user's database entry. If they match a session is created with the user. Otherwise the user is informed that they have entered an invalid username/password. The authentication and session creation logic is handled by the Django/Rest framework in order to ensure the user's information is handled securely.

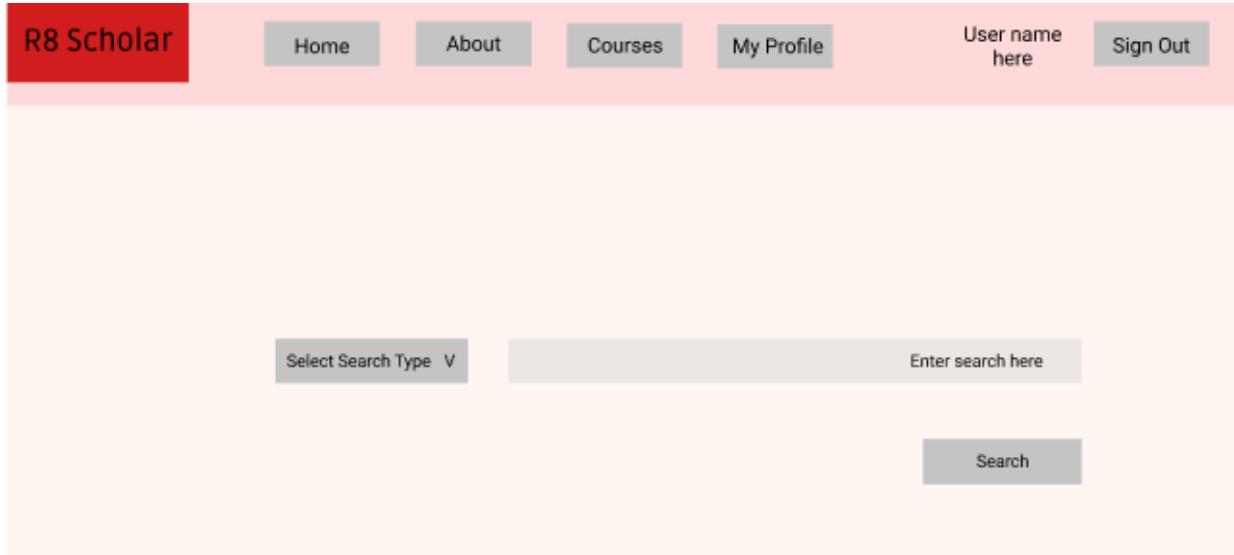


Test:

We have tested to ensure that only an email and password matching an existing user are logged in. This was done by clearing the user table in the database of any users; entering sample data into the page, and checking the database users table to ensure it was still empty. We have also tested that a session with the user is established when valid data is entered. This was done using sample user accounts originally created to test the signup page.

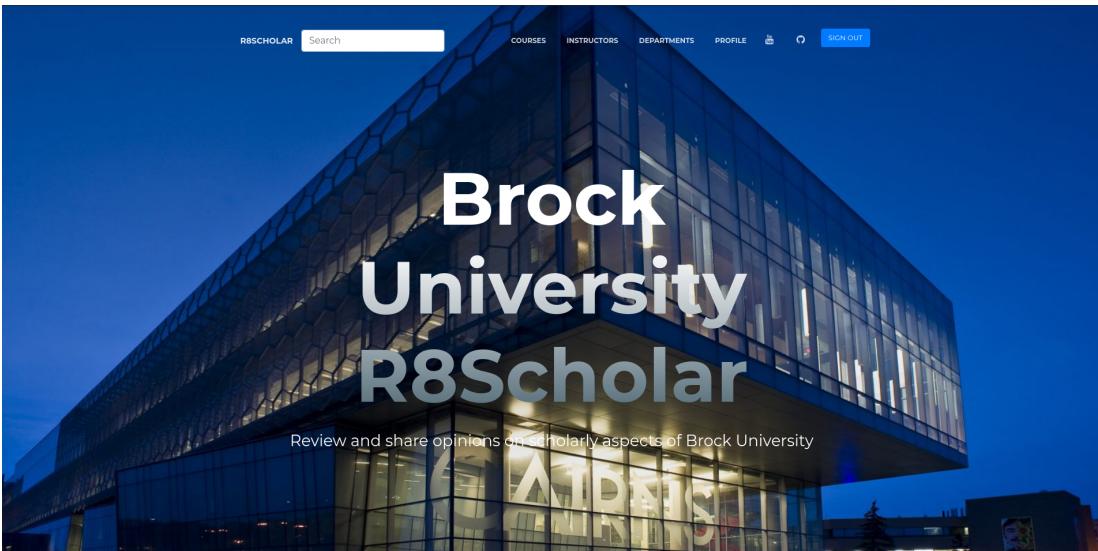
Logout:

Design: Our logout design was to be accessible from the navbar after logging in.



Implementation:

The logout functionality is accessible from the nav bar once a user has logged in. The logout request is passed to the server side where the user's session is ended and all session data is cleared. This is handled by the Django/Rest framework.



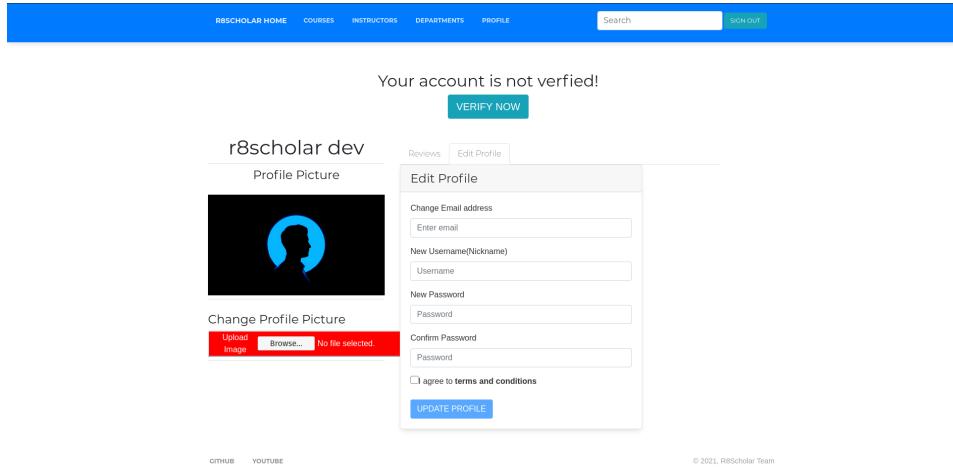
Test: was logged out, and there was no longer a session.

Profile Page:

Design: Our initial design did not have a profile page layout, although we did have plans for a profile page to be created where you could edit your user information.

Implementation:

The profile page displays the nickname and profile picture of the currently logged in user. If your account has not yet been verified this page will prompt you to verify by entering the verification code associated with your account.



Test:

We created a sample user account and navigated to the profile page. We verified the account.

Instructor Page:

Design: The instructor page would have the rating of the professor as well as various classes the professor taught and the rating of those. It would also display the professors department.

Implementation:

The instructor page displays an instructor's overall rating, based on the average of the ratings from each of the reviews associated with the instructor. It also displays the department that the instructor is associated with, with a link to the page for the department. The reviews associated with the instructor are also displayed on this page. Reviews can be created about the instructor from this page.

The screenshot shows the R8Scholar Instructor Page for Dave Bockus. At the top, there is a blue header bar with navigation links: R8SCHOLAR HOME, COURSES, INSTRUCTORS, and DEPARTMENTS. To the right of these are search, sign-up, and login buttons. Below the header, the main content area has a white background. It starts with the name "Dave Bockus" and his department, "Department of Computer Science". A section titled "Overall Rating" shows a 5-star rating. Below this are sections for "Popular Instructors" (listing Dave Bockus and Earl Foxwell) and "Popular Courses" (listing COSC 0N01 and COSC 0N02). At the bottom, there is a link to the "Department of Computer Science". The footer contains links for GITHUB and YOUTUBE, and a copyright notice: "© 2021, R8Scholar Team".

Course Page:

Design:

The course page would have the course, the rating based on various fields, as well as reviews for the course. The initial design also had a link to the professors that have taught this course previously.

R8 Scholar Home About Courses My Profile User name here Sign Out

Biology2P98
Dr.Stevens

Overall Rating
★★★★★ 3.7/5

Lectures
★★★★★ /5

Assignments
★★★★★ /5

tests/quizzes
★★★★★ /5

Difficulty
★★★★★ /5

Note: difficulty isn't included in overall rating

This Course has also been taught by:

Dr.Fee
Professor Brown
Dr.Good

Holly Parker 10/02/20
Overall Rating
★★★★★ 3.7/5
This class taught me a great deal about both animals and plants. I was able to learn a lot from the lectures but found More... Was this helpful? Yes No 33 5

Holly Parker 10/02/20
Overall Rating
★★★★★ 3.7/5
This class taught me a great deal about both animals and plants. I was able to learn a lot from the lectures but found More... Was this helpful? Yes No 33 5

Holly Parker 10/02/20
Overall Rating
★★★★★ 3.7/5
This class taught me a great deal about both animals and plants. I was able to learn a lot from the lectures but found More... Was this helpful? Yes No 33 5

More...

add review

Implementation:

The course page displays a course's overall rating, based on the average of the ratings from each of the reviews associated with the course . It also displays the department that the course is associated with, with a link to the page for the department. The reviews associated with the course are also displayed on this page. Reviews can be created about the course from this page.

The screenshot shows the RBScholar website interface. At the top, there is a blue header bar with navigation links: RBSCHOLAR HOME, COURSES, INSTRUCTORS, and DEPARTMENTS. To the right of these are a search bar, a 'SIGN UP TODAY' button, and a 'LOGIN' link. Below the header, the main content area has a white background. The title 'COSC 2P03' is displayed in large bold letters. Underneath it, the subtitle 'Advanced Data Structures' is shown. A horizontal line separates this from the 'Overall Rating' section, which displays a 5-star rating. Another horizontal line follows, leading to the 'Popular Instructors' section, which lists 'Dave Bockus' and 'Earl Foxwell'. A third horizontal line leads to the 'Popular Courses' section, listing 'COSC 0N01' and 'COSC 0N02'. A fourth horizontal line leads to the 'Department of Computer Science' section. At the bottom of the page, there are links for GITHUB and YOUTUBE on the left, and a copyright notice '© 2021, RBScholar Team' on the right.

Department Page:

Design: The department page displays an department's overall rating, based on the average of the ratings from each of the reviews associated with the department. It also displays the most popular instructors and courses associated with the department, with a link to the pages for the instructors and courses. The reviews associated with the department are also displayed on this page. Reviews can be created about the department from this page.

The screenshot shows the RBScholar website interface, similar to the previous one but for a specific instructor. The title 'Dave Bockus' is displayed in large bold letters. Below it, the subtitle 'Department of Computer Science' is shown. A horizontal line separates this from the 'Overall Rating' section, which displays a 5-star rating. Another horizontal line follows, leading to the 'Popular Instructors' section, which lists 'Dave Bockus' and 'Earl Foxwell'. A third horizontal line leads to the 'Popular Courses' section, listing 'COSC 0N01' and 'COSC 0N02'. A fourth horizontal line leads to the 'Department of Computer Science' section. At the bottom of the page, there are links for GITHUB and YOUTUBE on the left, and a copyright notice '© 2021, RBScholar Team' on the right.

Instructors listing page:

Design: The instructors listing page has the instructors name, rating, department, and how many courses the instructor teaches.

Name	Rating	Department	Courses
Name Lastname	5.0	Computer Science	4
Name Lastname	5.0	Computer Science	4
Name Lastname	5.0	Computer Science	4
Name Lastname	5.0	Computer Science	4
Name Lastname	5.0	Computer Science	4
Name Lastname	5.0	Computer Science	4
Name Lastname	5.0	Computer Science	4

Implementation:

The instructor listing page pulls a list of instructors from the database. This information contains a rating, the instructor's name, the department said instructor belongs to, as well as the rating of the department itself. This allows the user to see how the instructor compares to the department itself, allowing the user to have a better gage of what to expect if taking a course taught by the instructor.

Rank	Name	Rating	Department	Department Rating
1	Dave Bockus	★ ★ ★ ★ ★	Computer Science	★ ★ ★ ★ ★
2	Earl Foxwell	★ ★ ★ ★ ★	Computer Science	★ ★ ★ ★ ★
3	Basil Nanayakkara	★ ★ ★ ★ ★	Mathematics	★ ★ ★ ★ ★
4	Robson De Grande	★ ★ ★ ★ ★	Computer Science	★ ★ ★ ★ ★
5	Brian Ross	★ ★ ★ ★ ★	Computer Science	★ ★ ★ ★ ★
6	Sheridan Houghten	★ ★ ★ ★ ★	Computer Science	★ ★ ★ ★ ★
7	Michael Winter	★ ★ ★ ★ ★	Computer Science	★ ★ ★ ★ ★
8	Naser Ezzati-Jivan	★ ★ ★ ★ ★	Computer Science	★ ★ ★ ★ ★

Test: Check database using api.

Courses listing page:

Design: Our initial design had options to sort by departments, course name, as well as rating with the ability to look at the courses

The initial design features a header with 'R8 Scholar' and navigation links for Home, About, Courses, My Profile, User name here, and Sign Out. Below this is a grid of course filters: ALL (selected), Biology, Computer Science, ect, ect, ect, ect, ect. A link to 'Advanced Search' is present. The main content area displays a table of courses with columns for Course Name, Professor, and Score. Each row includes a 'View Course' button.

Course Name	Professor	Score	
Biology2P87	Dr. Stevens	3.7/5	View Course
Biology2P87	Dr. Fee	4.1/5	View Course
Biologyy2P90	Dr. Stevens	3.6/5	View Course

Implementation:

The Course listing takes courses from the database, which currently has over 3800 brock courses, which are displayed alongside their course code, rating and department. Each course code also has a respective name in the database. The courses are by default listed alphabetically and display their rating as well as the department of each respectively.

The implementation shows a header with R8SCHOLAR HOME, COURSES, INSTRUCTORS, DEPARTMENTS, a search bar, and links for SIGN UP TODAY and LOGIN. The main content is titled 'Courses' with a 'Filter Options' button. Below is a table listing 12 courses, each with a rank, name, rating (5 stars), and department (Aboriginal Adult Education).

Rank	Name	Rating	Department
1	ABED 2F01	★★★★★	Aboriginal Adult Education
2	ABED 2F02	★★★★★	Aboriginal Adult Education
3	ABED 3F01	★★★★★	Aboriginal Adult Education
4	ABED 3F02	★★★★★	Aboriginal Adult Education
5	ABED 3F30	★★★★★	Aboriginal Adult Education
6	ABED 3F40	★★★★★	Aboriginal Adult Education
7	ABED 4F14	★★★★★	Aboriginal Adult Education
8	ABED 4F15	★★★★★	Aboriginal Adult Education
9	ABED 4F16	★★★★★	Aboriginal Adult Education
10	ABED 4F17	★★★★★	Aboriginal Adult Education
11	ABED 4F18	★★★★★	Aboriginal Adult Education
12	ABED 4F84	★★★★★	Aboriginal Adult Education

Test: Check database using api page.

Departments listing page:

Design: Our plan for Department listing was to keep it similar to course page listing in order for the user to have familiarity going between the two.

Implementation:

The departments listing page has all of the departments listed alphabetically, with the information being pulled from the database. Displayed alongside the department name is the rating of the department based on department reviews, as well as the top course and top instructor such that someone looking at a department can see exactly the perks of what said department can offer.

Name	Rating	Top Course	Top Instructor
Aboriginal Adult Education	★★★★★	ABED 2F01	no data
Aboriginal Teacher Education	★★★★★	ABTE 8P01	no data
Accounting	★★★★★	ACCC 0N01	no data
Adult Education	★★★★★	ADED 1P31	no data
Administration	★★★★★	ADMI 0N11	no data
Applied Disability Studies	★★★★★	ADST 3P82	no data
Academic English as a Subsequent Language	★★★★★	AESL 1P92	no data
Applied Computing	★★★★★	APCO 1P00	no data
Arabic	★★★★★	ARAB 1F00	no data
Astronomy	★★★★★	ASTR 1P01	no data
Biochemistry	★★★★★	BCHM 0N01	no data
Biology	★★★★★	BIOL 1F25	no data

Test: Check database using api page

Release:

Iteratives:

Release 1 goals:

- Create new account
 - Email authentication @brocku.ca
- Sign in to student account
 - Login/Logoff
- Edit page for the user
 - Change password
 - Change nickname
- Settings page for admin
- Viewing professor/course/department page
- Add Professor/Course/other to site
- Modify Professor/Courses/Other
- View listing of courses/professors/departments
- Get top professors
- Get top courses (per department)
- Create a review for professors, courses, departments, other
 - Rating

User stories:

Signup

Story: As a user, I would like to be able to sign up for a R8Scholar account.

Acceptance criteria:

1. Check to determine if the user has a valid Brock email account
 - a. If there the user provided a valid email address then check to see if the password is valid
 - i. If the password is valid allow user to create account
 - ii. If the password is invalid inform user that password is invalid
 - b. If the email is invalid the user will be informed to provide a Brock email address
2. User will have to agree to terms and conditions
3. The user will have to input a confirmation code which will be sent to their email address
4. User will be brought to login page after entering confirmation code

Email Confirmation

Story: As a user, I would like to have a confirmation email sent to my email address when I'm creating my account.

Acceptance criteria:

1. Check to determine if the user has a valid Brock email account
 - a. If there the user provided a valid email address then a confirmation email will be sent to that address
 - b. If the email is invalid the user will be informed to provide a Brock email address
2. The email will contain a hyperlink which allows the user to jump instantly to the R8Scholar site

Log In

Story: As a user, I would like to be able to log in to the site using my email address and password.

Acceptance criteria:

1. Check to determine if the email address is valid
 - a. If the email address is valid check to see if the password is valid
 - i. If the password is valid allow user to log in
 - ii. If the password is invalid inform user that password is incorrect
 - b. If the email address is invalid inform user that email is invalid
2. After logging in, the user will be brought to the main homepage

Edit Profile

Story: As a user, I would like to be able to go to my profile and edit my nickname and password

Acceptance criteria:

1. Check to see if the user is logged in
 - a. If not, prompt log in (see log in story)
2. Enter current password, and enter new password
 - a. If current password is correct, the new password is accepted
3. User will remain at their profile page

Create Review

Story: As a user, I would like to create a review using an input form for a specific course, instructor or department.

Acceptance criteria:

1. Only enable user review form if they are a valid user with a verified email address
2. Only allow review to be submitted if all the necessary parts of the form are filled

View Instructors

Story: As a user, I would like to be able to view a list of instructors by clicking on the “instructors” tab on the top menu bar.

Acceptance criteria: As a user, when I select the instructor tab, it will take me to the instructor page, which will then have a list of instructors that can be further filtered.

View Departments

Story: As a user, I would like to be able to view a list of departments that the university offers.

Acceptance criteria: As a user, when I select the department tab, it will take me to the department page, which will then have a list of departments that can be further filtered. It will also display ratings, the top course, and the top instructor.

View Courses

Story: As a user, I would like to be able to view a list of available courses by clicking on the "Courses" tab on the top menu bar.

Acceptance criteria: As a user, when I select the courses tab, it will take me to the courses page, which will then have a list of courses and departments that can be further filtered.

Search

Story: As a user, I would like to be able to use a search feature to specifically find the information that I want in an efficient manner.

Acceptance criteria: As a user, I will be able to input what I am searching for into the search bar, and the system will return all matching objects.

Sprints:

Sprint 1 (Feb 1st - Feb 14th):

Goals for the first week of sprints: My idea is to have the back end be a week ahead on the technical, so the week after the front end crew can just hook up the UI to what has been done.

- Front end: Create pages for each component of the site.
 - Sidebar as per figma design
 - Footer
 - Home page
 - About page
 - Courses page
 - Instructors page
 - Departments
 - Account page (just the skeleton, there wont be much)
 - Settings page
- Back End:
 - Create the database

- Setup views.py accordingly
- Create serializers for backend-frontend interaction
- Get API running for authenticating users, creating accounts, creating, reviews, and sending requested reviews to front end

Changes to requirements documentation:

- No changes compared to the documentation

Sprint 2 (Feb 15th - Feb 22th):

- Features for next sprint
 - Create account
 - Account authentication
 - Login/logout
 - Account settings
 - Change overall theme to light blue
 - Bonus: send email for authentication
- Decided to leave hosting to next week
- Decided sprints should be 1 week long
- Front end:
 - Fetching from the API
 - Trouble with user creation
 - How to query just one user, etc (views?)
- Back end:
 - Authentication started
 - Looked into built-in forums in Jango, some need to be extended
 - User creation forum adjusted, checked for @brocku.ca
 - How to send emails, being worked on

Changes to requirements documentation:

- Addition of account settings
- Changing theme color to blue

Sprint 3 (Feb 22nd - Feb 27th):

- Server side rendering would be a good option if it works
- Features for next sprint:
 - Get reviews working
 - Create oncreate method
 - Check how emails can be sent
- Backend:
 - Views for functionality
 - Models -> Update/User Verification
- Frontend:
 - API Requests
 - Style Fixes
 - Make pages work nicely together

- Fullstack:
 - API requests (Get/Create views)
 - Populate DB with reviews entities on main

Changes to requirements documentation:

- May change ISA in model (may need to be updated in requirements)
- Made nicknames unique
- Added new functional requirement for password
- (Should change department not have a specific review object)

Sprint 4 (Mar 1st - Mar 5th):

- Review creates the instructor first, departments, courses, reviews
- FullStack
 - API calls, get-create views (Luciao, Grant, James)
 - Populate database, text files, run scripts (Twino, Logan)
- Frontend
 - Styles (Seth, Munashe)
 - Functional components (all buttons should work)
 - Login/logout (Grant)
 - Changing password/nickname (Grant)
 - Course display page (frontend team)
- Backend
 - API calls (Grant, Luciano, James)
 - Database (James)

Features:

- Login/Logoff
- Email authentication @brocku.ca
- Confirmation emails to be sent out (to be done)
 - Password reset function
- Fill in database
- Views (api calls, and send data correctly)
 - Login/Logout
 - When a review is made, update the rating of review subject
- Edit page for the user
 - Change password
 - Change nickname
- Settings page for admin
- Make review

Changes to requirements documentation:

- Add ability to get top processors, courses, and departments

Teamwork policy:

Meetings:

Meeting 1 - First Group Meeting - Jan 11th

- Everyone attended and contributed, Meeting 1 summary includes more details
- James and Seth feel comfortable with web development.
- James, Seth, Twino(Mathew) and Erikas have some app development experience.
- James and Grant have some database experience.
- Everyone is familiar with GitHub.
- Seth offered to make a Youtube video to promote the project

Meeting 2 - Discuss Requirements Elicitation - Jan 18th

- Twino(Mathew) Puthiakunnel emailed and invited Professor Naser
- Everyone in the group attended the meeting
- Twino(Mathew) Puthiakunnel created shared Requirements Elicitation document
- James Sargent suggested using Django for databases
- Seth Shickluna-Pierce decided to work on database as well
- Logan Bell and Erikas Klimusinas decided to work on the UI using Figma
- Grant Nike and Luciano Ugalde decided to work on diagrams and may use draw.io
- Munashe Mansango agreed to work on requirements
- Twino(Mathew) Puthiakunnel decided to work on requirements as well

Meeting 3 - Share resources and information - Jan 22nd

- Figma was shared by Logan Bell
- PDF of UI development was shared by Erikas Klimusinas
- James Sargent shared tutorial on django web development
- Seth Shickluna-Pierce demonstrated Amazon AWS deployment at <http://3.96.54.189/>
- Seth Shickluna-Pierce registered the domain r8scholar.ca
- PDF of UI development was shared by Logan Bell
- Erikas Klimusinas started a discussion if there is a need for a registration page
- James Sargent suggested that github should be set up with the basic project soon
- Seth Shickluna-Pierce suggested using React for the front end of the site
- Luciano Ugalde and Grant Nike made a diagrams channel in Teams

Meeting 4 - Ask Professor Naser questions - Jan 25th

- Twino(Mathew) Puthiakunnel asked questions to Professor Naser on behalf of the team
- Logan Bell and Erikas Klimusinas Suggested Changes to be made in order for the back end to be compatible with UI
- Twino(Mathew) Puthiakunnel offered to be in charge of full-stack coordination

- Seth Shickluna-Pierce offered to be in charge of front end coordination
- James Sargent offered to be in charge of back end coordination
- Group decided on updated requirements for system entities
- James Sargent began the setup of the back end project
- Seth Shickluna-Pierce began the setup of the front end

Meeting 5 - Finalize Requirements - Jan 29th

- Seth Shickluna-Pierce, James Sargent, and Twino(Mathew) Puthiakunnel discussed how the teams should be split up and what topics they should cover during sprints
- Twino(Mathew) Puthiakunnel made channels for Sprint Planning and Sprint Review

Meeting 6 - Sprint Planning 1 - Feb 1th

- Twino(Mathew) Puthiakunnel conducted the meeting
- M.Masango updated the GitHub Readme

Meeting 7 - Sprint Refinement 1 - Feb 8th

- (Feb 11) M.Masango added About page content, added footer component
- (Feb 12) M.Masango added basic forum posting functionality

Meeting 8 - Sprint Review Meeting 1 - Feb 14th

- Scheduled FullStack Meeting 2
- Seth began discussion on hosting situation

Meeting 9 - Sprint Planning 2 - Feb 15th

- Group completed backlog
- Seth Shickluna-Pierce shared current hosting situation
- James Sargent shared the current status of the REST api
 - Initial models setup

Meeting 10 - Sprint Refinement 2 - Feb 18th

- Seth did fetching from API
- Grant has been working with user creation forms, and is working on email verification
- (Feb 18) M.Masango began work on the forum home page.
- (Feb 19) M.Masango added a basic Profile Page
- (Feb 20) M.Masango added custom forms and profile picture editing to Profile page

Meeting 11 - Sprint Planning 3 (+ Review 2) - Feb 22th

- Seth made signup with accounts work
 - Seth demonstrated account signup
- James fixing the database

- Cleaning up models.py, admin.py and serializers.py
- Munashe added to account page, and settings page

Meeting 12 - Sprint Review 3 (+ FullStack 3) - Feb 27th

- Seth demonstrated new project structure, used webpack and babel
- Grant worked on new models, rewrote error messages
- (Feb 27) M.Masango added an admin settings page
- (Feb 28) M.Masango added the custom forms for additional functionality on the settings page.

Meeting 13 - Sprint Planning 4 (Release 1) - March 1st

Split tasks:

- API calls, get-create views (Luciao, Grant, James)
- Populate database, text files, run scripts (Twino, Logan)
- Styles (Seth, Munashe)
- Functional components (all buttons should work)
 - Login/logout (Grant)
 - Changing password/nickname (Grant)
 - Course display page (frontend team)
- API calls (Grant, Luciano, James)
- Database (James)
- Grant and James went over setting up local backend with Luciano

Meeting 14 - Sprint Review 4 (Release 1) - March 5th

- Seth demonstrated the new looks for the R8Scholar site

Meeting 15 - Release Meeting - March 7th

- Grant suggested changes
- James removed migrations from github

Meeting 16 - Release Submission Meeting - March 7th

- Reviewed documentation

Project Proposal Meetings:

Project Proposal - Jan 11th to Jan 15th

- Created google docs during team Meeting 1
- Everyone added their information on Jan 11th
- Twino(Mathew) Puthiakunnel: Scheduled project proposal meeting with agenda Jan 11th
- Luciano Ugalde: Added more details to timeline Jan 12th
- Seth Shickluna: Started working on proposal summary/statement Jan 13th
- Munashe Masango: Worked on detailing the software engineering process for the project Jan 14th
- Twino(Mathew) Puthiakunnel: Made proper cover page, added more to agenda for meeting Jan 14th
- Seth Shickluna: Edited proposal summary/statement Jan 15th

Project Proposal Meeting Jan 15th

- Logan Bell, Twino(Mathew) Puthiakunnel, James Sargent, Seth Shickluna-Pierce, Munashe Mansango, Grant Nike, Luciano Ugalde attended meeting
- Logan Bell, Twino(Mathew) Puthiakunnel, Seth Shickluna-Pierce, Luciano Ugalde, James Sargent, and Grant Nike worked together on proposal during meeting
- Seth Shickluna-Pierce agreed to submit the project proposal on behalf of the group

Requirements Document Meetings:

Requirements Document - Jan 18th to Feb 7th

- Seth Shickluna-Pierce made a separate channel to discuss database requirements Jan 18th
- Twino(Mathew) Puthiakunnel wrote initial details needed for cover page Jan 18th
- Seth Shickluna-Pierce and Grant Nike expanded on Use Cases Jan 18th
- James Sargent added details for System Architecture Jan 18th
- Luciano Ugalde and Seth Shickluna-Pierce added details to Risk Factors Jan 18th
- Seth Shickluna-Pierce shared rough document for contents of the database Jan 19th
- Munashe Masango added list of Functional and Non-functional requirements with their tables Jan 20th
- Logan Bell worked on UI Jan 20th
- Grant Nike added to Functional and Non-functional requirements Jan 21st
- Munashe Masango added more tables for Use Cases Jan 21st
- Logan Bell worked on UI Jan 21st
- Luciano Ugalde and Grant Nike added to list of Diagrams required Jan 22nd
- Munashe Masango corrected a Use Case table Jan 22nd

- Seth Shickluna-Pierce added in Database Design information and Entity tables Jan 22nd
- Twino(Mathew) Puthiakunnel modified Cover Page and added in automated Table of Contents as well as the Table of Contents Template for planning Jan 22nd
- Twino(Mathew) Puthiakunnel added Non-Functional Requirements Jan 22nd
- Seth Shickluna-Pierce added in entity diagram Jan 23rd
- Munashe Masango added to Product Requirements Jan 24th
- Logan Bell worked on UI Jan 24th
- Twino(Mathew) Puthiakunnel added to Product Requirements Jan 24th
- Luciano and Grant created initial use-case and class diagrams on Jan 24th
- Grant updated the class diagrams for the architectural design based on the conclusions made in Meeting 4 Jan 26th
- Munashe Masango reformatted document Jan 29th
- James modified database design entities Jan 29th
- Seth Shickluna-Pierce updated user interface requirements Jan 30th
- Seth added Stakeholder requirements Jan 31st
- Grant Nike corrected Entities to Instructors Feb 1st
- Munashe Mansango added Deployment Diagrams Feb 1st
- Twino(Mathew) Puthiakunnel updated Table of Contents Feb 1st
- Munashe Masango, James Sargent, Seth Shickluna-Pierce, Grant Nike, and Twino(Mathew) Puthiakunnel worked together for document formatting, and adding additional information. 33 edits in total throughout the document. Feb 1st
- Grant Nike updated the numbering for all the contents meant to be on the table of contents. Feb 2nd
- Grant Nike added in use case diagrams Feb 2nd
- Munashe Masango edited and added details to requirements tables Feb 3rd
- Munashe Masango added more sequence diagrams Feb 4th
- Logan Bell researched usability studies for home pages Feb 4th
- Luciano Ugalde added more use case diagrams Feb 4th
- Seth Shickluna-Pierce added a use case diagram Feb 4th
- Seth Shickluna-Pierce edited titles Feb 5th
- Grant Nike updated table of contents and corrected numbering Feb 5th
- Munashe Masango updated requirements tables and corrected numbering Feb 5th
- Twino(Mathew) Puthiakunnel updated table of contents and checked through document for errors Feb 5th
- Seth Shickluna-Pierce made adjustments to Class Diagram Feb 5th
- Grant Nike removed unnecessary information from Requirements Document Feb 5th
- Seth Shickluna-Pierce looked over Requirements Document for errors. Feb 7th
- Twino(Mathew) Puthiakunnel finalized formatting for submission Feb 7th
- Seth Shickluna-Pierce submitted Requirements Document Feb 7th

Requirement Document Submission - Feb 7th

- Worked as a team on requirements document finalization

Full Stack Meetings:

FullStack Meeting 1- Feb 13th

- Logan Bell and Luciano Ugade attended meeting
- Discussed state of FullStack

FullStack Meeting 2- Feb 16th

- Logan Bell, Luciano Ugade and
- Twino(Mathew) Puthiakunnel attended meeting
- Logan Bell, Luciano Ugade and Twino(Mathew) Puthiakunnel Worked on Log book
- Logan, Luciano and Twino decided on documentation updates every 3 weeks in-line with releases

FullStack Meeting 3 - March 3rd

- Twino, Logan, Luciano met with Seth to discuss API requests and DB population
- Twino and Logan decided to use beautiful soup for the DB
- (March 3) Merged front-end branches

Misc Meetings:

Diagram Group Meeting (Grant and Luciano) - Jan 24th

- Compared notes on Use-Case Diagrams and decided on grouping various requirements together
- Collaborated on rough draft of UML class diagrams through reviewing the database layout and current functional requirements
- Compiled list of questions to be discussed with group during meeting 4