Software Requirements Specification for Rate My Scholar

Authors: Aaron Muysson, Daniel Figueroa

Customer: Scholars

Instructor: Naser Ezzati-Jivan

1. Functional Requirements

- The system shall allow users to search for a scholar by school.
- The system shall allow users to search for a scholar by name as well.
- The system shall display all the scholars at a given school that are pertinent.
- The system shall display the rating of quality for a given scholar.
- The system shall display the comments placed by other scholars about a given scholar.
- The system shall allow the user to rate the quality of a given scholar.
- Furthermore, the system shall allow the user to provide verbal feedback in the form of a comment.
- The system must interact with a database to yield the results of the scholars pertinent to a school. Encompassing all individual staff, professors, students etc. of that school.
- The system shall allow the user to register for an account through prompting the user for a username and password.
- The system shall allow the user to reset their password by navigating to a forgot my password page on the system.
- The system shall allow the user to login to their account and give them the ability to leave reviews and peruse them..
- The system shall allow the user to log out of their account and still peruse reviews.

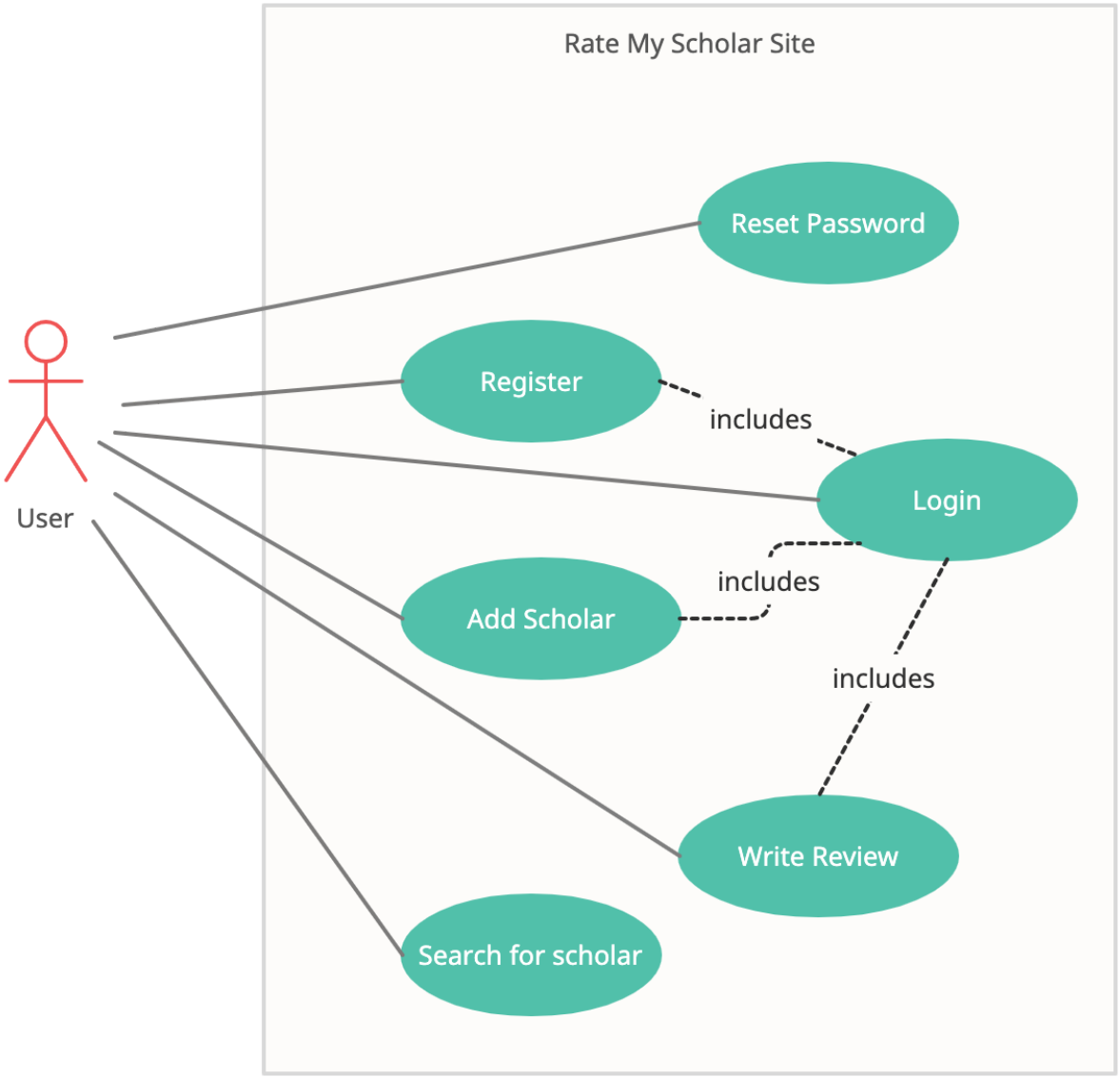
2. Nonfunctional Requirements

- The feedback from the users will require a degree of moderation to prevent effective harassment of an individual.

- The speed of the search results are to be relatively quick as the database will hold a lot of the pertinent data.
- The system will have to be operable at all times. Specifically 24 hours a day and all days of the week must have minimal, if any, downtime.
- The passwords of users are to be hashed and salted to maintain the utmost security of the user.
- The personal information demanded from the user is to be kept at a minimum as to ensure greater anonymity for all users.
- The system should be mobile friendly as well in addition to the refined look of the computer counterpart.
- The system should be scalable to the potential large number of users that will be present especially with the existence of many scholars to peruse from as well as the amount of individuals that could use the site.
- The system shall be able to support at least 20 simultaneous users and more if possible.
- The system shall only ever be down for more than a minute for cases such as maintenance on the system.
- The system shall be easy to understand and properly documented in addition to utilizing certain data structures to make maintainability of the software a simpler process.
- The manageability of the system must be quite simple especially due to the scale of the system and must be accounted for in the development of the system.
- The data stored in the database is to be kept consistent and accurate whenever possible as to avoid use of the database on non-existent scholars.
- Furthermore, the data must be able to be reliably and consistently fetched with minimal error.
- The user experience of the system must be one defined by minimal error or load time for all features in addition to providing a convenient means to accomplish all the aforementioned functional requirements.
- Ensure the system is functional on as many operating systems and web browsers as it is a site-based system.

3. Use Case and Sequence Diagrams

3.1 Use Case



3.1 Use Case Explanations

System	Rate My Scholar

Use Case Name	Reset Password
Actors	User
Description	The user forgets their password and needs to reset their password.
Stimulus	The user clicks the reset password button on the log in page.
Response	The user is redirected to a page which prompts them to enter their email and they will be sent an email for the reset.
Comments	N/A

System	**Rate My Scholar**
Use Case Name	Register
Actors	User
Description	The user wants to register as a user on the site so they can interact with more features.
Stimulus	The user clicks the button on the navigation bar to register or the sign up button on the login.
Response	The user is redirected to a page which prompts them to enter their email address, username and password for their account to make their account. Once it is entered, the user is successfully signed up as a user
Comments	N/A

System	**Rate My Scholar**
Use Case Name	Login
Actors	User
Description	The user wants to log in to their registered account
Stimulus	The user clicks the log in button on the navigation bar.
Response	The user is redirected to a page where they will be prompted to enter their credentials and provided they enter the proper information, they will be logged in successfully.
Comments	N/A

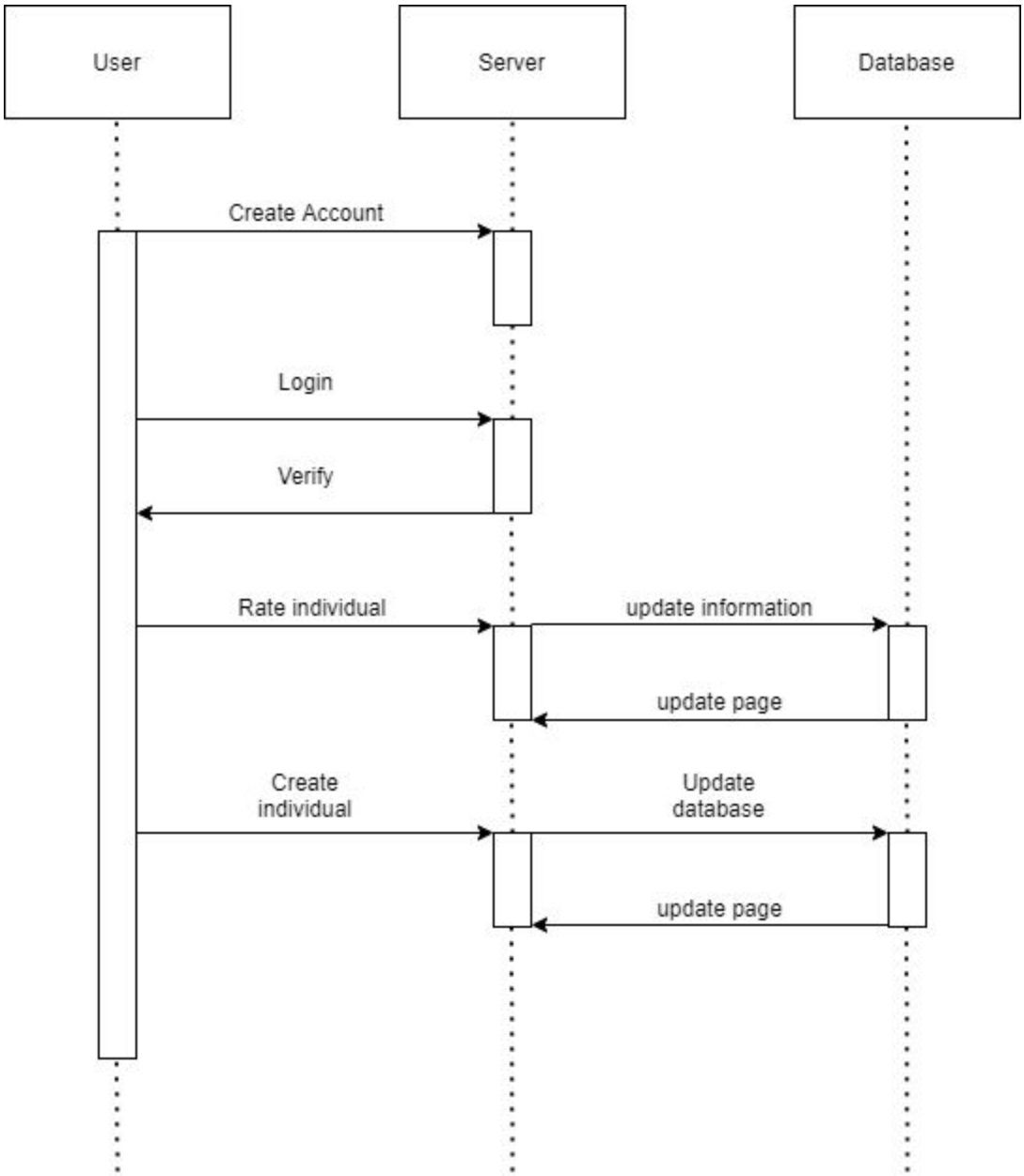
System	**Rate My Scholar**
Use Case Name	Add Scholar
Actors	User
Description	A scholar does not exist in the database and the user wishes to add that scholar to the site.
Stimulus	The user searches for the scholar and finds no result, clicks the link which prompts the user to create a new scholar for that search.
Response	The user is redirected to a page which prompts them to enter the information of this user and add it to the database
Comments	N/A

System	**Rate My Scholar**
Use Case Name	Write Review
Actors	User
Description	The user wants to write a review as a means of feedback for the scholar and attributing to their level of quality.
Stimulus	The user clicks the pertinent result in the search results and furthermore clicks to leave a review once they access the page of the specific scholar.
Response	The user is redirected to a page which prompts them to enter the star rating for the scholar as well as leave optional comments about the scholar.
Comments	This is only allowed if the user is signed in.

System	**Rate My Scholar**
Use Case Name	Search for Scholar
Actors	User
Description	The user wants to find the entry for a specific scholar.
Stimulus	The user enters a name of a school or professor in the search bar and presses the enter button.
Response	The user is redirected to a page housing the search results of that given query with a myriad of scholars to select from if the query yields that much. Otherwise

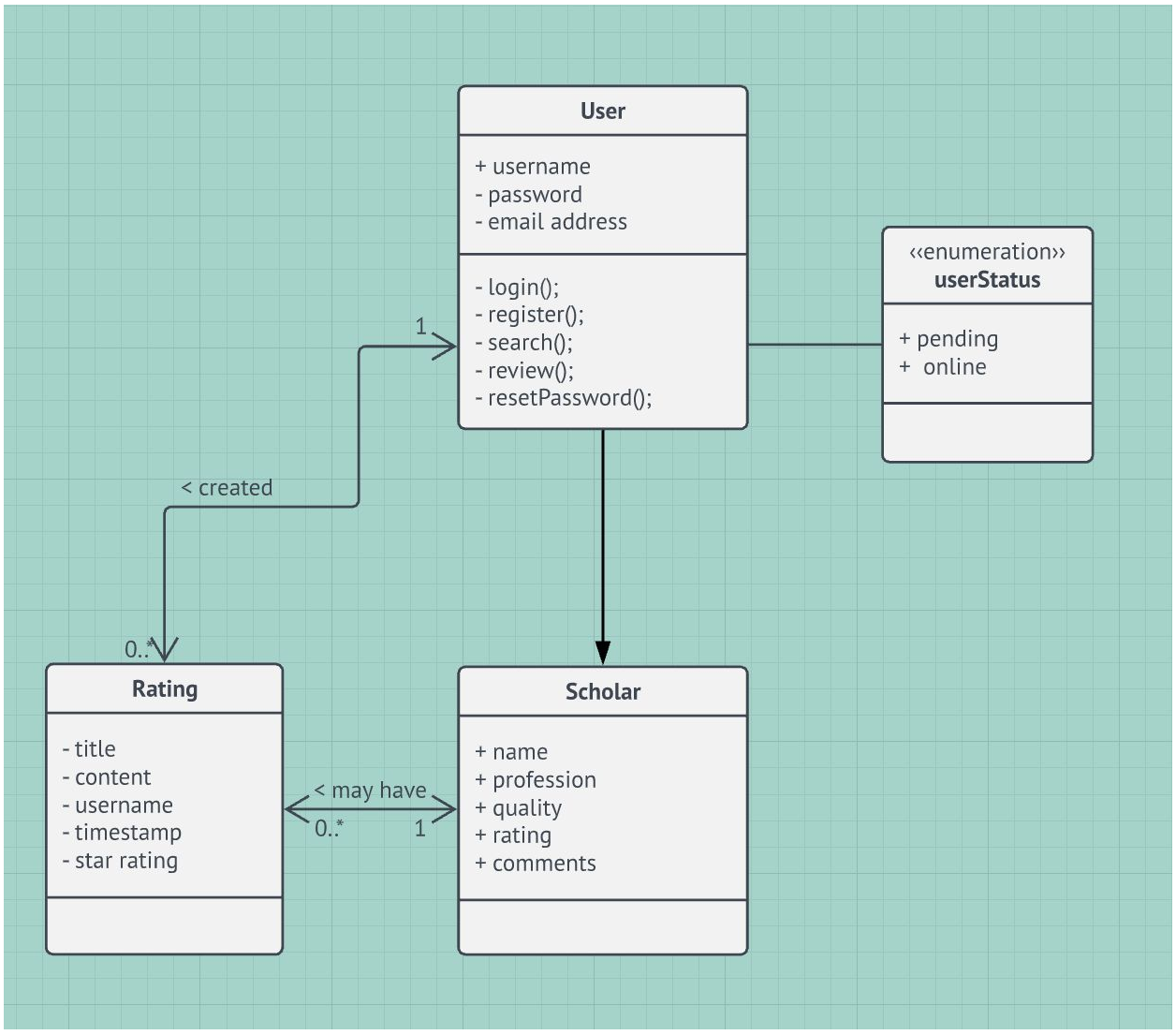
	prompts the user to add a scholar should there be no entry for that name and school.
Comments	N/A

3.2 Sequence Diagram



4. System Architecture Diagrams and Explanation

4.1 Class Diagram



4.2 Class Diagram Explanation

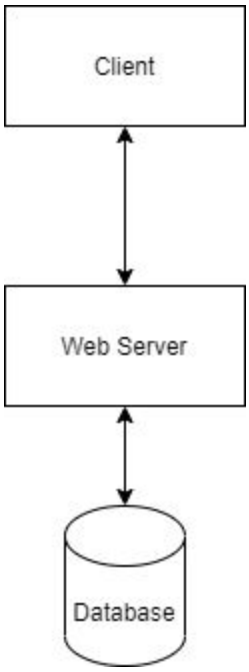
The class diagram demonstrates a relationship between the user and the scholar with two major side components in the form of classes. The user is concerned with being logged in or not and otherwise a valid user in regard to its functions as some have privileges only granted to registered and signed in users. Namely the ability review and by extension add a scholar to the directory but that is not the usual case for the system in the overall architecture.

The user has a public username but private password and email address evidently. Additionally, it is the user that has a lot of command in respect to the system having all the functions pertinent such as login, search, review etc. Namely login, reset password, and register are concerned with the validation as a legitimate user in userStatus and one must be a valid user to create a rating for a scholar.

The rating consists of a title, username, timestamp, star rating and of course the content making up the body of the rating. These are applied to the entry of a scholar in which the scholar themselves are represented by their own class.

This class has a name, profession, quality, rating and comments and is a more 1-sided entity in which they are only subject to receiving feedback or updates on information but not similar to the user class in terms of agency.

4.3 System Architecture Diagram

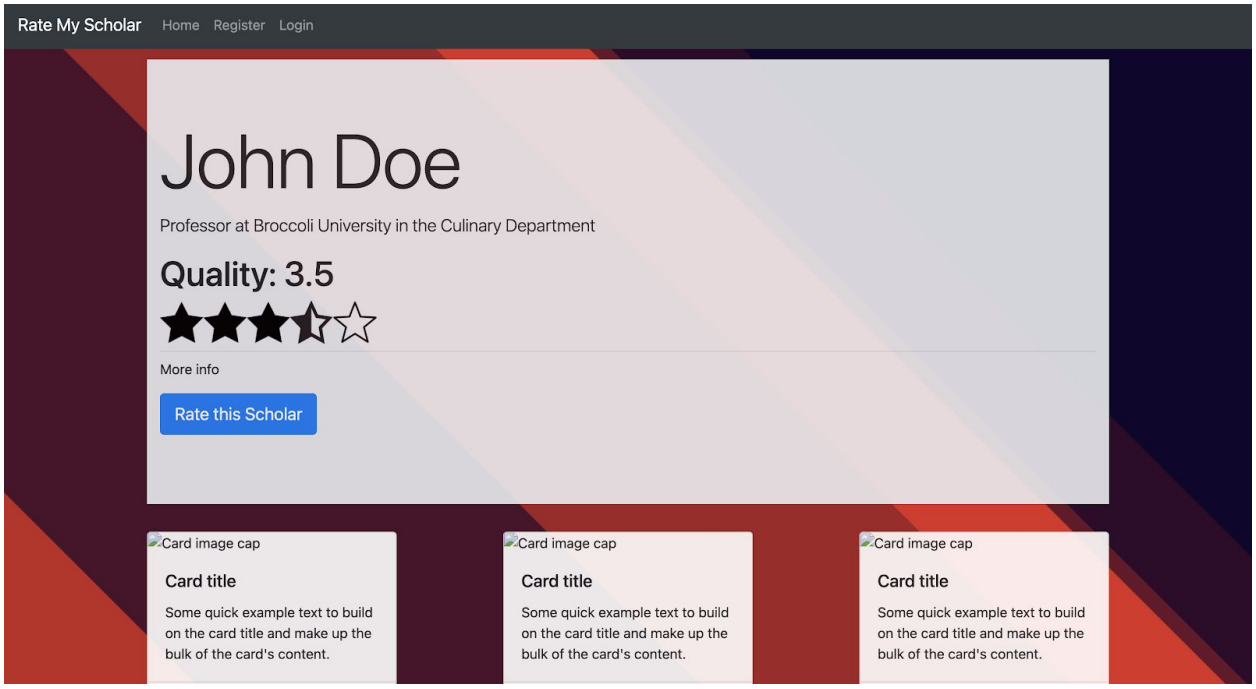


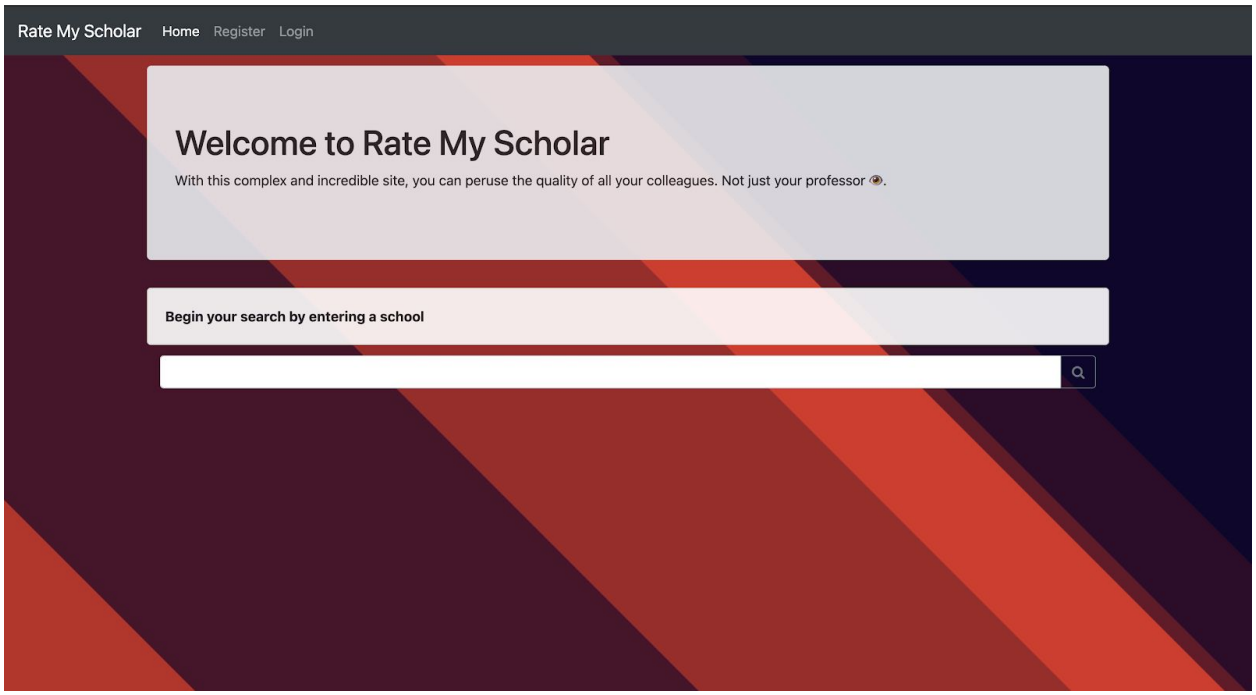
4.4 System Architecture Diagram Explanation

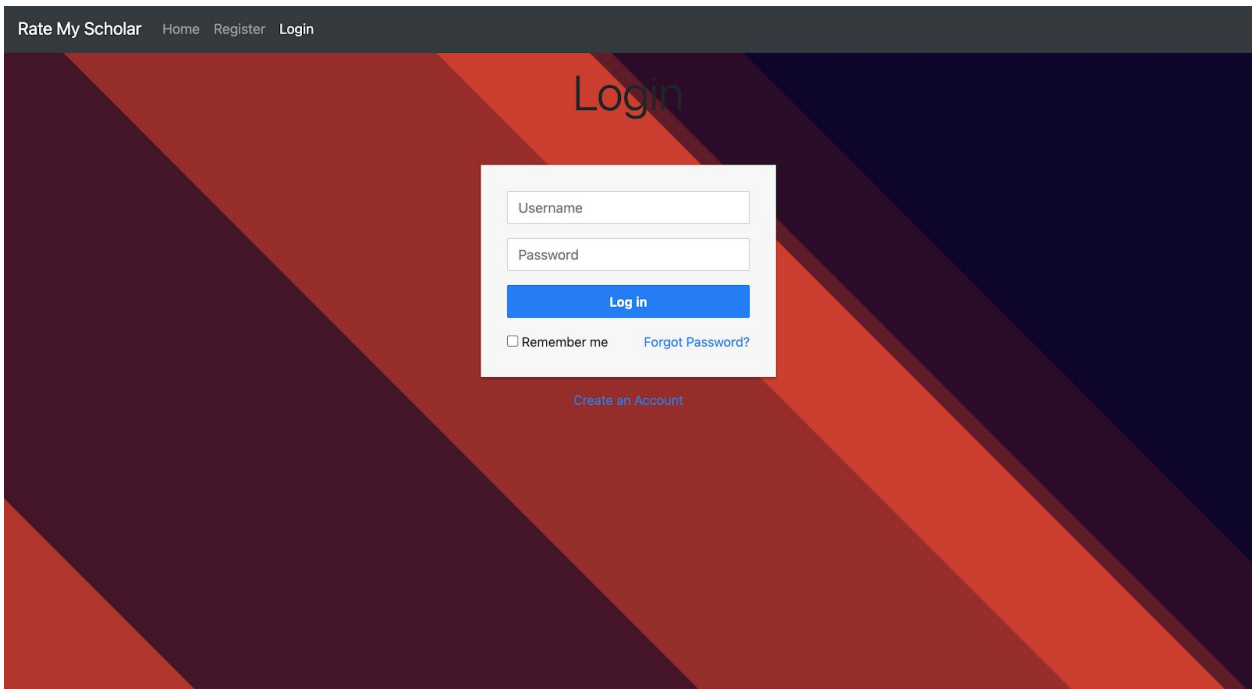
This is the fundamental design of the web application we are developing as our system and it is an admittedly very simplistic architecture due to the nature of the system of rate my scholar in which users effectively visit the site to search for a scholar and read/write details on the quality of the scholar.

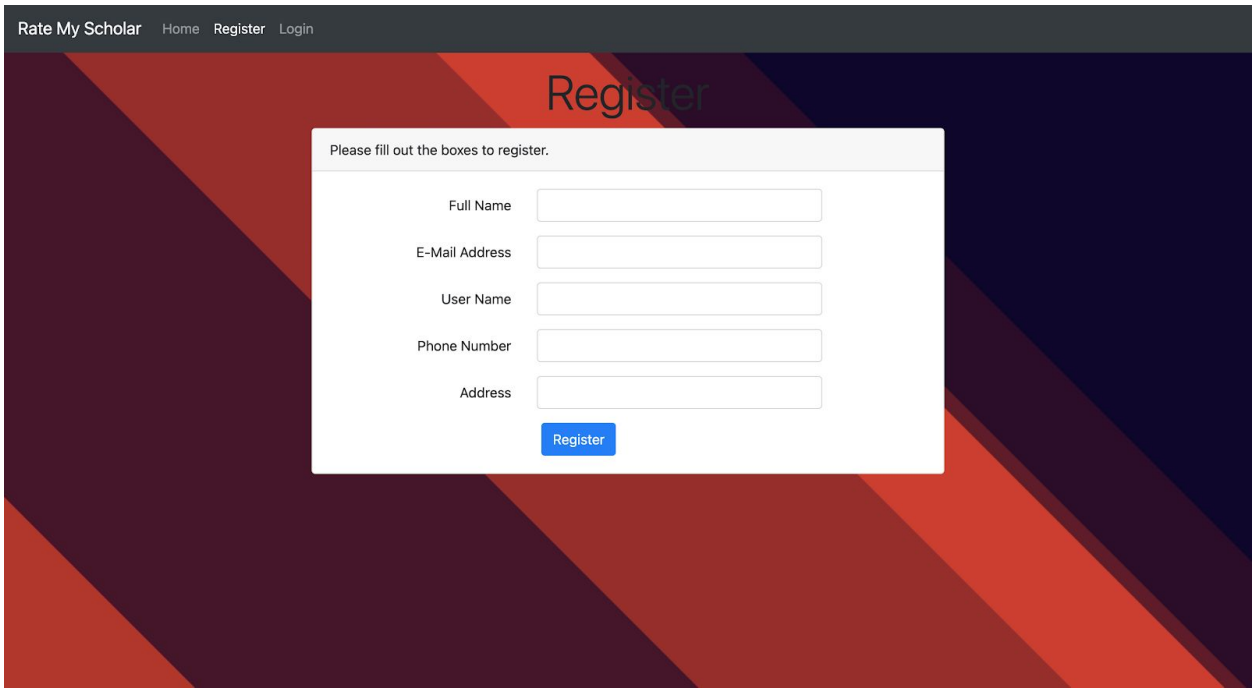
The user/client interacts with the web server housing our system in the form of a website application through registering and logging into the website in which that data gets sent to the server to validate their credentials and legitimize themselves as users. Once that is complete, the user, in turn, interacts with the database containing information on all possible scholars on all possible schools in its contents to yield a list of scholars for the user to peruse or review on which could potentially send data back to the database but otherwise displays it for the user to read primarily.

5. User Interface Design

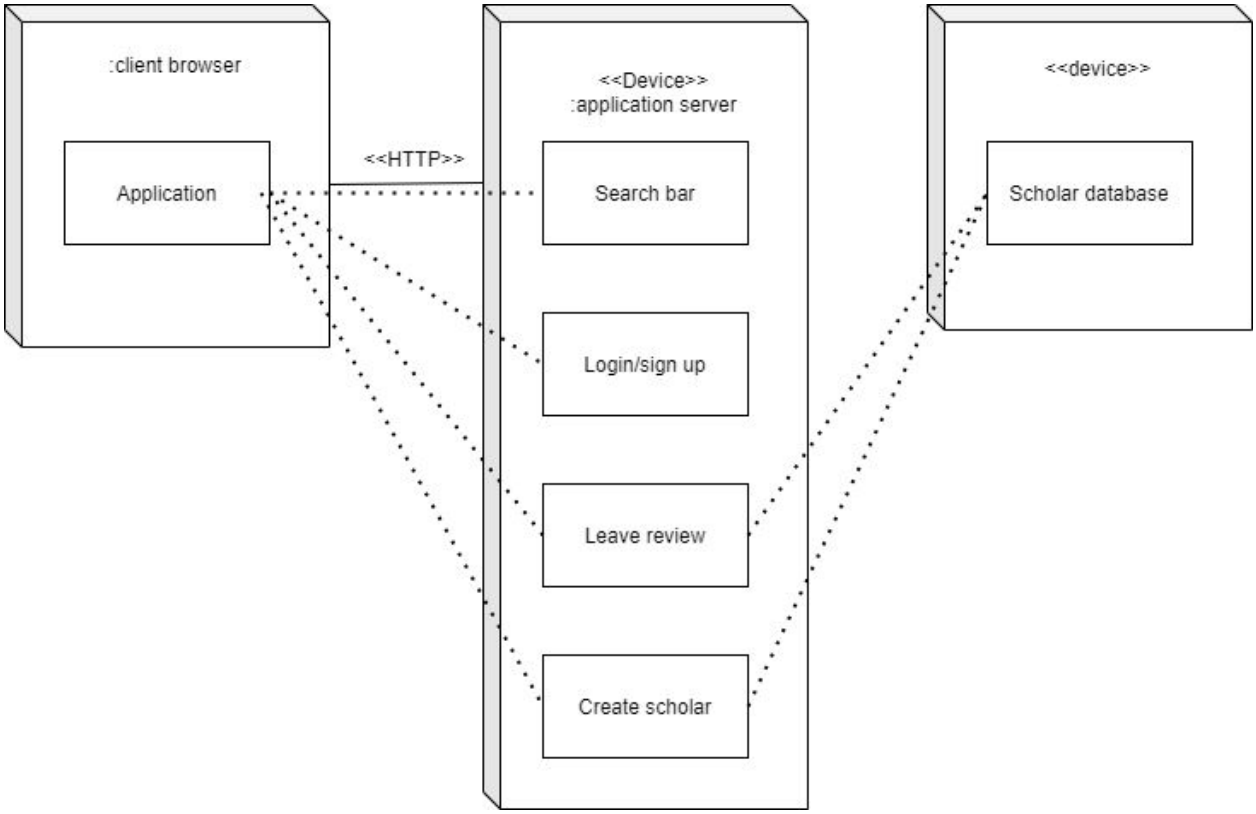

Rate My Scholar
Home
Register
Login
John Doe
Professor at Broccoli University in the Culinary Department
Quality: 3.5
More info
Rate this Scholar
Card image cap
Card title
Some quick example text to build on the card title and make up the bulk of the card's content.
Card image cap
Card title
Some quick example text to build on the card title and make up the bulk of the card's content.
Card image cap
Card title
Some quick example text to build on the card title and make up the bulk of the card's content.


Rate My Scholar
Home
Register
Login
Welcome to Rate My Scholar
With this complex and incredible site, you can peruse the quality of all your colleagues. Not just your professor.
Begin your search by entering a school


Rate My Scholar
Home
Register
Login
Login
Username
Password
Log in
Remember me
Forgot Password?
Create an Account


Rate My Scholar
Home
Register
Login
Register
Please fill out the boxes to register.
Full Name
E-Mail Address
User Name
Phone Number
Address
Register

6. Deployment Diagram and Explanation



Explanation:

This diagram shows the structure of the run time system. It contains the nodes “Client browser”, “Application server” and lastly the “database”. The individual nodes have essential functions that when combined together allow for the full functionally. The client browser allows for interoperability and connects to the website via the HTTPS protocol and from there the user can access information via a server which retrieves the information from the database. The browsers supported will be Google Chrome, Mozilla FireFox, Opera and Microsoft Edge. The server will run using Heroku. The database will be stored on the server in MongoDB.