Software Requirements Specification
COSC 4P02

Version 1.0

Due: Sunday 7th Feb at 23:55

R8Scholar

Team Leader:
Twino Puthiakunnel / mp13ko@brocku.ca / 5564182 / mp13ko
Secondary Team Leader:
Seth Shickluna / ss16wn@brocku.ca / 6217558 / SethShickluna
Erikas Klimusinas / ek15kg@brocku.ca / 5903547 / ErikasK
Grant Nike / gn17az@brocku.ca / 6349302 / GrantNike
James Sargant / 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

1.0. Introduction

1.1 Purpose

COSC 4P02 R8Scholar Requirements Elicitation:

Stakeholders:

- Us (Developers)
- Professor Naser
- Users (brock students / people with a brocku email) - Anonymous
- Testers - other brock students
- Potentially the people being rated on the site

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

Use Cases:

- Create a review for professors, courses, departments, other
    - Comments
    - Rating
    - Option to add tags about professor characteristics
        - Hard marker
        - Timely
        - Lots of Homework
- Reply to Review
- Create new account
- Sign in to student account
- Students should be able to search up professors or courses
- General forum thread for each topic of discussion
- Viewing professor profile page
- Add Professor/Course/other to site
- Modify Professor/Courses/Other
- Search Feature

System Architecture:

- Django based
- Model - Database structure
- View - Site control
- Template - HTML templating

Risk Factors of the system:

- Leaked information
    - industry standard password encryption
    - AWS has security features built into the servers they host
- Site crashing
- Unprofessional comments
    - Run comments through a profanity filter before posting
    - Report function

2.0. Requirements

2.1 Functional Requirements

2.1.1 Sign Up

Use Case Name	Sign Up / Create Profile
Trigger	The user selects the “sign up” on the website
Precondition	The user has not created a profile for their email address
Basic Path	1. The system opens a form to enter information 2. The user will enter the correct format of data into the form 3. The system confirms the information with the database and the user 4. The profile is created and the user redirected to the profile page
Alternative Path	None noted
Postcondition	The database has been updated
Exception Paths	If the user has already created a Profile using that given email address then the use case abandons
Other (Notes)	This can only be used for each email address once

2.1.2 Log In

Use Case Name	Log in to profile
Trigger	The user selects the “log in” on the website
Precondition	The user has created a profile previously using a defined email address
Basic Path	1. The system opens a log in dialogue 2. The user inputs the correct format of data (text) 3. The database authenticates whether that user is allowed access 4. The user is redirected to the
Alternative Path	There is the possibility of being directed to the login via a direct link
Postcondition	The user is redirected and the log are updated
Exception Paths	If the user inputs invalid information there will be an error and it will allow you to re-enter the text.
Other (Notes)	

2.1.3 Log Out

Use Case Name	Log Out
Trigger	The user selects the log out option on the website
Precondition	The user was already logged in and accepted into the database.
Basic Path	1. The system prompts the user to make sure they want to log out 2. The system will redirect the user to the homepage of the website
Alternative Path	None noted so far
Postcondition	The user is no longer authorised within the database.
Exception Paths	The user decides against logging out of the website and thus the use case is abandoned.
Other (Notes)	

2.1.4 Edit Profile

Use Case Name	Edit Profile
Trigger	The user selects the Edit feature within the profile.
Precondition	The user must be authorised, be viewing the profile page they created.
Basic Path	1. The profile information form is presented in an editable view 2. The user edits the desired fields in order make changes 3. The system confirms before making any changes to the profile.
Alternative Path	None noted
Postcondition	The database is updated with the new information from the form.
Exception Paths	If the user updates the information with invalid content it will prompt them to fix it. If the user cancels the changes, there are no new changes and the use case is abandoned.
Other (Notes)	

2.1.5 Create Review

Use Case Name	Create Review
Trigger	The selects the review function on the page
Precondition	The user must be authenticated in order to be able to rate an entity.
Basic Path	1. The user selects their numeric rating of the entity in question 2. The system waits for the user to enter the review text and press the confirmation 3. The system prompts the user to confirm the score they have put in and the input text. 4. The system shows the user the review they just created
Alternative Path	The user may opt to not input any text in which case the system will confirm that, then placing a placeholder in the text field(s).
Postcondition	The database is updated and thus calculates the new average score for said entity.
Exception Paths	If the user does not confirm their entry the system will abandon the rating process and revert the database to the previous state.
Other (Notes)	The username of the user is optional to the publishing of a review.

2.1.6 Edit Review

Use Case Name	Edit Review
Trigger	The user selects the edit function on the review.
Precondition	The user must be authorised as well as being a review that is created by that user.
Basic Path	1. The system makes sure the input is of the correct type (text for a review or numeric for ratings) 2. The system updates the entities profile 3. The system shows the user the newly edited review
Alternative Path	None noted
Postcondition	The database is updated with the new information we get from the review section.
Exception Paths	If the user chooses not to publish said edit the system will leave the editing view and the use case abandoned
Other (Notes)	This also includes the ability to remove a review

2.1.7 Search For Entity

Use Case Name	Search For Entity
Trigger	The user selects the search field in the website
Precondition	There is a non-empty database of entities and thus can be queried
Basic Path	1. The system reads the input from the user 2. The database queries for the given conditions of the search parameter 3. The results (entities) are displayed by the system 4. The user picks an entity from that list 5. The system redirects the user to the corresponding profile
Alternative Path	None noted
Postcondition	None noted
Exception Paths	If the user inputs a search term that does not correspond to anything we have stored in the database, the user will be prompted to change the search term

2.1.8 Write Comment

Use Case Name	Write Comment
Trigger	The user selects the comment section below a review that has been posted
Precondition	The user must be authorised and there must exist a comment to use.
Basic Path	1. The system awaits the input from the user of correct type (text) 2. The system confirms if that text if final 3. The system then shows the user the comment they made as well as the original review
Alternative Path	None noted
Postcondition	The database is updated with the new comment to the given review
Exception Paths	If the user chooses to cancel the comment the system should remove the text and revert to the previous view
Other (Notes)	Username optional.

2.1.9 Edit Comment

Use Case Name	Edit Comment
Trigger	The user selects the ‘edit’
Precondition	The user must be logged in and there must be a previously made comment by the same user
Basic Path	1. The system will wait for the user to enter the text 2. The system will confirm the text with the user 3. The comment text will be published replacing the old with the new text
Alternative Path	None noted
Postcondition	The database is updated, changing the previously saved text with the new text.
Exception Paths	If the user wants to, there is the ability to cancel the edit, leaving the old text unchanged in the database.
Other (Notes)	Username optional.

2.1.10 View Entity By Category

Use Case Name	View Entitiy by Category
Trigger	The user selects the option on the website to view “all” the members of some category.
Precondition	The category of entity (Professor, Course etc) should not be empty within the database
Basic Path	1. The system waits for the user to select the given category of entity. 2. The system will query the database for the given data type 3. The system displays the entities in that list 4. The system waits for further action from the user
Alternative Path	None noted
Postcondition	None noted
Exception Paths	If there is nothing within the database that matches the given category the use case should abandon and redirect the user back to the previous page (use case).

2.1.11 Contact System Administrator

Use Case Name	Contact System Administrator
Trigger	The user selects the “contact us” section of the website.
Precondition	There are no pre-conditions
Basic Path	1. The system presents the user with a form to contact the administrators of the system 2. There are text fields for the system to accept inputs from the user 3. The systems asks the user to confirm the message they are looking to send 4. The message is sent to the administrators, who can then respond to any issues raised
Alternative Path	The user may also find the form through an external hyperlink provided in some other medium i.e. email
Postcondition	There will be an update to users who are already authenticated, there will be an update to the database with the details of the contact the user has had with the administrators.
Exception Paths	The user can cancel any form of communication and not send through the message, in this case the use-case is abandoned and there are no postconditions for registered users.

2.1.12 Create Forum Post

Use Case Name	Create Forum Post
Trigger	The user selects the community section on and entity's profile
Precondition	The user should be authorised in order to gain access to the community section of the profile
Basic Path	1. The system waits for the user to input the text into the edit field 2. The system asks the user to confirm their message for the forum post 3. The system publishes the message and then allows for comments and interaction on the forum posts.
Alternative Path	None noted
Postcondition	The database is updated with all the information within the forum post and stored
Exception Paths	If the user cancels their intention of posting to the forum, the use case is abandoned and there are no postconditions
Other (Notes)	Username optional.

2.1.13 Comment on Forum Post

Use Case Name	Comment on Forum Post
Trigger	The user selects the comment function on a previously posted forum post
Precondition	The user must be authenticated within the system and there should also be a forum post already made.
Basic Path	1. The system waits for the user to input the text into the edit field 2. The system asks the user to confirm their comment for thatforum post 3. The system publishes the message and then allows for comments and interaction on the forum comment.
Postcondition	The database is updated with the messages written into the comments on that individual
Exception Paths	If the user cancels their intention to comment on the forum post there should be and abandonment of intention and no post conditions
Other (Notes)	Username optional.

2.1.14 Respond to Review

Use Case Name	Respond to Review
Trigger	The Entity selects the reply function on the reviews on the profile.
Precondition	The entity should be authorised and verified to be the responsible party for the review
Basic Path	1. The system waits for the user to input the text into the edit field 2. The system asks the entity to confirm their comment for that forum post 3. The system publishes the message to the review
Alternative Path	None noted
Postcondition	The database is updated with the message added to the review.
Exception Paths	If the entity decides to cancel their intention of replying to a comment the use case is abandoned.
Other (Notes)	

2.1.15 View All Reviews

Use Case Name	View All Reviews
Trigger	The entity can go to the page on which you can view all the reviews and ratings
Precondition	The entity should be authorised and verified to be the responsible party for the review
Basic Path	1. The system displays a section with all the ratings of that entity 2. The system displays a section with all the review messages written about that entity
Alternative Path	The entity may view all the ratings and reviews by clicking on their score on the profile.
Postcondition	None noted
Exception Paths	If the entity has no ratings and reviews published at that time, the system should notify them of that and abandon the use case.
Other (Notes)	

2.1.16 Search Through Reviews

Use Case Name	Search Through Reviews
Trigger	The entity selects the search feature on website
Precondition	The entity is already authorised and is on the page where they view all the reviews
Basic Path	1. The entity enters some form of keyword into the search field 2. The system queries the database for that given keyword 3. The system presents the results of the search, listing reviews that match the search term
Alternative Path	None noted
Postcondition	None noted
Exception Paths	If there is nothing matching the search term, the system presents an exception for the user.
Other (Notes)	

2.1.17 Voting System

Use Case Name	Voting system (Upvote/Downvote)
Trigger	The user selects the vote they desire on the website post.
Precondition	The user is authorised within the system
Basic Path	1. The system awaits the users selection of a vote 2. The system calculates the new tally (upvote - downvote) 3. The system displays the new tally to the user 4. The system indicates which vote is selected and goes back to waiting
Alternative Path	The user may vote directly from an external link to the post (possibly embedded)
Postcondition	The database is updated with the current tally of the votes
Exception Paths	The user may double click a vote and thus abandoning the use case and reverting all tallies.
Other (Notes)	

2.1.18 Report Post

Use Case Name	Report post
Trigger	The user selects the functionality to report
Precondition	The user must be authorised within the system
Basic Path	1. The system presents the user with a form for further information 2. The system takes the input and asks the user to confirm their report 3. The system posts the Report to the Moderator-level users 4. The system sends the user a message to confirm the report's status
Alternative Path	The user uses a direct link to reporting a post on the website
Postcondition	The database is updated with the information contained within the report.
Exception Paths	The user can cancel the intention to report a post and thus the use case is abandoned and the changes reverted.

2.1.19 Remove Review

Use Case Name	Remove Review
Trigger	The moderator selects the remove function within the moderation tools on the review
Precondition	The user MUST be of Moderator-level permission within the system or greater.
Basic Path	1. The system presents the moderator with a form for post removals 2. The system takes the input of the moderator and stores it 3. The system confirms with the moderator if they wish to proceed 4. The system removes the review and the profiles are refreshed (all calculations) 5. The system logs the removal and sends the reporting user the form from the moderator
Alternative Path	None noted
Postcondition	The database recalculates all the relevant data and updates
Exception Paths	The moderator can cancel their intent to remove the review and the use case is abandoned and there is no change to the database

2.1.20 Remove Comment

Use Case Name	Remove Comment
Trigger	The moderator selects the remove function within the moderation tools on the comment
Precondition	The user MUST be of Moderator-level permission within the system or greater.
Basic Path	1. The system presents the moderator with a form for post removals 2. The system takes the input of the moderator and stores it 3. The system confirms with the moderator if they wish to proceed 4. The system removes the comment and the profiles are refreshed 5. The system logs the removal and sends the reporting user the form from the moderator
Alternative Path	None noted
Postcondition	The database updates
Exception Paths	The moderator can cancel their intent to remove the comment and the use case is abandoned and there is no change to the database

2.1.21 Remove Topic

Use Case Name	Remove Topic
Trigger	The moderator selects the remove function within the moderation tools on the topic
Precondition	The user MUST be of Moderator-level permission within the system or greater.
Basic Path	1. The system presents the moderator with a form for post removals 2. The system takes the input of the moderator and stores it 3. The system confirms with the moderator if they wish to proceed 4. The system removes the topic and the profiles are refreshed 5. The system logs the removal and sends the reporting user the form from the moderator
Alternative Path	None noted
Postcondition	The database updates
Exception Paths	The moderator can cancel their intent to remove the topic and the use case is abandoned and there is no change to the database

2.1.22 Change Visibility

Use Case Name	Change Visibility
Trigger	The moderator selects the visibility function within the moderation tools on the post
Precondition	The user MUST be of Moderator-level permission within the system.
Basic Path	1. The system presents the moderator with a form for post visibility changes 2. The system takes the input of the moderator and stores it 3. The system confirms with the moderator if they wish to proceed 4. The system changes the visibility the post and the profiles are refreshed 5. The system logs the change of visibility
Postcondition	The database updates the status of the post
Exception Paths	The moderator can cancel their intent to remove the comment and the use case is abandoned and there is no change to the database

2.1.23 Ban User

Use Case Name	Ban User
Trigger	The Administrator will select the ban function within the Administrator tools on the user's profile
Precondition	The Administrator has the appropriate level of permissions
Basic Path	1. The system presents a form for the admin 2. The system awaits the confirmation of the input 3. The system confirms with the admin that they would like to remove the user 4. The user is removed and the change logged. 5. The user is notified by the system via a message
Alternative Path	None noted
Postcondition	The database is updated for the removal of the user and all scores across their reviews recalculated. This also results in the email address used in that user profile being blacklisted from the system.
Exception Paths	The admin may cancel their intent to ban the user from the system and then the use case is abandoned and the database unchanged.

2.1.24 Add Entity

Use Case Name	Add Entity (ReviewItem)
Trigger	The Administrator selects the feature to create a new entity on the administrator control panel
Precondition	The Administrator has the appropriate level of permissions within the system
Basic Path	1. The admin is presented with a form by the system 2. The system waits for the input of the fields necessary for that entity 3. The system asks the admin to confirm whether they would like to add the entity as is. 4. The system adds the entity to the list and logs the addition
Alternative Path	None noted
Postcondition	The database is updated with the new entity and its attributes and data.
Exception Paths	The admin can cancel their intent to create a new entity in the system. The use case is then abandoned and the database is unchanged.

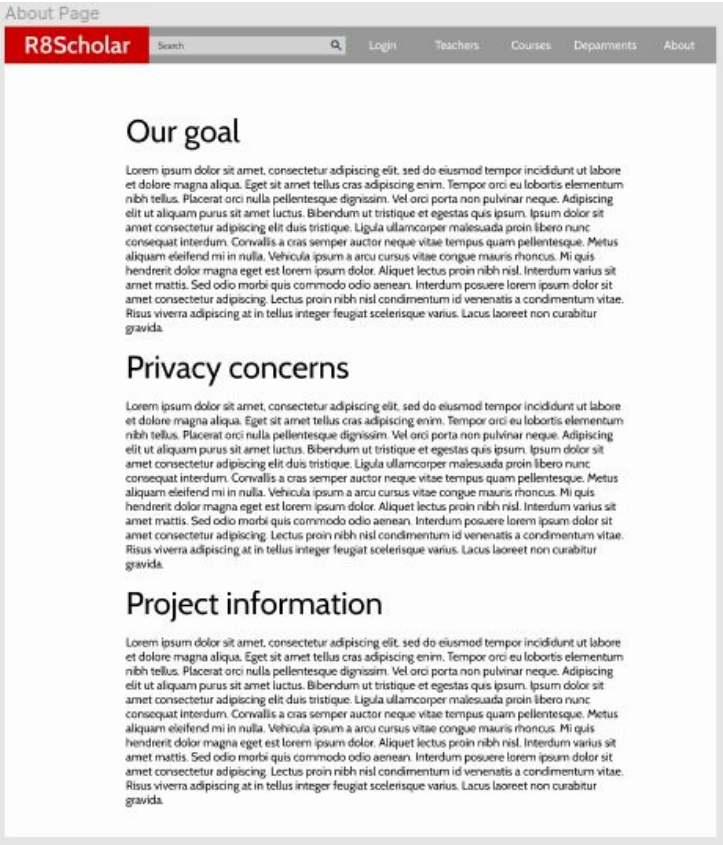
2.1.25 Remove Entity

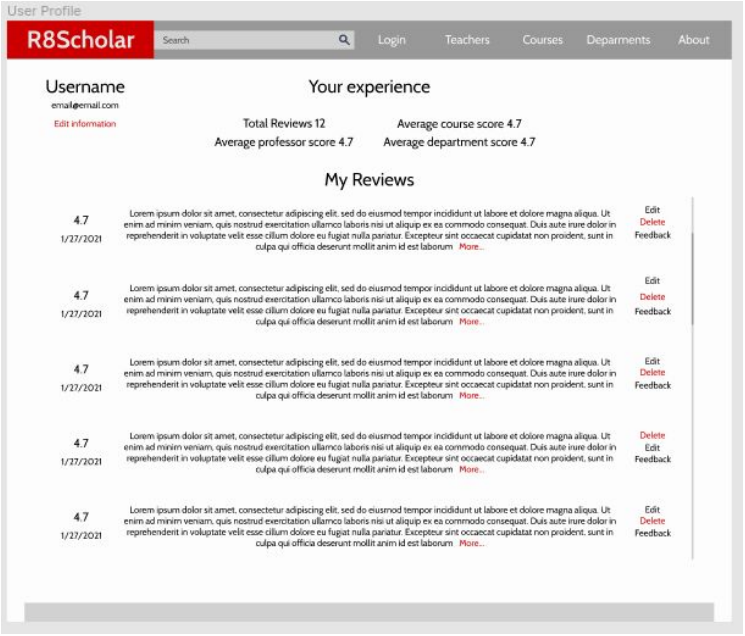
Use Case Name	Remove Entity (ReviewItem)
Trigger	The Administrator selects the remove entity function on the entity's profile.
Precondition	The Administrator has the appropriate level of permissions within the system
Basic Path	1. The system present the admin with form for entity removals 2. The system confirms that the user would like to remove the entity 3. The system removes the entity from the system 4. The removal is logged within the system.
Alternative Path	None noted
Postcondition	The database is updated after the removal of the entity.
Exception Paths	The admin can cancel their intention to remove an entity

2.2 User Interface Requirements

- UI must deliver system functions to users in a streamlined and easy-to-use way
    - Options to navigate to each page allowed by their user type
        - User
            - Login
            - Signup
            - Courses
            - Instructors
            - Departments
            - Home
            - About
        - Moderator
            - Moderation Pages
        - Admin
            - Admin menu
- Each page to display navigation and a footer
            - Settings
            - Signout
        - Display courses, departments, instructors
    - Navigation bar with links to each page
        - Navigation: Signed in
            - Display account drop down menu
    - Links to account
        - Display home, about, login, signup
    - Footer: Github link Other Info that would go in a footer such as a contact email or link to about page
- Home page
    - Displays to each of courses, departments, instructors (top ones)
    - Statistics
    - Information
- About page
    - Shows information about the project, the developers and FAQ
- Sign In / Sign up
    - Forms to complete the required actions
    - Login: Link to signup
- User Profile:
    - Show the user's email
    - Give option to edit profile information
    - Show statistics on reviews left by that user such as their average rating among other things
    - In a list, show all reviews left by that user alongside buttons which allow for editing deleting and viewing comments on them

- Courses
    - Display all courses in a list
    - Option to create a review
    - Filter by Course
    - Option to Search Courses
    - Courses/course_name
        - Show all reviews for course_name in a list
        - Department which course belongs to
        - Clicking on review shows the in depth review
            - Displays comments
        - Average Rating
- Departments and Instructors should have the same UI design as courses
- Creating Reviews
    - Large text box to write content
    - Boxes to input score out of 5 on predefined criteria
        - Lectures
        - Feedback
        - Organizational Skills
- Visual Example of UI Design





2.4 Stakeholder Requirements

- Minimal ads.
- Professional visuals.
- Searchable using names, course names, departments with autocomplete.
- Contact to get reviews removed.
- Be able to rate staff, and departments. May be able to rate TAs as well.
- Review students that professors worked with in Project Courses.

2.5 Non-Functional Requirements

2.5.1 Product Requirements

Requirements which specify that the delivered product must behave in a particular way.

Usability
- There should be an option for colourblind or other colour scheme themes in order to have the site be usable by as many different types of users to be given access.

Efficiency
- The search should be efficient (auto completing or querying as they type).
- Any course can only have one course page.

Security
- The authentication is only for brocku.ca domain accounts and accounts should be validated as real Brock accounts
    - User account information should be handled and stored securely(i.e hash passwords before storing)
- All emails are stored within R8Scholar.
- All passwords are hashed.

Dependability
- Only one account for each email address.

2.5.2 Organizational Requirements

Requirements which are a consequence of organizational policies and procedures e.g. process standards used, implementation requirements, etc.

Operational
- Site must update courses and professors as they are added
- Must have the ability for a moderator to remove comments
- Must update displayed statistics each time a review is made

Developmental
- Ability to add entities to database
- Remove entities
- Must look professional

2.5.3 External Requirements

Requirements which arise from factors which are external
to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Regulatory

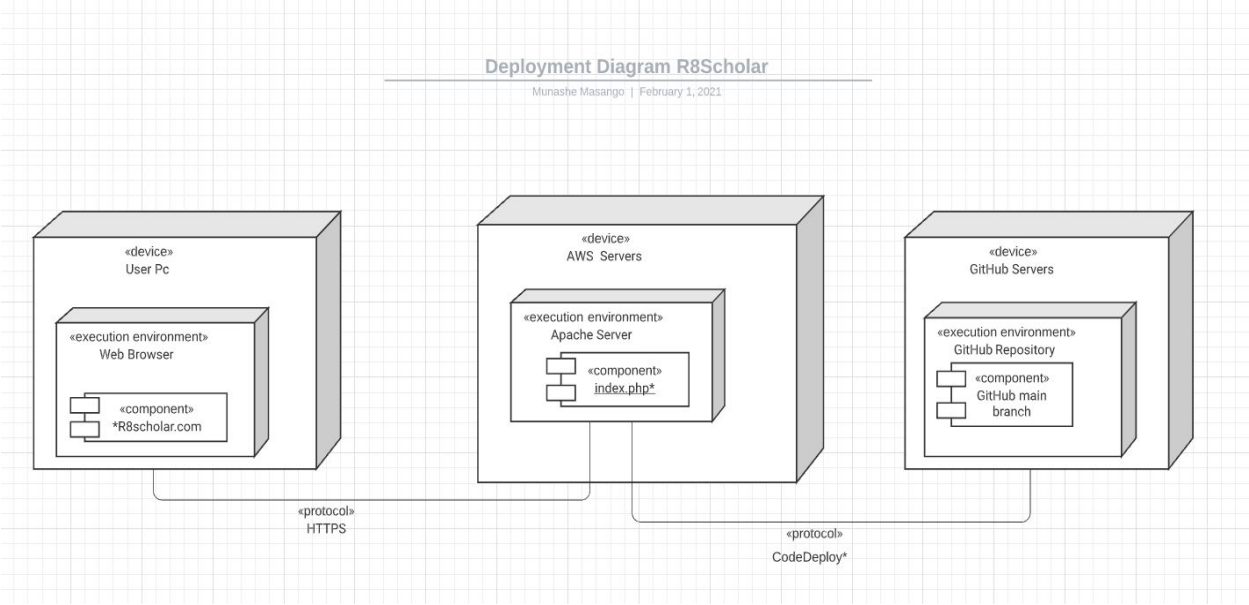
- All information on site comes from publicly available data.
- Any comments/reviews on site are not reflective of R8Scholar or it’s teams opinions.

Ethical

- All software used is Open Source and used as allowed by their respective licenses.
- R8Scholar reserves the right to remove any posts it deems unethical or abusive.

3.0 Architectural Design

3.1.1 Deployment Diagram



3.1.2 Deployment Explanation

User PC

- Web Browser
    - The Web Browser is the primary interfacing method for the users of the product specifically. There is a connection from the users' computer to the AWS servers via HTTP(S) protocols. This connection with the server will then be further secured within the server using the user's authentication details (which they specify and should remember).
- R8Scholar website
    - Is an artifact that represents the instantiation of the website for the user. This is presented via web browser given that the browser supports all the requisite technologies.

AWS Servers

- Apache Server
    - Barebones Linux server supported by Amazon Web Services. The Apache technologies involved include those that AWS supports (like Amazon Keyspaces). The reasoning for using AWS and Apache is that there are advantages in performance when scaled. This website is ideally going to be used by many people in the future, this allows us to grow the tables in the required databases. Other useful features include being highly available and secure while

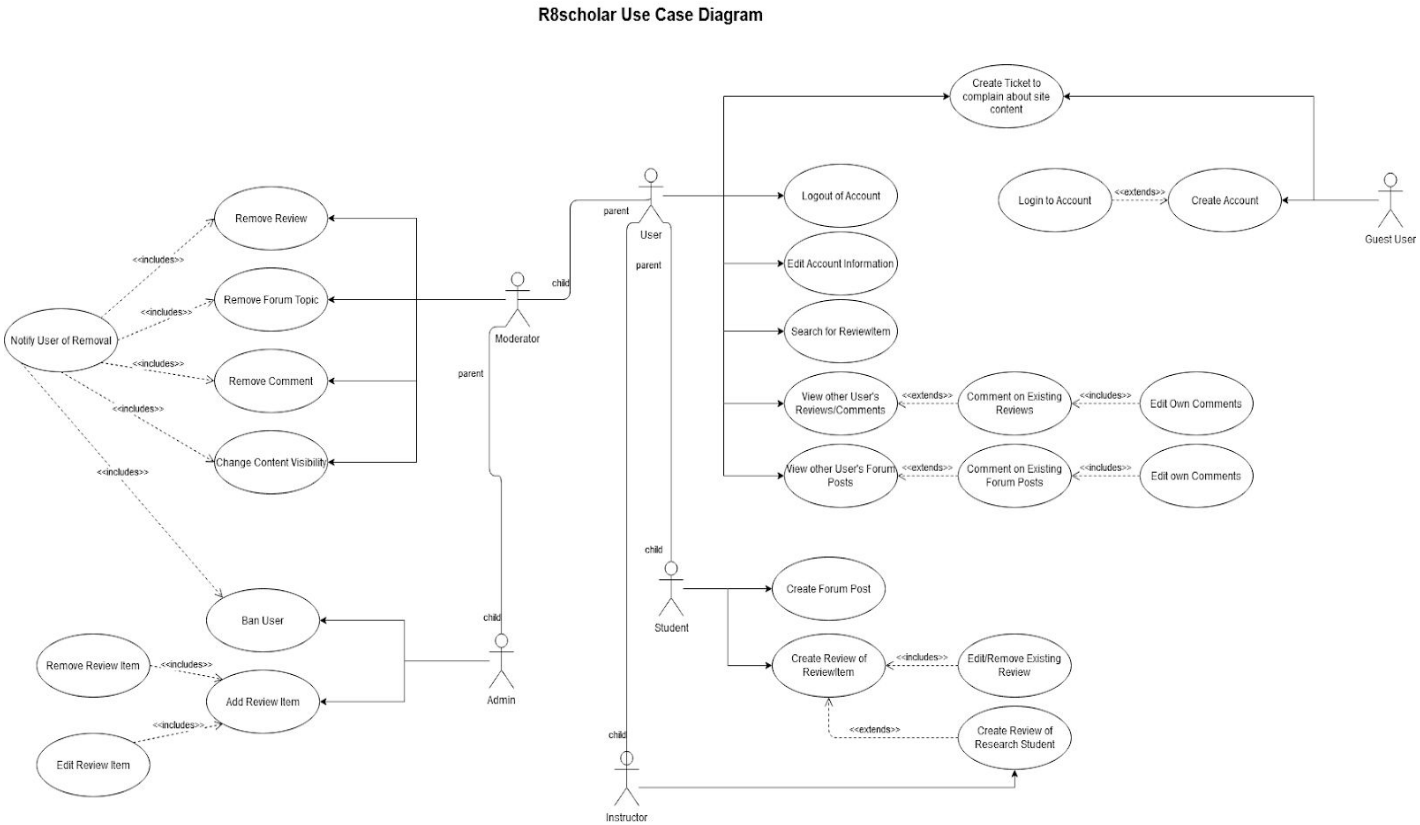
still allowing the development team to minimise costs by using open-source softwares.

- Database files (index.php*) - is an example file that demonstrates that the Apache server can access instances of files within the database to present as web pages to the user, via the connection method above.

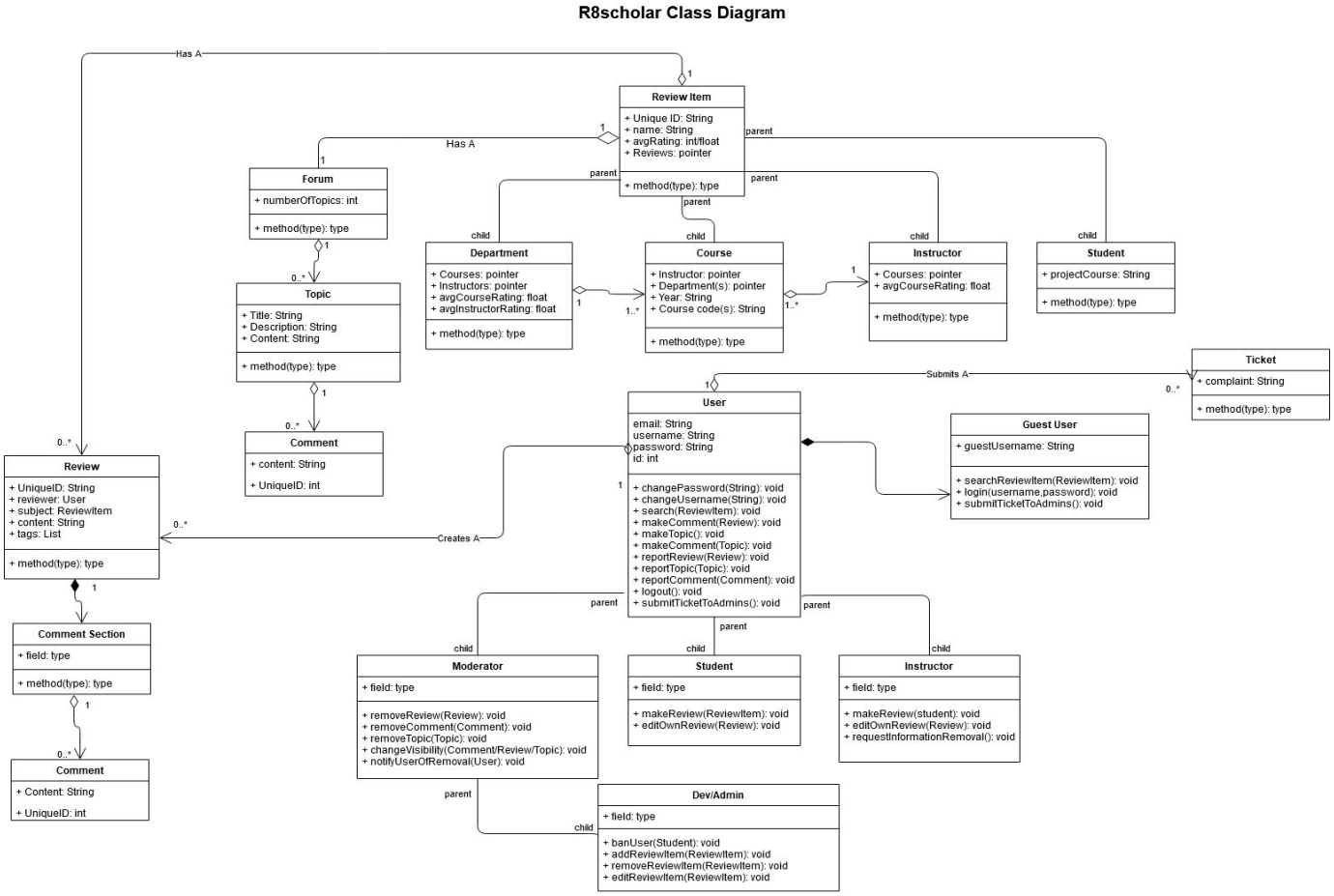
GitHub Servers

- Repositories
  - The GitHub repository is used for code version control within the product life cycle the AWS server is connected to the GitHub Servers via technologies supported by the server and the like (CodeDeploy). The repository
- Main Branch
  - The main branch will be the instance of the current (or latest deployment) version of the website. The AWS server will pull the versions of the code from this branch. The use of this method for version control and updating the server are useful as they provide some key features to the developers of the product. The developer is able to minimise downtime on the website, centralise any control and collaborate with a group effectively, while maintaining the integrity of the data.

3.2 Use Case Diagram(s)

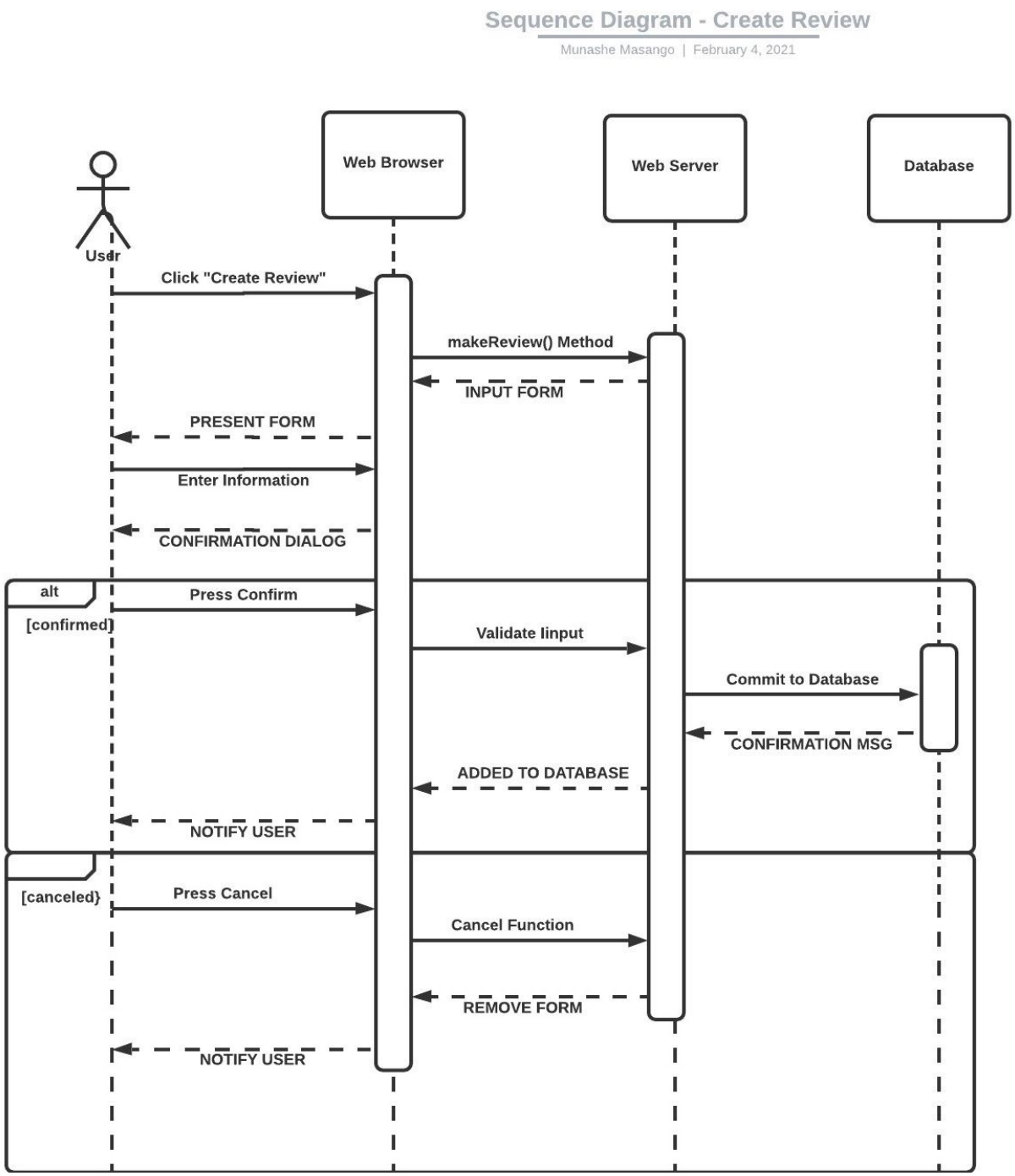


3.3 Class Diagram(s)

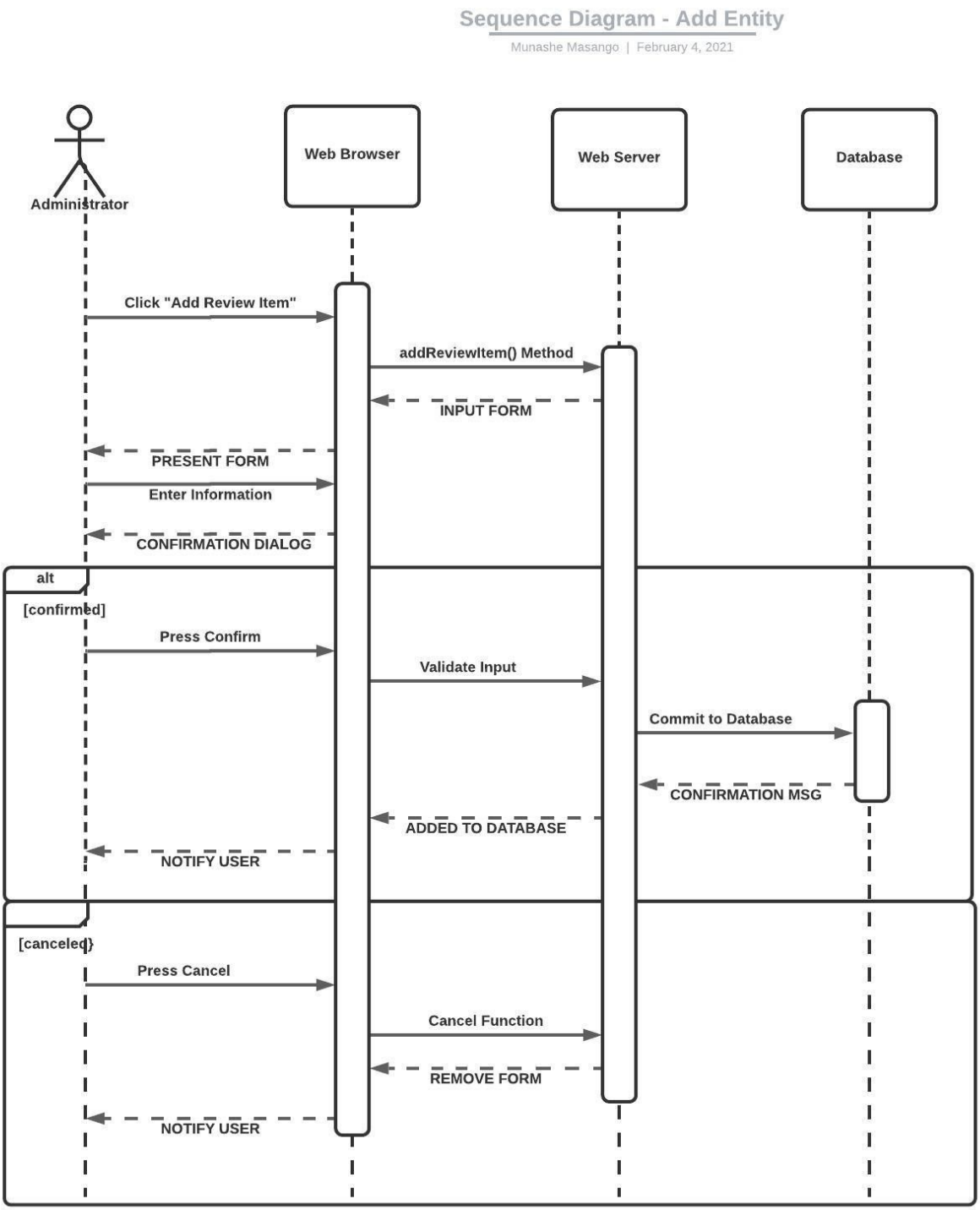


3.3 Sequence Diagram(s)

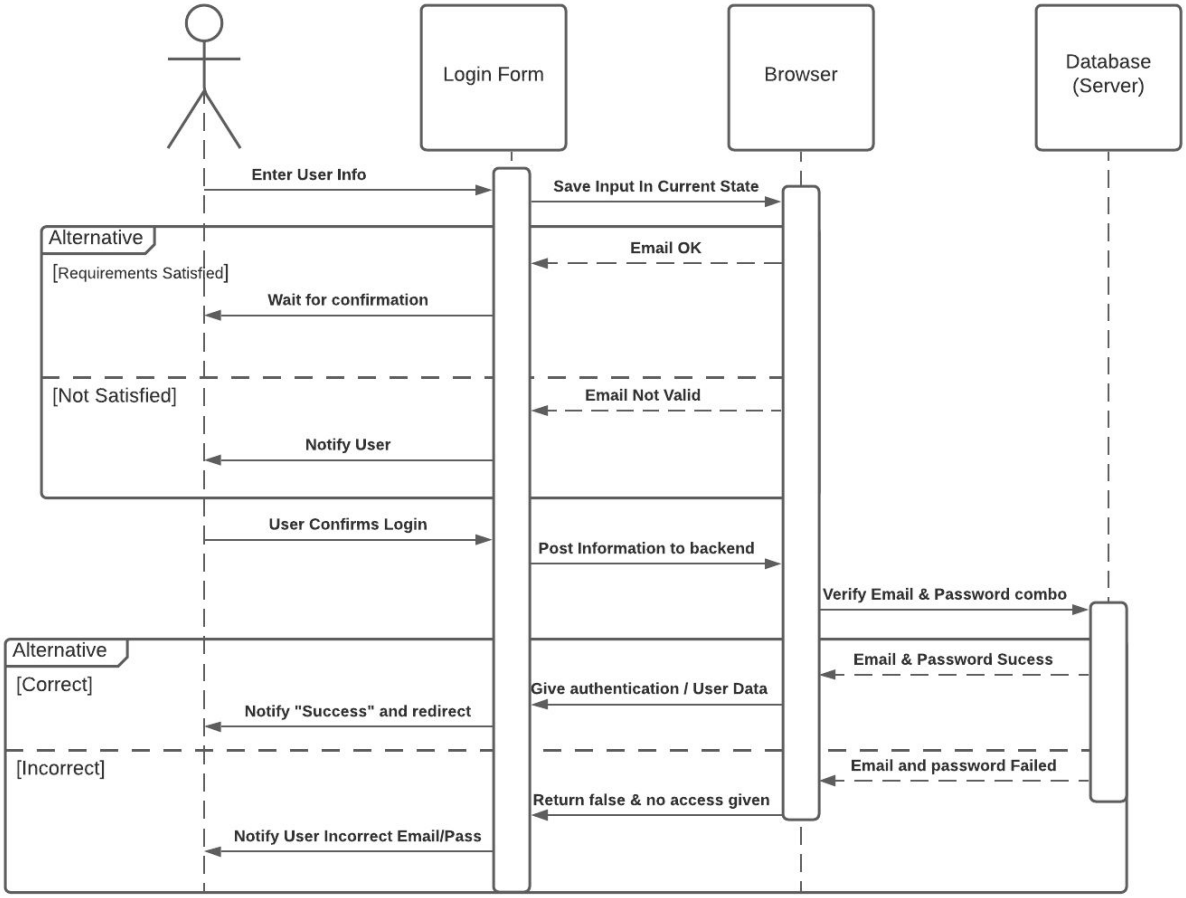
3.3.1 Create Review



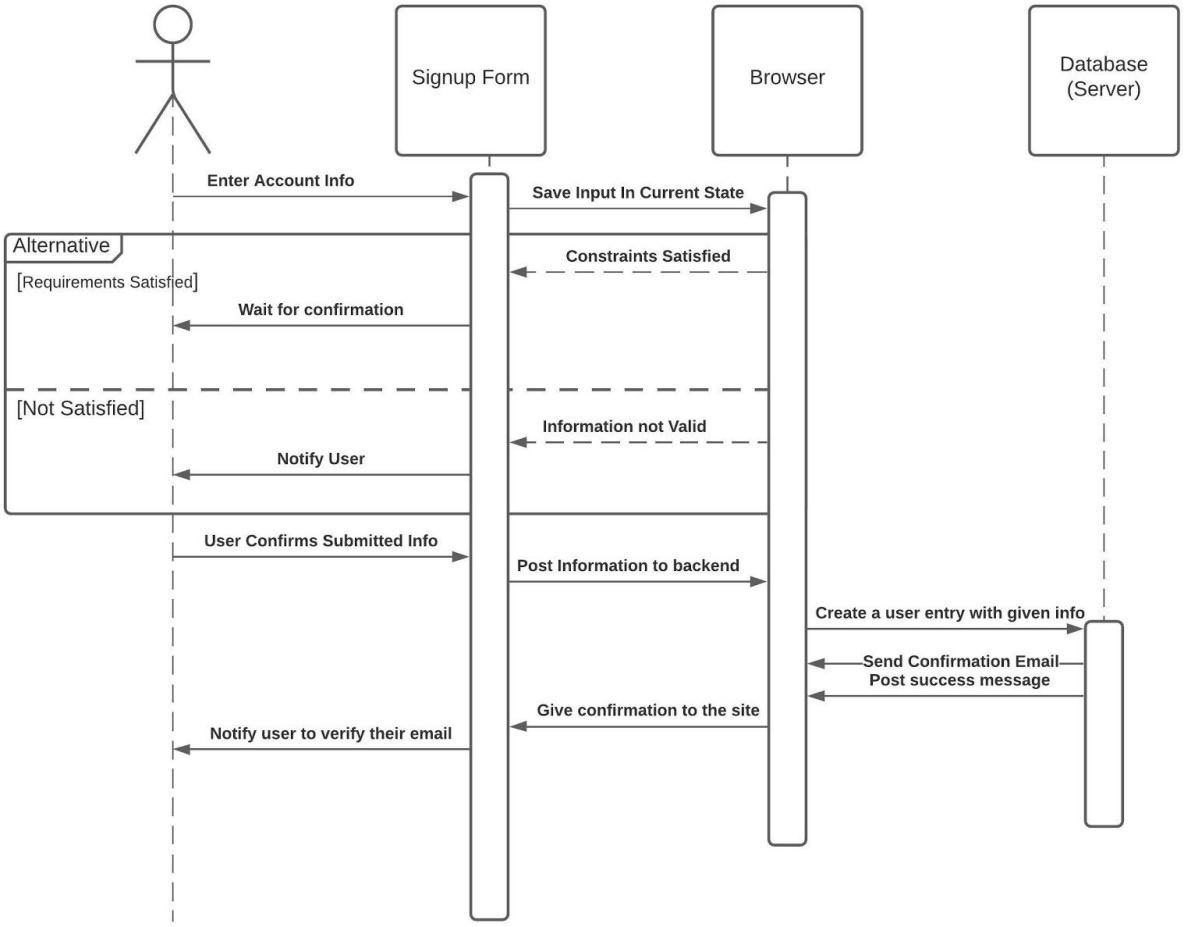
3.3.2 Add Review Item



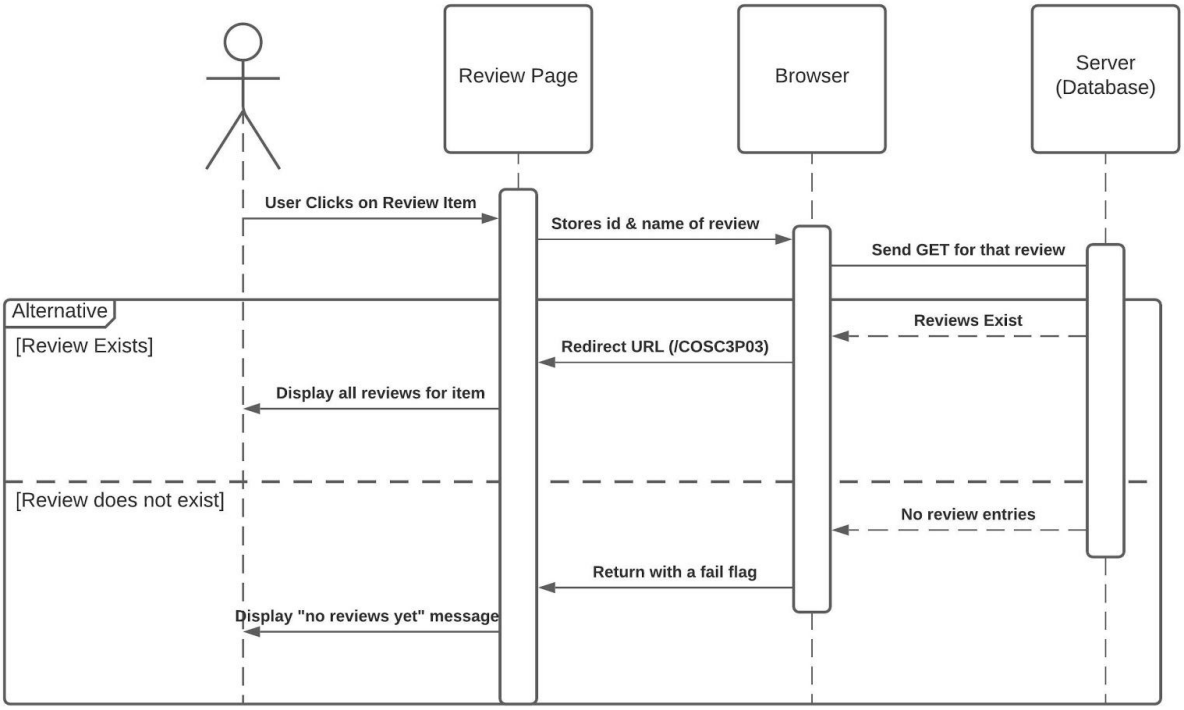
3.3.3 Login


Login Form
Browser
Database (Server)
Enter User Info
Save Input In Current State
Alternative
[Requirements Satisfied]
Email OK
Wait for confirmation
[Not Satisfied]
Email Not Valid
Notify User
User Confirms Login
Post Information to backend
Verify Email & Password combo
Alternative
[Correct]
Email & Password Sucess
Give authentication / User Data
Notify "Success" and redirect
[Incorrect]
Email and password Failed
Return false & no access given
Notify User Incorrect Email/Pass

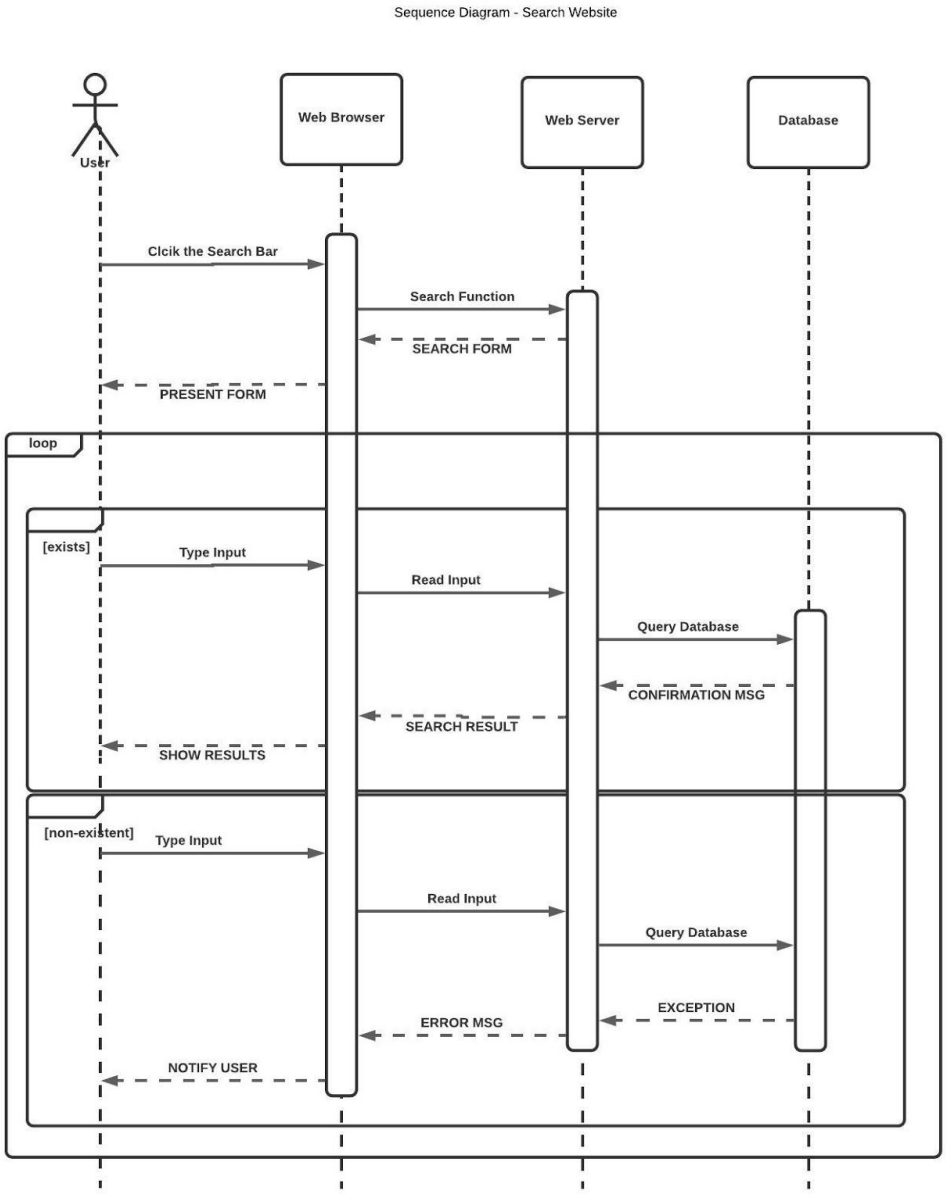
3.3.4 Signup


Signup Form
Browser
Database
(Server)
Enter Account Info
Save Input In Current State
Alternative
[Requirements Satisfied]
Constraints Satisfied
Wait for confirmation
[Not Satisfied]
Information not Valid
Notify User
User Confirms Submitted Info
Post Information to backend
Create a user entry with given info
Send Confirmation Email
Post success message
Give confirmation to the site
Notify user to verify their email

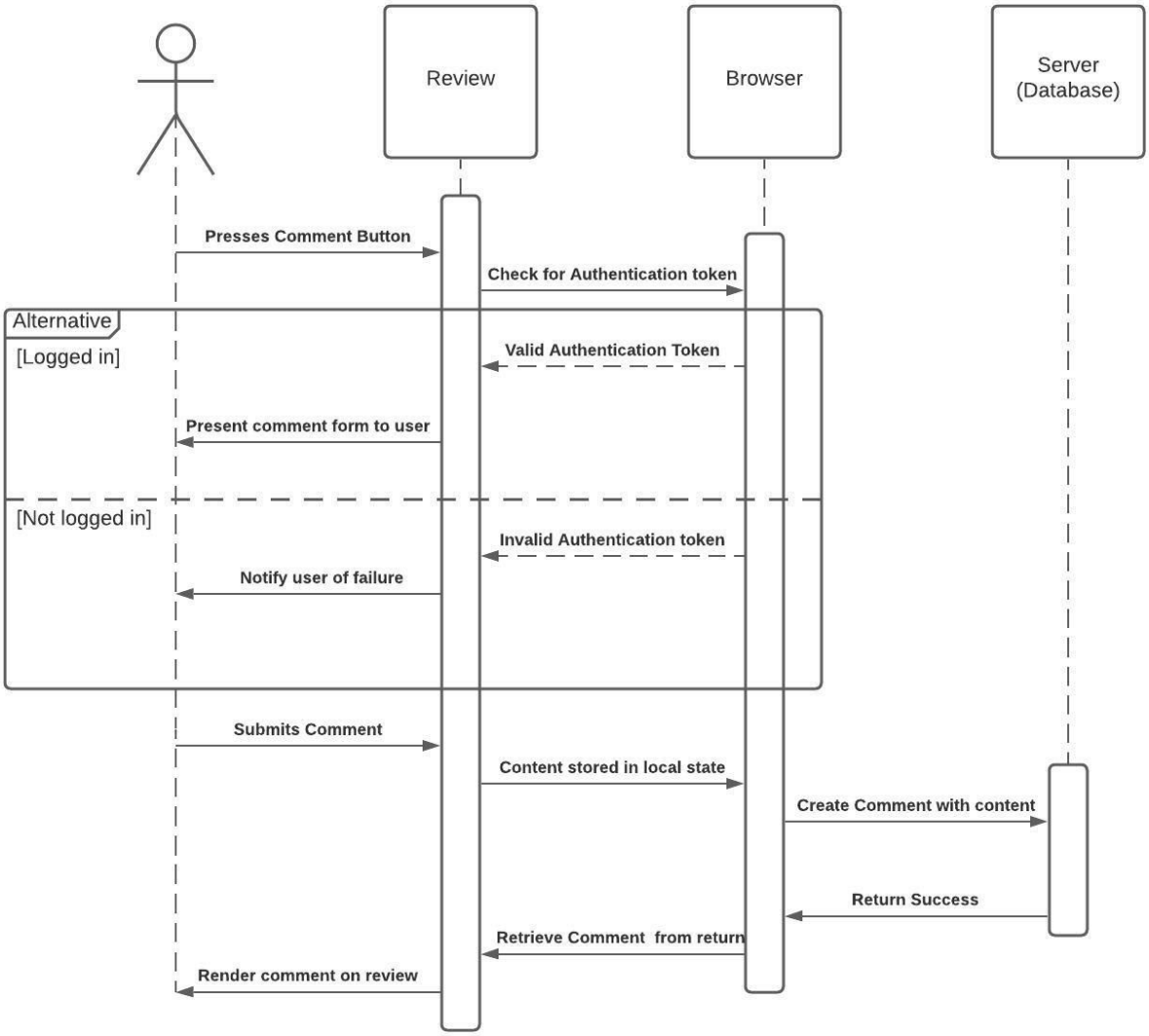
3.3.5 Selecting Review Item



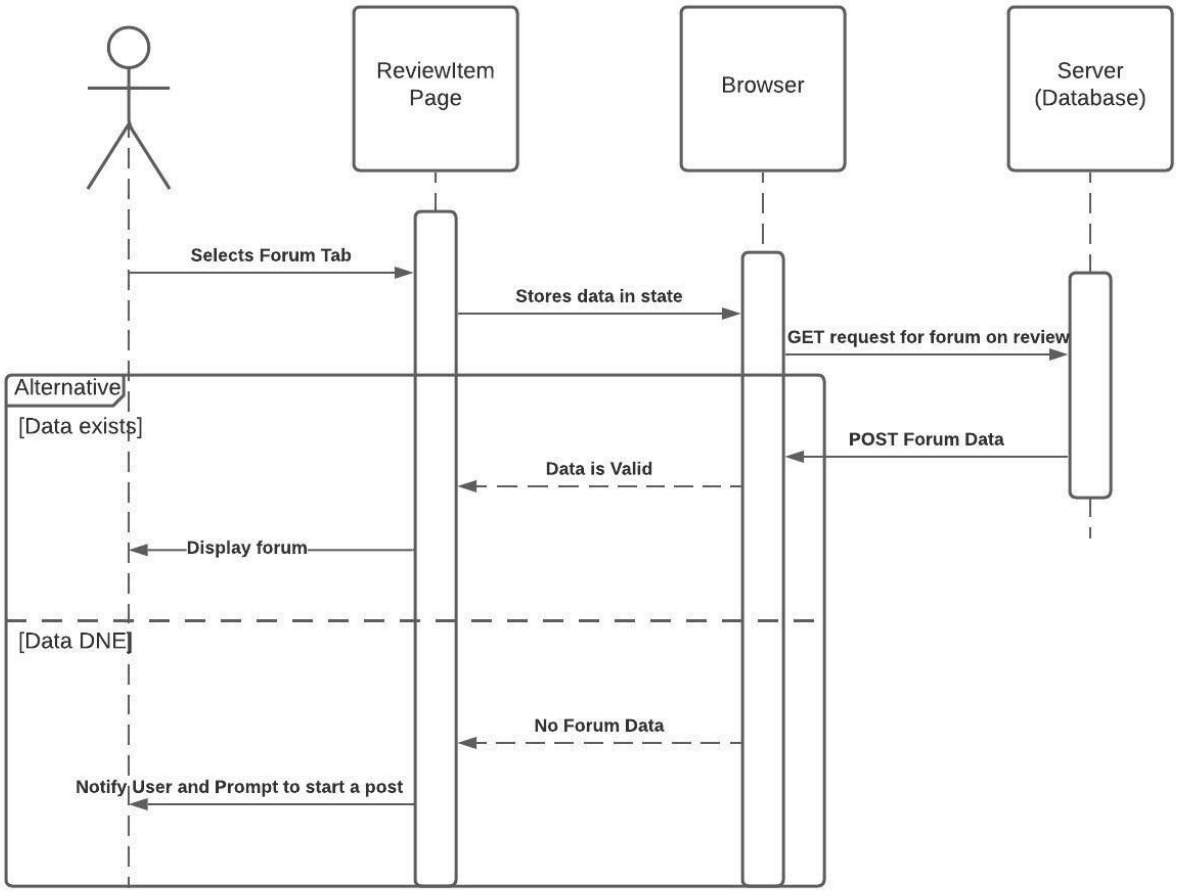
3.3.6 Search Website



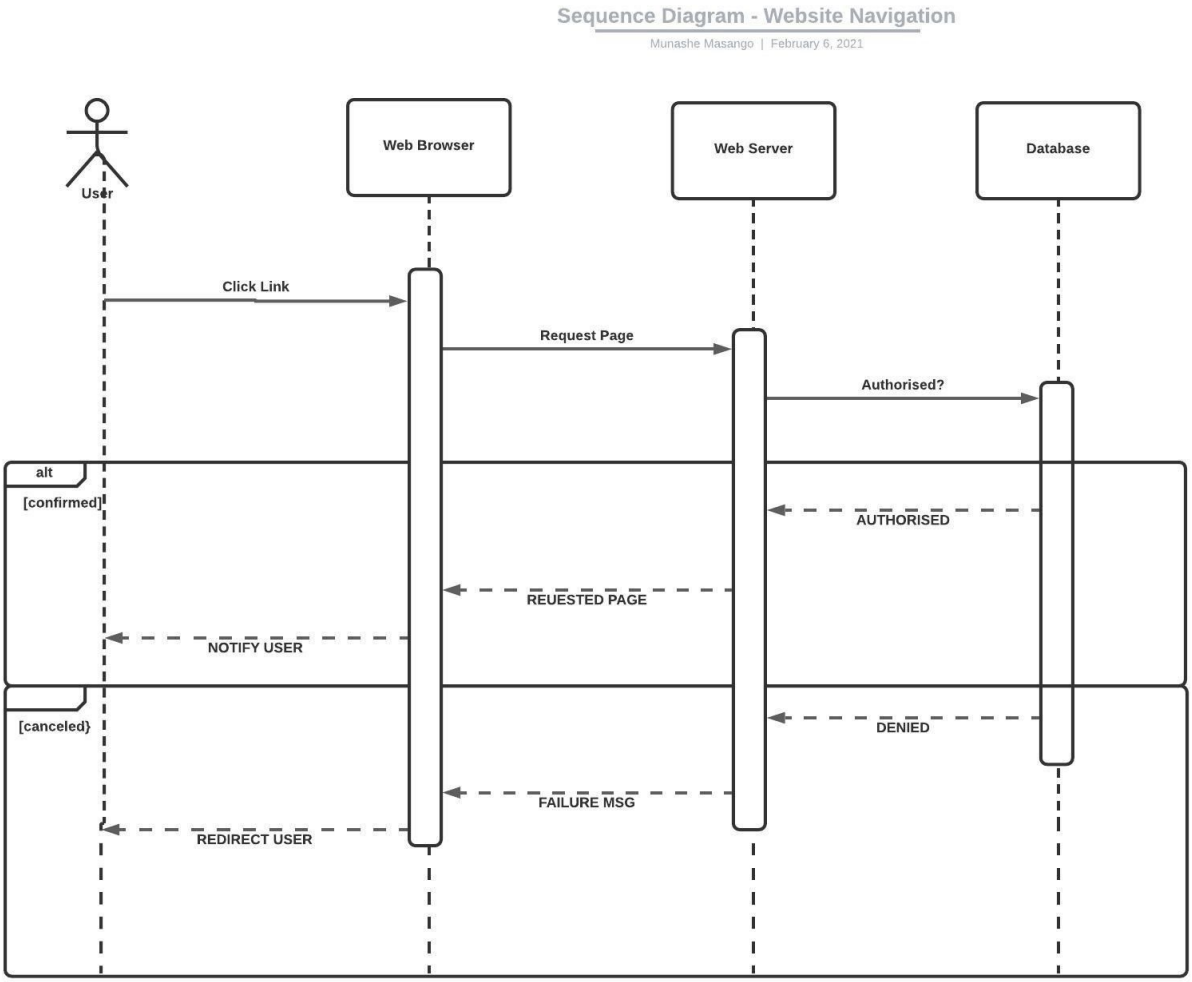
3.3.7 Comment on Review



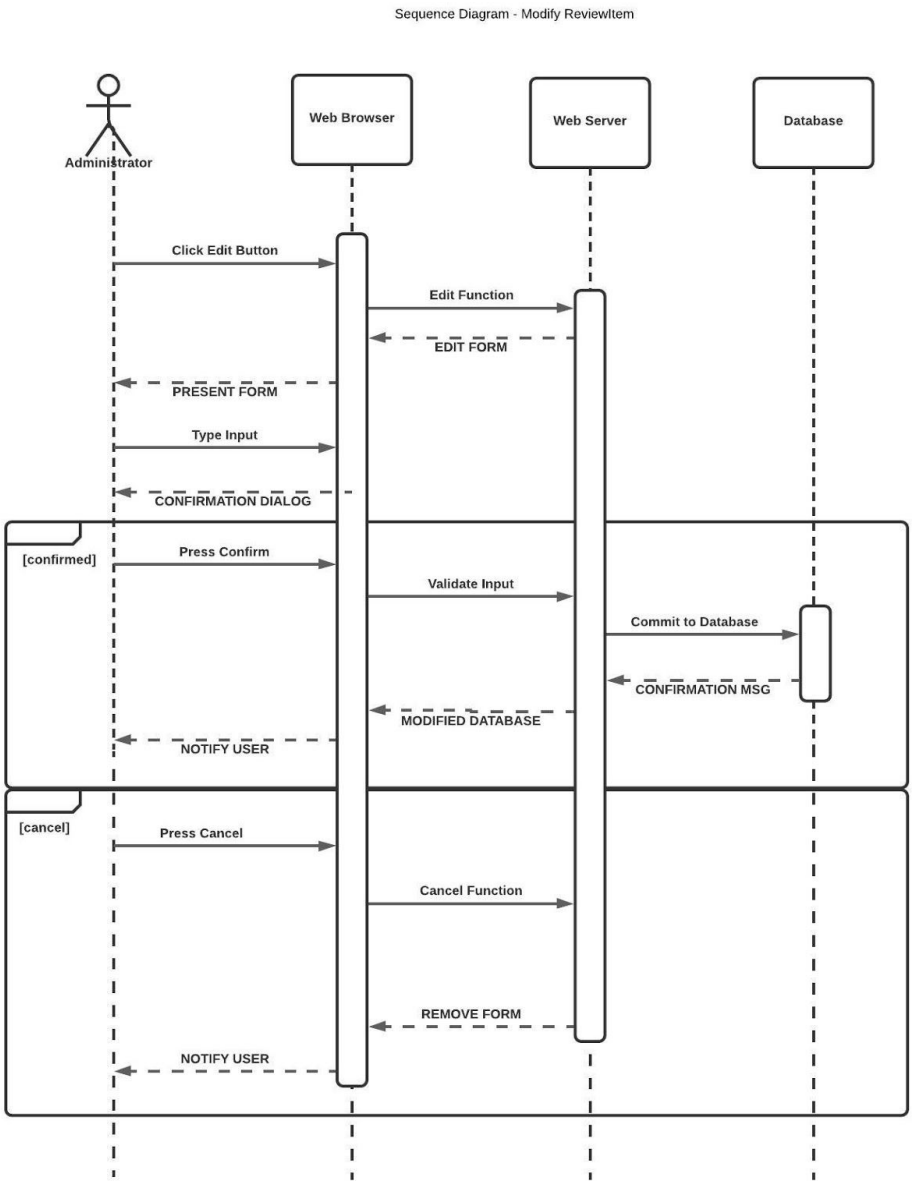
3.3.8 Accessing Forum



3.3.9 Navigation

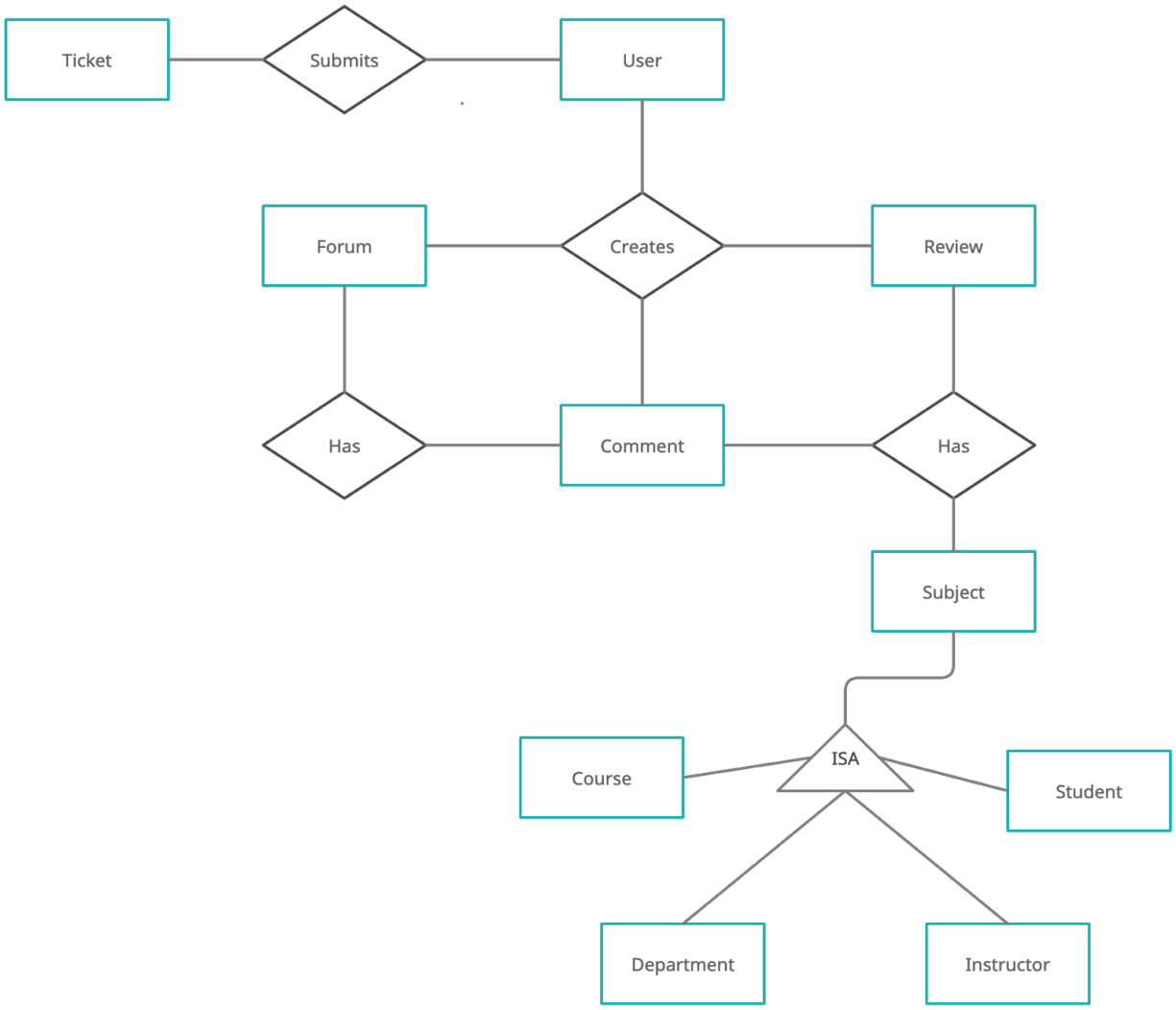


3.3.10 Edit Review Item



4.0. Database Design

4.1 ER Diagram



4.2 Database Tables

4.2.1 User Entity

Data Item	Type	Description	Comment
Email	Text/String	Email of the user	Primary Key
Username	Text/String	Username submitted on signup	
Password	Encrypted text/String	Account password	This would be encrypted (only the user themselves would know what it is)
Reviews	Pointer	Review Entity	This would point to each review this user left
Comments	Pointer	Comment Entity	
Forums	Pointer	Forum Entity	
UserType	String	Type of user (dev, student, admin, instructor)	Handles permissions, etc
isVerified	boolean	Used to determine if the user has verified their email	Probably done by sending a code to the provided email
verificationCode	string	Generated upon account creation to verify the given email	Code is emailed to user then they input it Into a first time page
DateCreated	Date	Date account was created on	

4.2.2 Subject Entity

Data Item	Type	Description	Comment
name	String	Name of dept/instructor/student/course	Course full name [dept+code], primary key

4.2.3 Course Entity

Data Item	Type	Description	Comment
Department	Text/String	Department in which the course is from	ex: COSC
Code	Text/String	Course code	ex: 2P03
Avg Rating	int	The average score this course was rated	
Reviews	Pointer	Review Entity	reviews for course
Instructor	Pointer	Instructor Entity	Instructor who taught this course
Aliases	Pointer	Course entity → cross-listed courses	

4.2.4 Instructor Entity

Data Item	Type	Description	Comment
department	Text/String	Department in which the course is from	ex: COSC
Avg Rating	int	The average score this course was rated	
Reviews	Pointer	Review Entity	reviews for this prof
overallCourseScore	int	Overall score from course reviews	

4.2.5 Review Entity

Data Item	Type	Description	Comment
Id	number	Id number of the review (for book keeping), primary key	Generate from date + random string?
reviewer	User	Points to the user entity who wrote this review	
subject	Subject Entity	Points to the subject (professor / course)	

		of the review	
content	Text/String	Text review written by a user	
Rating	int	Score this rating gave	
reports	int	Amount of times this review has been reported	
tags	text/String	Pre-defined tags about the course/prof for ease of use	
comments	Comment Entity	Points to each comment that has been left on the review	This could be any number >= 0
Date	Date	Date the review was made	
visibleTo	string	Denotes who this review is visible to	Ex: visibleTo: ‘public’

4.2.6 Comment Entity

Data Item	**Type**	**Description**	**Comment**
Comment Id	int	Id number of the review	Combined key
Review Id	int		Combined key
Name	string	Name of commenter	
Content	string	Comment nested in review	
Child	Comment entity	Points to all replies to this comment	
DatePosted	Date	Date which the comment was made	
reports	int	Amount of times this review has been reported	

4.2.7 Forum Entity

Data Item	Type	Description	Comment
id	Int	Primary key	Not sure maybe date + number?
Subject	Subject Entity	Course, prof, dept, student that the forum is regarding	
Title	String	Basically the top level content of the forum. Question/statement user addressed in forum creation	
Comments	Comment item	Top level comment on a forum	

4.2.8 Department Entity

Data Item	Type	Description	Comment
Id	number	Id number of the review	
Name	string	Name of department	
Avg Rating (Overall score)	int/Numerical	The average score this course was rated	
Reviews	Pointer	Review Entity	reviews for this dept
Overall course score	Int		
Overall Professor Score	Int		

4.2.9 Ticket

Data Item	Type	Description	Comment
Id	int	Primary key	
user	User	Person submitting the ticket	
content	text/string	Information	
DateSent	Date		