Senior Capstone Project: Design Document

Undefined {?} Parameter

Melynda Lindhorst, Blake Bartlett, Joseph Lee, Braeden Bodily

Executive Summary

Our goal is to create a quiz environment that is robust enough for college-level studying, but remains simplistic and intuitive to maintain a larger, more casual audience. This environment will be built as a mobile-friendly website to promote ease-of-use. Quiz questions will be generated and maintained by the users to ensure the most relevant information is kept at the forefront for consumption. Gameplay elements will be introduced to create incentive for returning users.

The focus for this application is primarily for college students but considered to be anyone with an affinity for quizzes or a student at any level of education. The entire success of this project rests in the hands of the users. They will create, moderate, and ultimately decide the fate of quiz questions. Power like that must be approached as a hierarchical tiered system: the more you contribute, the more influence and power you have. By giving this power to the users, it allows for more personalized quiz generation with relevant and up-to-date questions.

Having the users as such an integral part of the functionality has been proven to work well in forums and sites like Wikipedia. User-priority is something other quiz systems do not offer. It provides a dynamic functionality that allows the program to scale to suit any user's needs. It also promotes cooperation. When people build something together, it gives them a motive to continue to work. We hope to mimic that comradery and team work in our application.

Background and Technical Requirements

There is a large offering of quiz-related applications in the world. The necessity of our application lies in its user-oriented design. Other quiz applications can offer you a bulk of questions with varying degrees of quality. Sometimes you can modify your search to get slightly more specific results. The purpose of these applications is not to help the user find questions to increase their learning experience about a subject; instead it's about memorizing facts, selling question databases, general knowledge, or competition between friends. While these are all fine qualities, they alone will not help anyone truly learn a subject.

To facilitate problem-solving and active learning, our application will provide multiple question formats. Essay, mathematical, true-false, multiple choice, mix-and-match—these are some of the different question formats that will be offered to users to create more complex and dynamic quizzes. Implementing these formats will require some research into natural language processing to decipher what the users' input expresses.

Quiz generation will also require research into machine learning or, at the very least, some statistical analysis. A user will input different tags for a quiz generator to use to pick suitable questions for the quiz. To determine if a question can be added to a quiz, multiple characteristics of that question must be taken into account: user rating, tags, age of the question, number of quiz appearances, difficulty, and question-type. If quiz generation fails to assist or entertain a user, they can build their own quizzes using previously created questions.

Quiz generation also relies on whether you belong to a group or not. Different college and university courses can sign up to create quiz study-guides for tests. Questions from different classes at different schools with similar scopes can share quiz questions and are more likely to have their questions show up in randomly generated quizzes from similar courses.

In sharing those questions, competition is generated between the similar classes. This competitive spirit is perpetuated and encouraged throughout the application. Users can challenge friends, classmates, rival schools, or complete strangers. By challenging and winning against opponents, you gain titles and badges to denote prestige in addition to reward points. Challenges can be done live (competing against the opponent at the same time) or on a turn-by-turn basis which would allow both opponents to play together but not require them to be online simultaneously. Titles and badges will also be given out as uncompetitive challenges are met.

Participating in challenges, as well as general use of the application, will gradually produce statistics for a user and the user base. These statistics will be displayed on a user's homepage as charts and graphs, comparing them to themselves and others. The feature offers a baseline for users who wish to improve their performance and view their progress.

To facilitate both web and mobile interfaces, we will be designing the system in a Model-View-Controller pattern. The back end and data (model) layer will be separated from the interface, and all communication from the front end to back end will have to pass through the controller. We will be looking at the framework Dropwizard, which is designed to be friendly in both web and android platforms. To design the front end we will be researching into Angular JavaScript, or JQuery. If time permits we will design a mobile application to serve as an interface for mobile platforms, but if necessary it will be designed to be accessible through a browser on mobile devices.

Requirements

Section 1: System Architecture

1.0 User Interface

This is the user's interaction with the system. In order for QuizZing to provide competitive services, the interface needs to be intuitive and easy to use. To accommodate this, it will be designed to allow features to be used with minimal clicks and screens. For example, the quiz creating will have an "Auto Generate" button that can be clicked without selecting anything, thus allowing quizzes to be created with ease. The interface will be designed using HTML and CSS, supported with JavaScript.

1.1 Network Communication

This component receives information from web browsers and return HTML responses for the browser to display to the user. To facilitate easily implemented and robust communication, QuizZing will be built on the Dropwizard framework. Dropwizard uses built in frameworks like Jetty, and will provide the functionality for processing HTML requests, and allow us to dynamically parse the data and display it in predefined HTML views.

1.2 Question and Quiz Interaction

This component is responsible for allowing communication between QuizZing members and groups. It also provides functionality for the users to create, edit, and take quizzes. It will make use of categories, and tags to allow users to define specific quizzes to meet their individual needs. Through the use of ratings and flags, the users of QuizZing have the power to maintain the content of their individual groups, allowing scalability and targeted content. This will provide competitive play and cooperative question and quiz creation between QuizZing users.

1.3 Quizzing Analytic Services

This will be responsible for tracking user interaction with QuizZing to allow the system to provide needed content to the user. An example of this is if a user is consistently getting specific questions wrong in a quiz, the system will recognize and provide the user with more questions matching those categories. This will assist the learning process that will make QuizZing a tool instead of a game. This will enable smart quiz generation, and allow us to provide statistical analysis, such as bar graphs showing progress, for a user or their group.

1.4 Database

This component will store all of our data in a logical way. There will be multiple tables hosting users, groups, questions, and quizzes. The tables will be constructed to promote the use of our tag and rating systems for quiz generation. The different question types will also be stored in a way that suits them best.

Section 2: Personnel

2.0 Undefined Parameter Team

- **Blake Bartlett** – Blake's focus will be dedicated to the database. He will be responsible for constructing the database in an intuitive way and creating the necessary queries QuizZing needs to retrieve information.
- **Braeden Bodily** – Braeden will be focusing on the User Interface of the system. He will be working to design and implement the HTML front end of the system. He will implement the JavaScript necessary to allow communication from the view to the controller.
- **Joseph Lee** – Joseph will primarily be working on the communication between the views and the QuizZing management systems. He will pass formatted information between the two endpoints using HTTP methods with assistance from the Dropwizard framework.
- **Melynda Lindhorst** – Melynda's primary task will be to construct the QuizZing management system. She will take the information received from the database or views, format it, and send it to the correct recipient function.

Section 3: System Features

3.0 Rank 1 – Our system cannot function without these features.

- **Web User Interface:** Create the web pages, widgets, and services the user will be interacting with.
    - Dependencies: Dropwizard server, HTML Views
    - Co-dependencies: SQL Database
    - Component: User Interface
- **Database Configuration:** Create the database, tables, relations, and queries to store and retrieve information.
    - Dependencies: SQL Database
    - Co-dependencies: User Interface
    - Component: Database

- **Multiple Choice Question Creation:** Create the UI components, logic, and queries to allow users to create and store multiple choice questions.
    - Dependencies: User Interface, Database
    - Co-dependencies: Quiz Interaction
    - Component: Question Interaction
- **Quiz Interaction:** Create the UI components, logic, and queries to allow users to take a quiz and receive a score.
    - Dependencies: User Interface, Database
    - Co-dependencies: Question Interaction
    - Component: Quiz Interaction

3.1 Rank 2 – Our system will look like a finished product by implementing these features.

- **Additional Question Type Creation:** Create more question-types (essay, mathematical, true-false, multiple choice, mix-and-match), UI components, logic, and queries to allow users to create different types of questions.
    - Dependencies: User Interface, Database
    - Co-dependencies: Question Creation
    - Component: Question Interaction
- **Quiz Creation:** Create the UI components, logic, and queries to allow users to create and save quizzes.
    - Dependencies: User Interface, Database
    - Co-dependencies: Quiz Interaction
    - Component: Quiz Interaction
- **Register Users:** Create the UI components, logic, and queries to allow users to register for QuizZing as well as save their content.
    - Dependencies: User Interface, Database
    - Co-dependencies: Security
    - Component: Services
- **Security:** Create logic and alter queries to make sure that all data is secure.
    - Dependencies: Database, Web Server
    - Co-dependencies: None
    - Component: Services
- **User Interaction:** Create UI components, logic, and queries to allow communication between users in the form of friend/group/organization requests and small messages.
    - Dependencies: Register Users
    - Co-dependencies: None
    - Component: Services

- **Tag System:** Create UI components, logic, and queries to allow users to create new tags or use existing tags to catalogue their quizzes, questions, groups, and organizations.
    - Dependencies: Quiz and Question Creation
    - Co-dependencies: Analytics
    - Component: Quiz and Question Interaction, Services
- **Analytics:** Create UI components, logic, and queries to allow visual representations of a user's quiz scores to be displayed on their user page and after a quiz is completed.
    - Dependencies: Member Interaction, Question and Quiz Creation
    - Co-dependencies: None
    - Component: Services

3.2 Rank 3 – Our system feel polished and professional as well as interactive and fun if these features are implemented.

- **Android User Interface:** Using Dropwizard allows for all the internal logic to remain the same. New views need to be created to funnel the existing logic and information through to display.
    - Dependencies: Dropwizard Server
    - Co-dependencies: None
    - Component: User Interface
- **Group/Organization Interaction:** Create UI components, logic, and queries to allow for the creation of groups/organization and communication between these groups/organizations and other users.
    - Dependencies: Register Users
    - Co-dependencies: Group Quiz Generation
    - Component: Services
- **Group Quiz Generation:** Create UI components, logic, and queries for groups/organizations to collaboratively make quizzes with its userbase.
    - Dependencies: Quiz Generation
    - Co-dependencies: Group Interaction
    - Component: Question and Quiz Interaction
- **Competitive Play:** Create UI components, logic, and queries to allow for users, groups, or organizations to compete in quiz matches with time restrictions. Badges and titles will be the rewards.
    - Dependencies: Member Interaction
    - Co-dependencies: Titles/Badges
    - Component: Services
- **Titles/Badges:** Create UI components, logic, and queries to allow titles and badges to be awarded and displayed to users, groups, and organizations.

- Dependencies: Quiz and Question Interaction
- Co-dependencies: Competitive Play
- Component: Services

Tools & Techniques

1.0 Agile vs. Traditional

The software development process that we think will work best for our product is Scrum. We are all familiar with this methodology already and our versioning, bug tracking, and project management software will facilitate this. This agile process will help us identify bugs or address user testing concerns in a timely manner while still allowing for normal implementation workflow to continue.

2.0 Versioning and Bug Tracking

For our versioning and tracking we will be using Github. Github provides a great interface for git versioning, as well as a task management system. At the end of a sprint, we will branch the code, and use that for testing, this allows us to continue implementation while other members may still be testing their specific projects. By sourcing all these important aspects of development from a single entity, we can track progress and maintain team cooperation in a more organized and complete fashion. This will make it easy to allow our team to work in tandem without conflicting with other team mates.

3.0 Testing

We will be doing testing in two different ways. First, we will use unit tests to test the back end of the software. By creating unit tests as we go, it will provide a way for us to regression test. Second, we will beta test our software through user interactions. To encourage users to test we will create sample quizzes targeted at specific classes. Already a math teacher has agreed to offer her students benefits for trying out the system. We will also create a series of quizzes for our operating systems class. We will encourage students in our class to utilize, add to, and improve these quizzes in preparation for finals. To allow for easy feedback, we will provide a link on the website that a user can report bugs.

4.0 Documentation

There will be multiple levels of documentation involved with our website. Github will be responsible for tracking our progress and documenting statistics. In addition, documentation of our code, team member performance, and Quizzing tutorials will be written by the team. Any documents that will be read by users will be subject to a heavy review and testing process to ensure that they are clear, understandable, and informative.

5.0 Team Communication and Meetings

Due to the nature of Scrum, meetings will be scheduled for every other day to check team progress and encourage a steady work pace. In-person meetings will be held at least twice a week and during the beginning and end of sprints. For other meetings, we will do as many meetings as is possible in person, and if we are not able to meet in person, we will do so over Google Hangouts. The importance of sprint planning and review will be emphasized, as that will dictate the flow of development and influence our end product.

Timeline

	Blake	Braeden	Joe	Melynda
01/12 - 01/18	**User:** user tables, queries	**User:** pages, sessions	**User:** user obj, user actions	**User:** user obj translation
01/19 - 01/25	**Security:** db security	**Security:** UI for authentication	**Security:** authentication logic	**Security:** authentication logic
01/26 – 02/01	**Orgs/Groups:** tables, queries	**Orgs/Groups:** pages	**Orgs/Groups:** org/group obj	**Orgs/Groups:** org/group actions
02/02 - 02/08	**Questions:** new question tables	**Questions:** UI for new questions	**Questions:** new question translation	**Questions:** new question translation
02/09 - 02/15	**Questions:** new question queries	**Questions:** UI for new questions	**Questions:** new question translation	**Questions:** new question translation
02/16 - 02/22	**Quizzes:** implement new queries in code	**Quizzes:** refactor UI for new questions	**Quizzes:** logic for multiple question types	**Quizzes:** logic for multiple question types
02/23 - 03/01	**Tags:** tables, queries	**Tags:** tag UI	**Tags:** search/filtering logic	**Tags:** search/filtering logic
03/02 – 03/08	**Analytics:** scoring tables, queries	**Analytics:** graphing utilities and UI	**Analytics:** statistical logic, passing data	**Analytics:** statistical logic, format data
03/09 - 03/15	**Competition:** tables, queries, relations	**Competition:** competition UI components	**Competition:** active participant data	**Competition:** competition logic
03/16 - 03/22	*Spring Break: Padding Time (Titles/Badges)*	*Spring Break: Padding Time (Titles/Badges)*	*Spring Break: Padding Time (Group Quizzes)*	*Spring Break: Padding Time (Group Quizzes)*
03/23 - 03/29	**Titles/Badges & Group Quizzes**	**Titles/Badges & Group Quizzes**	**Titles/Badges & Group Quizzes**	**Titles/Badges & Group Quizzes**
03/30 - 04/05	**Finish loose ends.**	**Finish loose ends.**	**Finish loose ends.**	**Finish loose ends.**

Appendix

Section 1: Use Case Diagrams

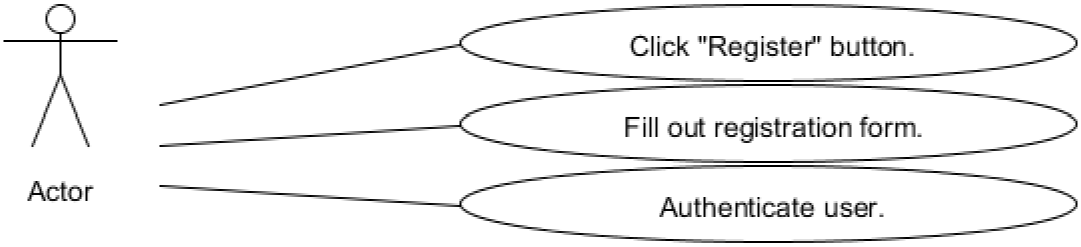
1.0 Register for the Website

Description: A new user to QuizZing will need to register for the website, allowing access to the full range of features (see diagram below).

1. Through the home page, the user will select the option to register.
2. They will fill out the necessary information on the registration form, including name, user name, password, email address, and more.
3. Once the registration form is filled out, the user clicks on submit. If any errors are found they will be highlighted in red, and an error message appears notifying the user to correct them.
4. Once all information is entered correctly, the user will be authenticated and directed to their profile page.

Related UI: Section 2.0 Home Page and Section 2.2 Registration Page

Figure 1A: Registration



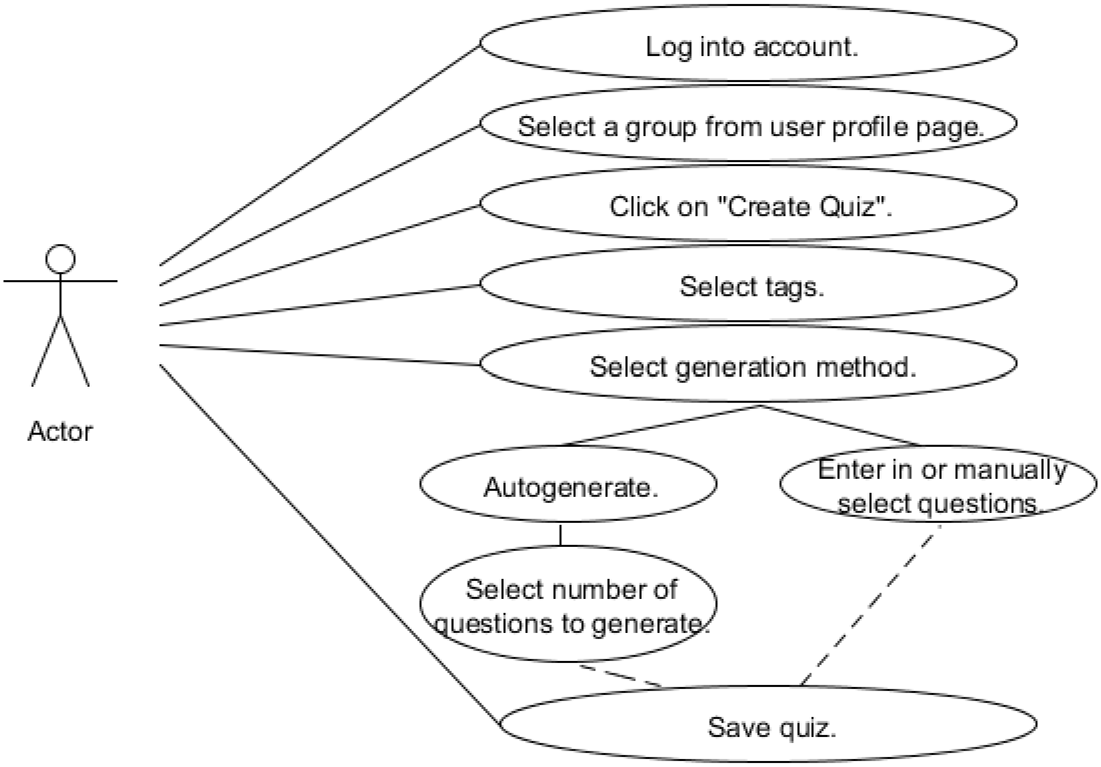
1.1 Create a Quiz

Description: Through their registered group or class the user will create a quiz by using tags and ratings, or use the feature to auto generate a quiz (see diagram below).

5. Through the home page, the user will log into their account.
6. The user will select a class or group from his list of registered courses.
7. From the group page, the user will select the option to create a quiz.
8. The user will customize the type of quiz they wish to create by adding tags, which can function as categories of questions, ratings, or difficulty.
9. They can either click on generate to allow the system to generate a quiz, or select the questions from a list, which is sorted according to selected tags.
10. If they select to generate a quiz, the user will indicate the number of questions to create.
11. The quiz can now be saved for later use by selecting the option to save.

Related UI: Section 2.5 Quiz/Question Generation Page

Figure 1B: Quiz creation


Log into account.
Select a group from user profile page.
Click on "Create Quiz".
Select tags.
Select generation method.
Actor
Autogenerate.
Enter in or manually select questions.
Select number of questions to generate.
Save quiz.

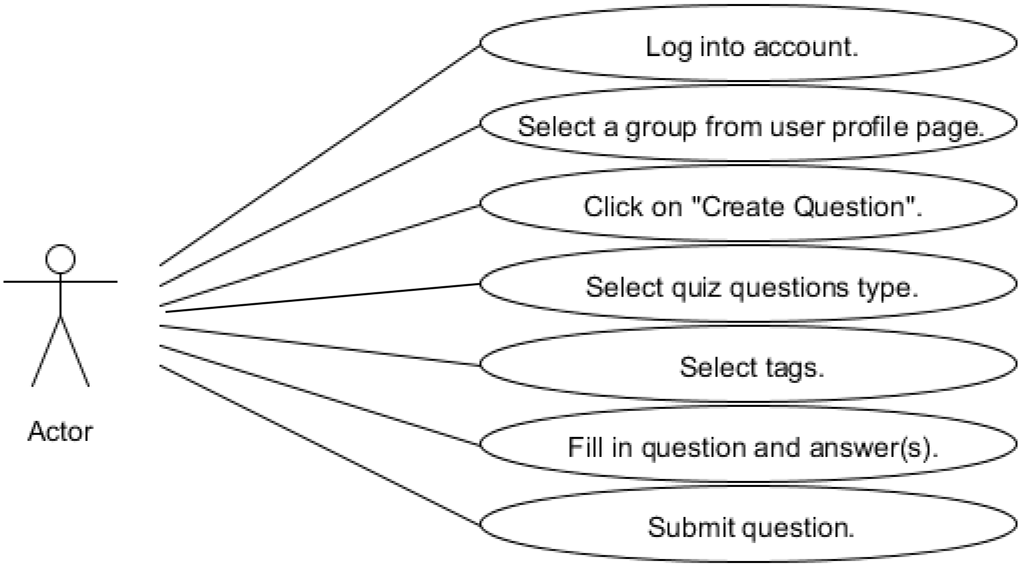
1.2 Add a Quiz Question

Description: The user can create their own custom questions by accessing the create a question section through the corresponding group or class (see diagram below).

1. Through the home page, the user will log into their account.
2. The user will select a class or group from his list of registered courses.
3. By clicking on the "create a question" link the user will be directed to a page to create a question.
4. The type of question is selected, for example: multiple choice, fill in the blank, or matching.
5. Tags are selected to add categories, difficulty, or specify a specific purpose of the question, such as midterm prep.
6. The user fills in the question and then enters the answer or answers depending on the question type.
7. The question is submitted by clicking on the submit button.

Related UI: Section 2.5 Quiz/Question Generation Page

Figure 1C: Question creation


Log into account.
Select a group from user profile page.
Click on "Create Question".
Select quiz questions type.
Select tags.
Fill in question and answer(s).
Submit question.
Actor

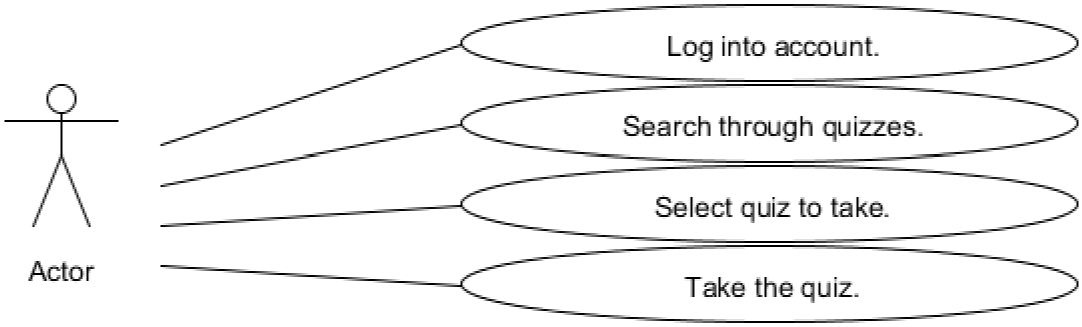
1.3 Take a Quiz

Description: The user can take a quiz or answer single questions at any time by performing a general or detailed search on subject matter (see diagram below).

1. Through the home page, the user will log into their account.
2. At any point, the user will be able to use the ubiquitous search bar at the top of the page to search through quizzes or questions.
3. The search may also be refined using our tag system to increase search granularity.
4. The user selects a question or quiz from the search results.
5. The user answers the question or questions and performs any other necessary actions (ie., flagging a question for review (see

Related UI: Section 2.6 Search Page and Section 2.4 Quiz Page

Figure 1D: Taking a quiz use case



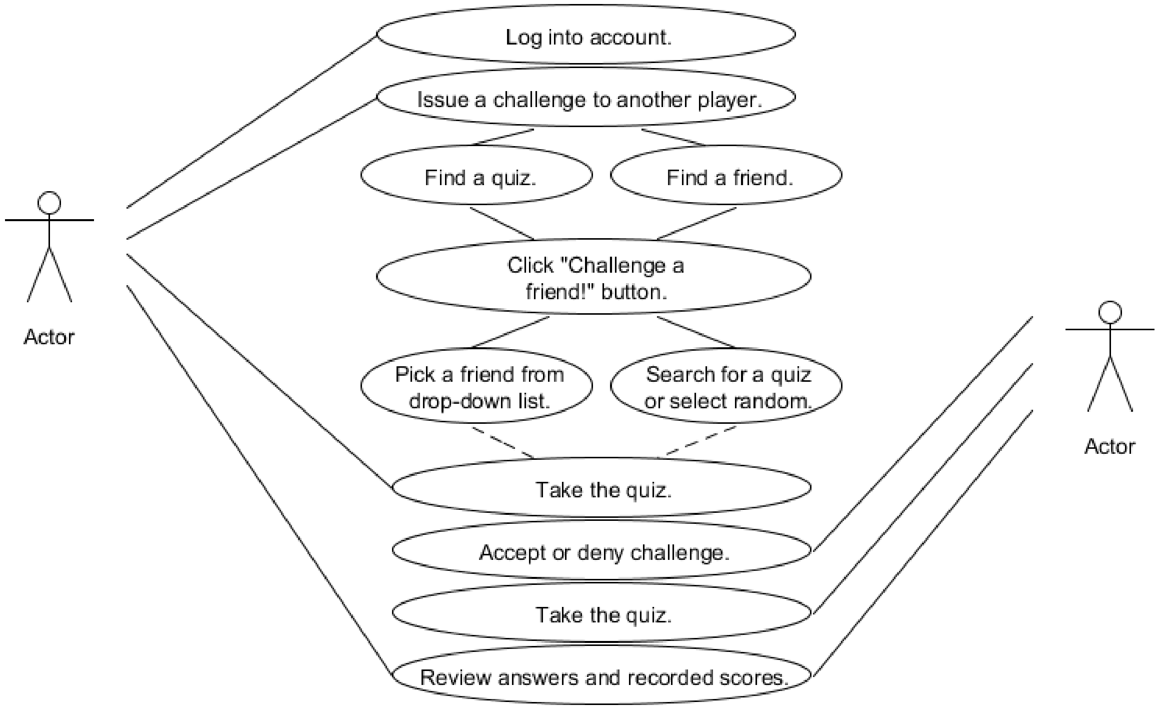
1.4 Challenge a User

Description: Users can access the gameplay aspect of QuizZing by challenging other users compete on a quiz. Winner is determined by time and accuracy. (see diagram below).

1. Through the home page, the user will log into their account.
2. On his profile page a user can select the option to challenge another player.
3. The user will find a competitor by searching among groups, user names, region, etc.
4. The user will find a quiz to challenge his friend to. It is also possible to challenge a player directly from a chosen quiz.
5. The user will be able to immediately take the quiz, even if his friend is offline.
6. Once the friend logs in, they will receive a notification on their profile page of the challenge.
7. If the user accepts the challenge, they will immediately be able to take the quiz, even if the challenger is offline.
8. Once both parties have completed the quiz the results will be displayed to both parties, in the form of a notification. They will see this immediately or next time they log in.

Related UI: Section 2.1 Profile Page and Section 2.4 Quiz Page

Figure 1E: Challenge a user


Log into account.
Issue a challenge to another player.
Find a quiz.
Find a friend.
Click "Challenge a friend!" button.
Pick a friend from drop-down list.
Search for a quiz or select random.
Take the quiz.
Accept or deny challenge.
Take the quiz.
Review answers and recorded scores.
Actor
Actor

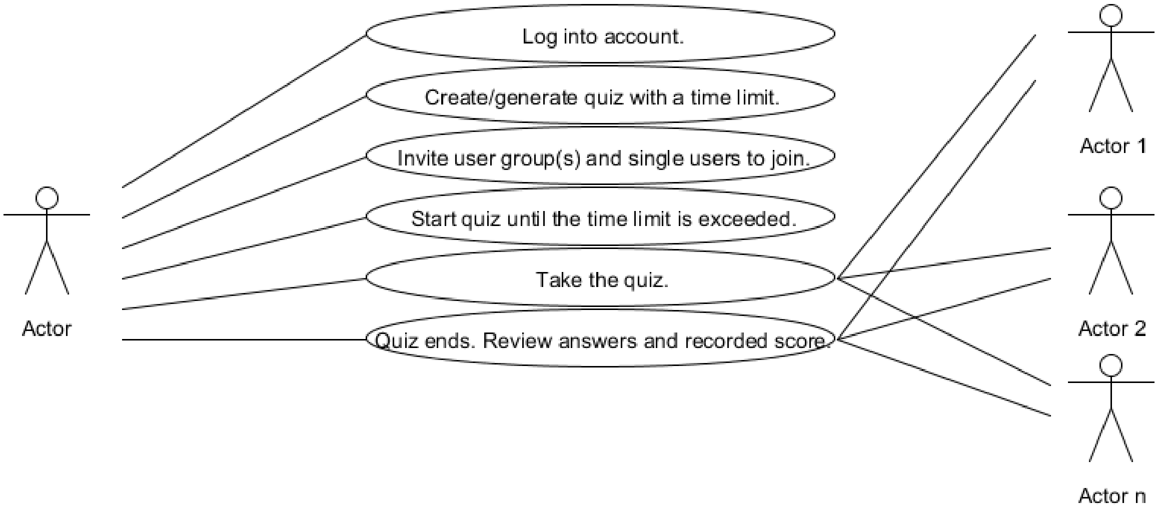
1.5 Challenge a Group

Description: Entire groups can participate in the gameplay aspects of QuizZing by using the challenge a group option. (see diagram below).

1. Through the home page, the user will log into their account.
2. The user creates a quiz for the group challenge (see Section 1.2).
3. User groups can be issued a challenge to take the quiz by selecting challenge a group through that group's page.
4. The challenger immediately takes the quiz, even if all other challengers are offline.
5. If the time limit is reached, the quiz is ended and the user sees their score.
6. When other users log into their profile page, they receive a notification of the challenge.
7. If the user accepts, they immediately take the quiz, until time limit is reached, and view their score.
8. Once a user has taken the quiz, they can view their rankings.
9. Each time a new user takes the quiz all previous users are sent notifications of the change in the rankings.

Related UI: Section 2.3 Group/Class Page

Figure 1F: Challenge a group


Log into account.
Create/generate quiz with a time limit.
Invite user group(s) and single users to join.
Start quiz until the time limit is exceeded.
Take the quiz.
Quiz ends. Review answers and recorded score.
Actor
Actor 1
Actor 2
Actor n

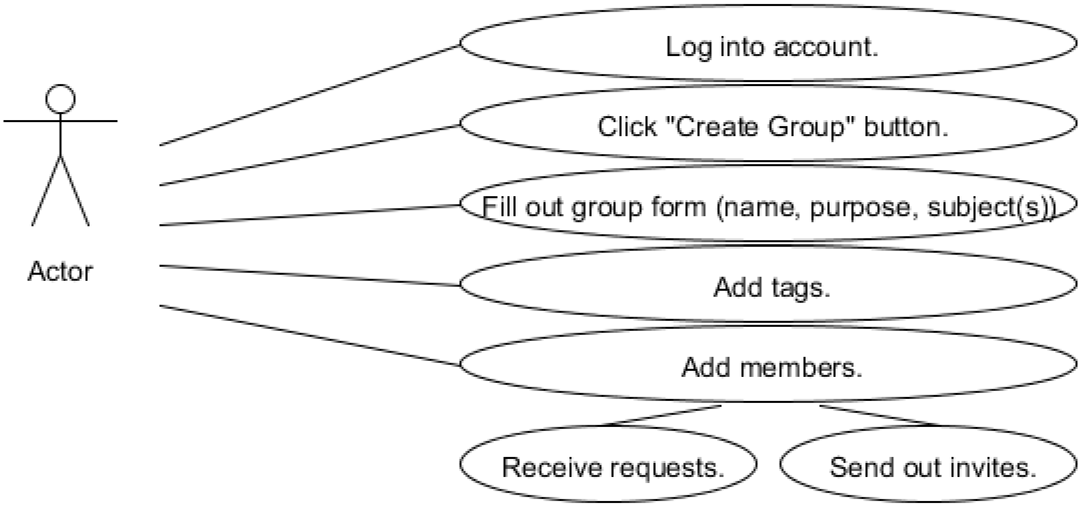
1.6 Create a Group

Description: Users can create groups for school or extracurricular purposes. They can invite other users to join or receive requests from other users to join. (see diagram below).

1. Through the home page, the user will log into their account.
2. The user will navigate to the "Create Group" button, located under the ubiquitous drop-down navigation menu.
3. The user will be prompted to fill out information for that group that includes the name, description/purpose of group, and subjects the group will cover.
4. The user will also have to add tags for that group so that quizzes and questions can be linked back to that group.
5. The user who is now the group owner will send out invitations to individual users to join the group.
6. Alternatively, users can ask to join the group by clicking the "Ask to Join" button on the group's profile page.

Related UI: Section 2.3 Group/Class Page

Figure 1G: Create a group



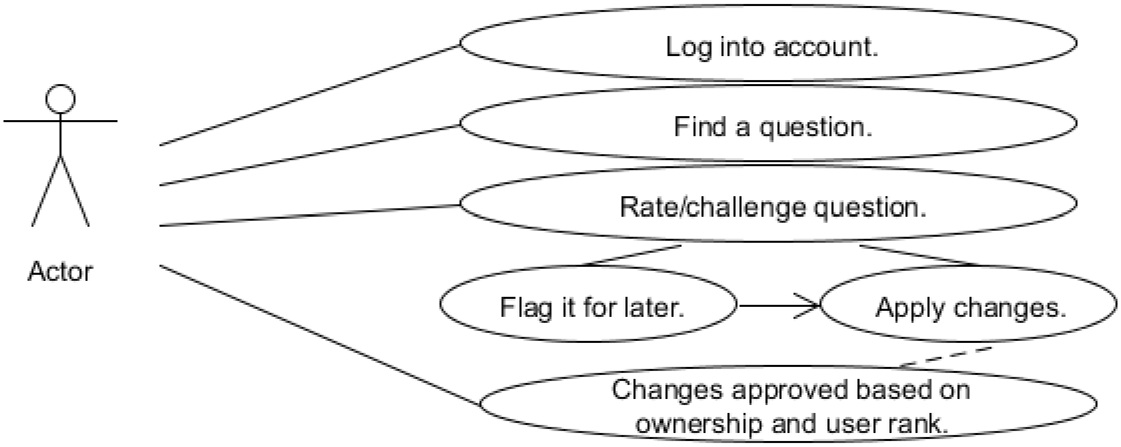
1.7 Rate or Challenge a Question

Description: Entire groups can participate in the gameplay aspects of QuizZing by using the challenge a group option. (see diagram below).

1. Through the home page, the user will log into their account.
2. Users will find a question either by searching or casually encountering questions during quizzes.
3. The user may want to rate the question or challenge the question for displaying incorrect information.
4. Users can choose to flag the question to rate or challenge at another time or provide feedback right then.
5. If the user chooses to flag the question, a notification will appear in their inbox and the question will be added to their flagged queue.

Related UI: Section 2.1 Profile Page and Section 2.4 Quiz Page

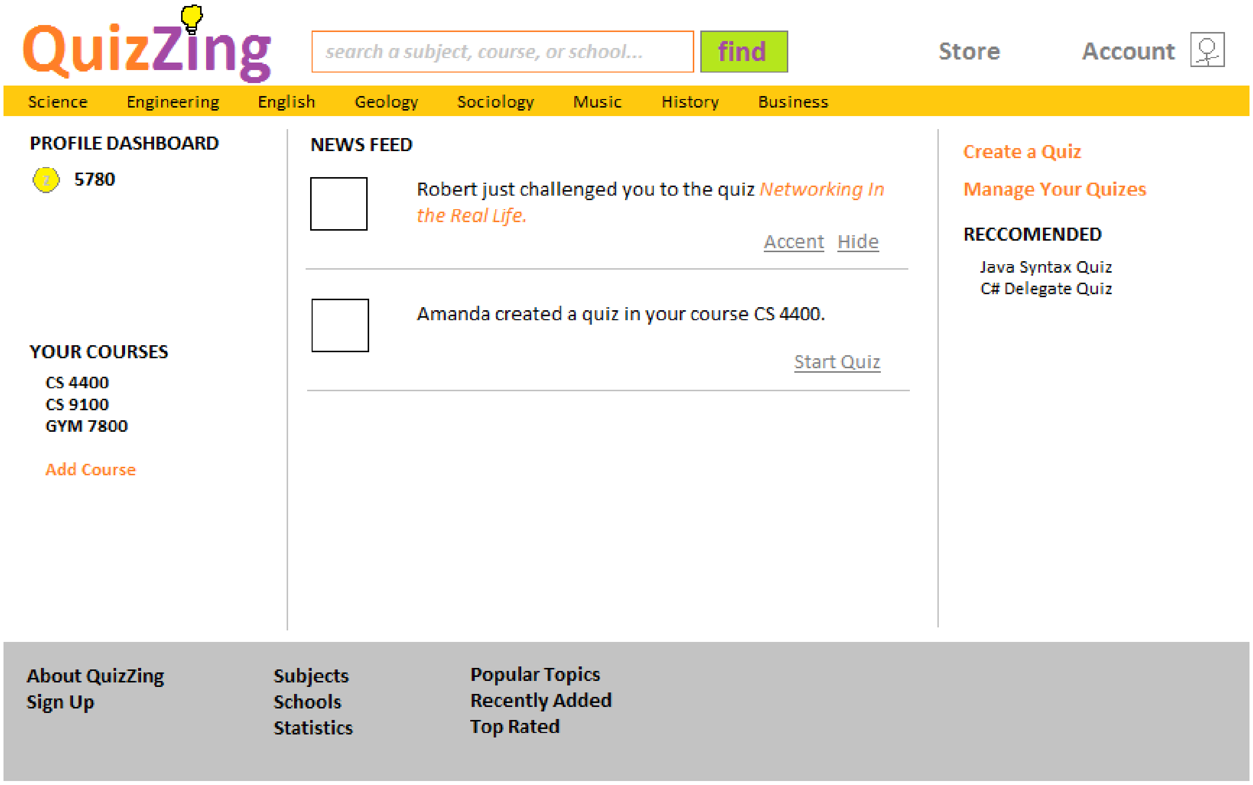
Figure 1H: Rate/challenge a question



Section 2: UI Sketches

2.0 Home

This is a sketch of the main page that every user will see when they first reach our website. It will provide a news feed, a link to register or login, and other side bars to be determined in the future.



PROFILE DASHBOARD

5780

YOUR COURSES

CS 4400
CS 9100
GYM 7800

Add Course

NEWS FEED

Robert just challenged you to the quiz *Networking In the Real Life.*

Accent Hide

Amanda created a quiz in your course CS 4400.

Start Quiz

Create a Quiz

Manage Your Quizes

RECCOMENDED

Java Syntax Quiz
C# Delegate Quiz

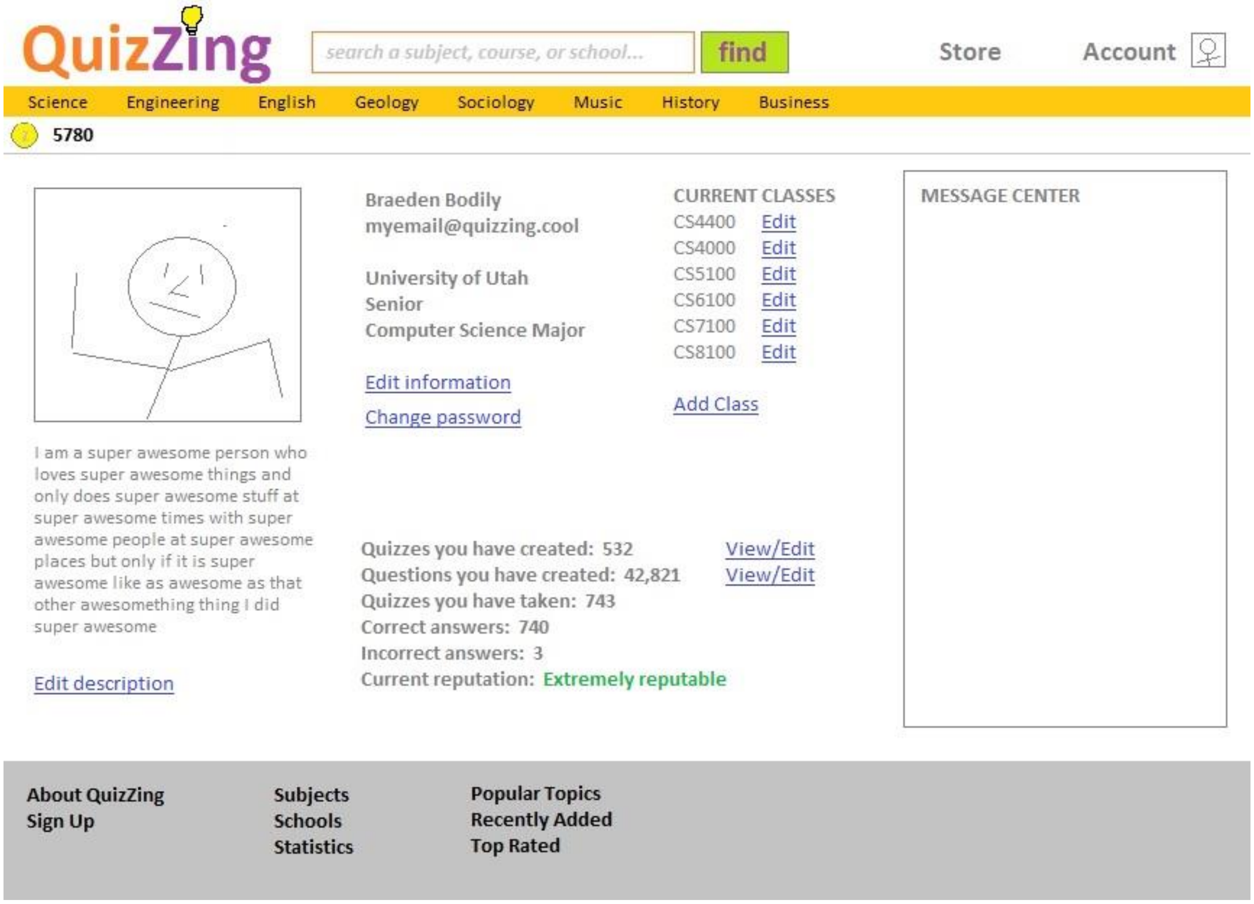
About QuizZing
Sign Up

Subjects
Schools
Statistics

Popular Topics
Recently Added
Top Rated

2.1 Profile

The profile page is what a user will see after they successfully register or log into their account. They can use it to view or modify their personal information, or class and registration.



Braeden Bodily
myemail@quizzing.cool

University of Utah
Senior
Computer Science Major

CURRENT CLASSES	
CS4400	Edit
CS4000	Edit
CS5100	Edit
CS6100	Edit
CS7100	Edit
CS8100	Edit

I am a super awesome person who loves super awesome things and only does super awesome stuff at super awesome times with super awesome people at super awesome places but only if it is super awesome like as awesome as that other awesomething thing I did super awesome

Quizzes you have created: 532 View/Edit
Questions you have created: 42,821 View/Edit
Quizzes you have taken: 743
Correct answers: 740
Incorrect answers: 3
Current reputation: Extremely reputable

2.2 Registration

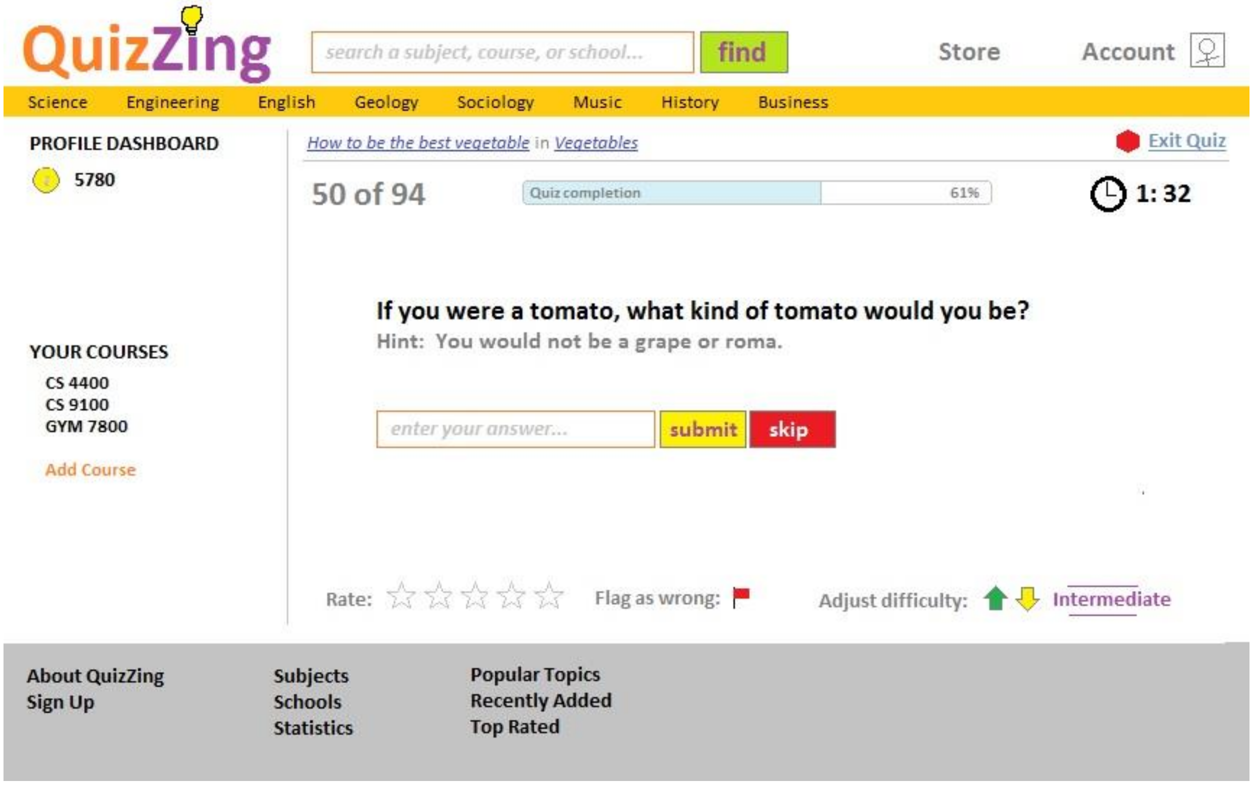
The Registration page shows what the user will see when they are registering for the webpage.

2.3 Group/Organization Page

The Group or Class page shows details and data about a particular group. It provides links to register for the class, view members, generate a quiz, or even try a quiz as a guest.

2.4 Quiz

The Quiz page is what the user will see while they are taking a quiz. It will provide the ability for a user to go forwards and backwards through the quiz, rate a question, flag a question, or even a link to challenge the question immediately.



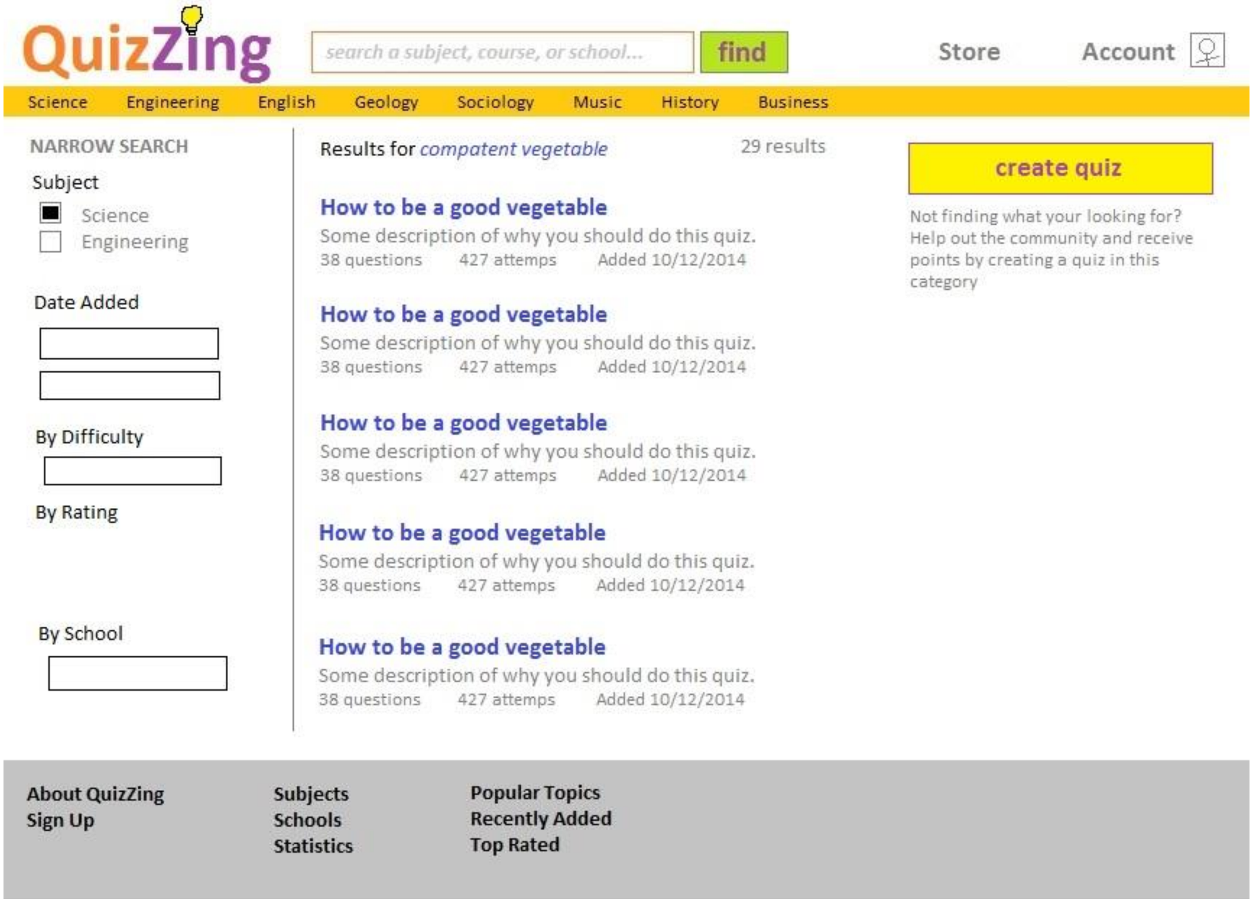
If you were a tomato, what kind of tomato would you be?
Hint: You would not be a grape or roma.

2.5 Quiz Generation

The Quiz Generation page is the page a user will access to create a quiz. It will be accessed through a Class or Group page. It provides the ability to toggle categories, difficulty, and ratings. Also, it provides a link for a user to auto-generate the quiz (based on their selections) or hand pick the questions themselves.

2.6 Search

A search bar will be at the top of most pages to allow the user to search for quizzes or questions. More detailed searches can also be performed. This will allow the user to get to what they need quicker.



Results for *compatent vegetable* 29 results

How to be a good vegetable
Some description of why you should do this quiz.
38 questions 427 attemps Added 10/12/2014

How to be a good vegetable
Some description of why you should do this quiz.
38 questions 427 attemps Added 10/12/2014

How to be a good vegetable
Some description of why you should do this quiz.
38 questions 427 attemps Added 10/12/2014

How to be a good vegetable
Some description of why you should do this quiz.
38 questions 427 attemps Added 10/12/2014

How to be a good vegetable
Some description of why you should do this quiz.
38 questions 427 attemps Added 10/12/2014

Not finding what your looking for? Help out the community and receive points by creating a quiz in this category