

Submission 04 - QuickQuestion

Loss of Team Member:

Unfortunately, we have lost a team member from this project. Joshua Brenner has dropped the class.

Outstanding bugs:

1. *Bug:* Clicking on the “Current Classes” or “Subjects” tabs at the top will route the user ONLY to the Algebra II questions page. This occurs after clicking one of the dropdown options. This also occurs when the user clicks a section underneath “Course Participation” on the user profile page.

Plan to Resolve: The plan to resolve this issue is to investigate ways to route to a course with a specific course_id or class_name. Currently, we have a list of classes defined in our database.js. These are all defined by unique class names, currently. Essentially, the solution requires us to be able to choose any of these class tabs on our pages and populate the appropriate class components.

2. *Bug:* When a user tries to navigate to the question’s discussion, after clicking the “View Answer” and “Discussion” buttons on the questions page, the user is redirected off the page to a single discussion post.

Plan to Resolve: This requires the team figuring out a way to populate a list of discussions underneath the question. This may mean that our components need to be redesigned. For instance, a question should have a list of discussions, which will then be populated after clicking the “Discussion” button on the question page.

3. *Bug:* User has no ability to add a question/discussion post.

Plan to Resolve: This is something that the team will need to investigate in the near future. This essentially means defining a correct component that will populate a list of questions to a particular submission. Each class has submissions that contain questions and discussions. Ideally, the plan is to add components that will populate a list of questions. After this is done within the mock database and server, we can then examine how to add a question to this populated list.

4. *Bug:* Dynamic functionality issues. Dynamic components such as discussion posts and active discussion posts (found on the user profile page) do not work.

Plan to Resolve: The team will have to revisit some workshops and learn more about dynamically creating components on pages. Additionally, our components may need to

change to reflect this.

5. *Bug:* When clicking the “Class Discussion” under the questions page, user is redirected to the profile page.

Plan to Resolve: We will need to create a component for class discussion. Once created, it will display all the discussions that are occurring within a class.

6. *Bug:* Static content is still displayed on pages.

Plan to Resolve: Currently, we have implemented functions to populate data about a user on the profile page. These functions are listed in database.js. However, the team still needs to implement functions to retrieve data for questions, active discussions, discussions under questions, class data, and progress data.

7. *Bug:* Clicking on a question on the question page will redirect to itself.

Plan to Resolve: The team will need to implement a way to populate data for different questions. Currently, our link just redirects back to the current component we have defined for a question. This requires investigating certain methods that we will need in server.js.

8. *Bug:* React components not fully defined.

Plan to Resolve: We do not currently have all of our react components fully functional. The components that are not fully defined include: class.js, replyEntry.js, submission.js. The plan is to either integrate these components sometime in the near future or reevaluate our ER diagram and make changes to components accordingly.

React Components:

answer.js

Our component for Answer represents an answer that is displayed after answering a question. Each question will have different answers, so we need to populate the appropriate answer after answering a particular question.

class.js

Our component for Class represents a class.

discussion.js

Our component for Discussion represents a discussion post for a particular question. Discussion posts will be populated after the user clicks “Discussion” on the question page. In

the future, the user will have the ability to add a discussion post and reply to a discussion post.

leftnav.js

Our component for Leftnav is responsible for populating a list of questions for a class. In addition, this component is responsible for revealing a drop down menu which displays the classes that a user can navigate to. After clicking, a user will be brought to the class and displayed questions that belong to that class.

progressbar.js

Our component for Progressbar is responsible for populating the progress data that is tracked as a user uses QuickQuestion. This data is populated on the user profile page (one example on our page is: “Java [55%]”)

question.js

Our component for Question is responsible for populating a question onto the question page. With this component, the question is displayed along with multiple choice inputs, “Love it!!”, “Too Hard!!”, “Remove it!!” buttons, and “Ask professor”, “View Answer”, and “Skip” buttons.

reply.js

Our component for Reply is responsible for populating a reply underneath a Discussion post component. Each discussion post has a list of replies populated underneath it.

replyEntry.js

Our component for ReplyEntry will be responsible for the handling of entering a reply to a discussion post, similar to that of the Facebook clone’s CommentEntry component.

submission.js

Our component for Submission represents a submission to a class. Each class can have numerous submissions, all of which contain questions and discussions for each question.

topnav.js

Our component for TopNav represents the component for the navbar found at the top of each page. It is responsible for transitioning the user to different subjects, classes, and the user’s profile page.

userdiscussion.js

Our component for UserDisc represents the component that is responsible for populating all of the discussions that a user has participated in. This is shown on the user profile page after clicking on the “Discussion” tab.

userinfoprofile.js

Our component for UserInfo represents the component that is responsible for populating data about the user. The data that is populated contains information about a user’s academic

history and courses that they participate in.

Individual contributions

Ethan

- Answer, question, toptnav, leftnav react components
- App.js
- Minor edits to userinfo component for routing
- Routing with react router in app
- The start of server methods to access DB
- Redirect to make index page the user's page

Ben

- Added code into Discussion and Reply components.
- Added skeletons for Class, ReplyEntry, Submission components
- Fixed routing w/ users and questions

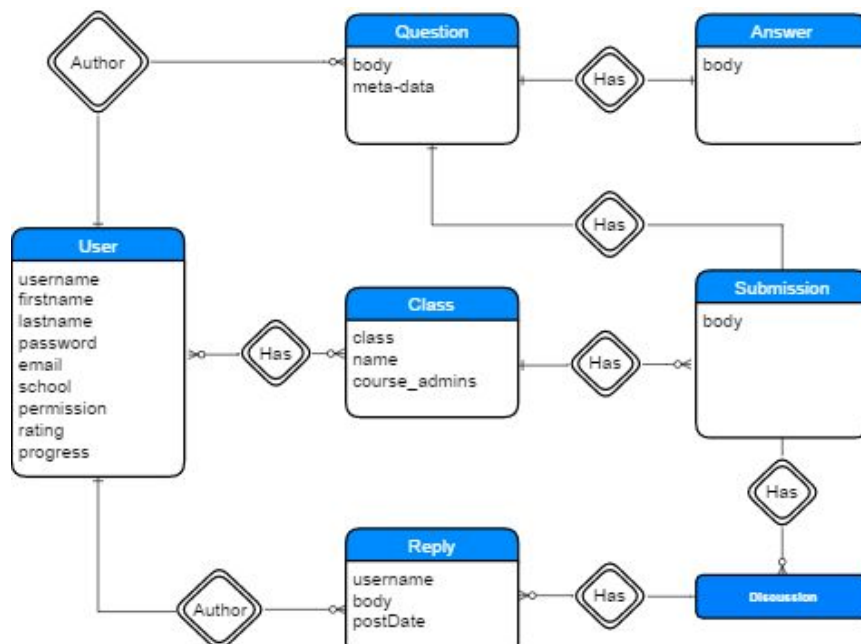
Rafal

- Added code into Userinfo and Userdiscussion components
- Added functions in server.js in order to retrieve data from the database
- Write up of database.js and it's components

Product Feature Cuts:

As of March 13th, 2016, the team has not fully decided if there are to be cuts made to the project. We will be evaluating our product over the remaining weeks to see if there needs to be changes made.

ER Diagram:



Our database currently represents the entities in our diagram the same way. However, we have added some attributes in each entity, which will be described below. We will need to work more on finalizing a complete ER diagram in the future.

User:

- firstname
- lastname
- username
- password
- email
- school
- permission
- rating
- progress
- grade
- GPA
- major
- feed

Class:

- classname
- submission
- users

Question:

- id
- question (contents)
- answer

Discussion:

- author
- title
- contents
- replies

Replies:

- id
- author
- postdate

Submission

- id
- questions
- discussions