

Snap-2-Ask

Design
Document

Raz Friman

Raymond Martin

Vipul Kohli

Roman Stolyarov

Elena Villamil

Table of Contents

EXECUTIVE SUMMARY	2
USE CASE DIAGRAM	3
FEATURE SUMMARY	4
TUTORS AND STUDENTS	4
TUTORS.....	5
STUDENTS.....	5
SOFTWARE ARCHITECTURE.....	6
DATABASE MODEL	9
USER INTERFACE	10
IOS MOBILE UI	10
WEB UI.....	13
TESTING	17
USABILITY TESTING	17
REVIEW OF OUR TEST TEAM	17
OUR PERFORMANCE AS A TEST TEAM	17
TEAM REFLECTION.....	19
TECHNICAL CHALLENGES.....	19
EXTRACURRICULAR CONCEPTS	19
WHAT WE WOULD DO DIFFERENTLY	19
SUMMARY AND FUTURE FEATURES.....	21
DATA DICTIONARY APPENDIX	22
users	22
answers.....	23
questions	23
categories	24
subcategories.....	25
validationQuestions	26
verified_categories	27



EXECUTIVE SUMMARY

Snap-2-Ask is the world's first image-based, crowdsourced question-and-answer service. Our mission is to make it as easy as possible for students to get quick and accurate solutions to their study questions by motivating tutors around the world to answer them. Due to the ease of our image-based framework, Snap-2-Ask benefits both student and tutor substantially.

Using Snap-2-Ask, students can simply "snap" a photo of their homework problem and send it to our database for public viewing. This saves them tremendous time because it eliminates the need to type the text of their question to an online forum. Students can instead quickly crowd source problems that might include substantial text or complex symbols, figures, graphs, or equations. Additionally, students do not waste time searching a database of millions of questions looking for ones similar to theirs.

Tutors, on the other hand, are given the opportunity to put their knowledge and skills to work and be compensated for their effort. They can earn money simply by spending time at their own computer.

Snap-2-Ask allows students to get help quickly use the tools they are most familiar with and tutors to get paid by working from the comfort of their own homes.

Snap-2-Ask is composed of a team of 5 programmers.

Raz Friman – Website and mobile developer

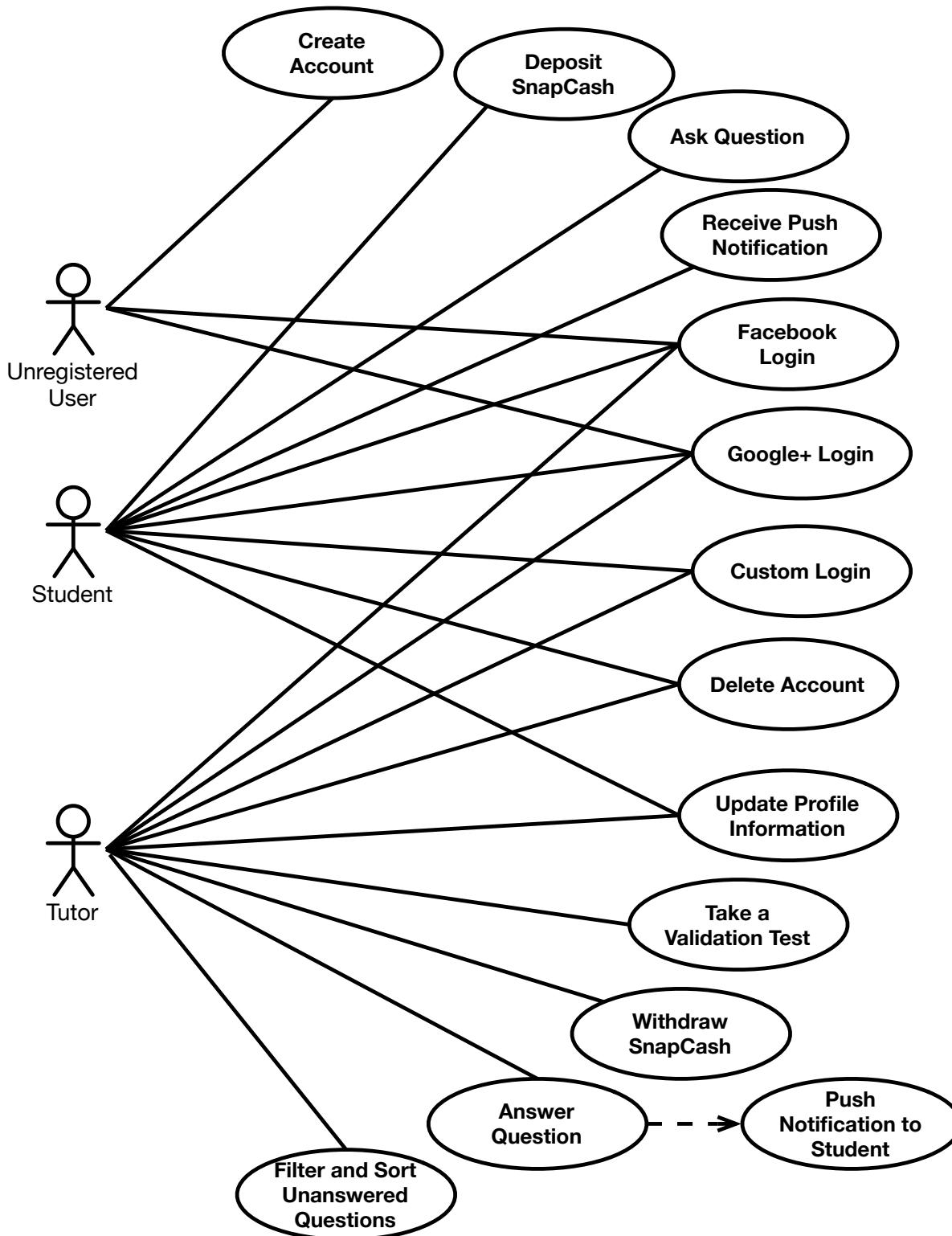
Raymond Martin – Website designer

Vipul Kohli – Website developer

Roman Stolyarov – Website and backend API developer

Elena Villamil – Database manager

USE CASE DIAGRAM





FEATURE SUMMARY

TUTORS AND STUDENTS

- Create an Account: Users will be able to create an account using their email address as a unique identifier. When creating the account users will need to input their first and last name, email address, password.
- Custom Login: Once the account is created, the user enters their email address and password to sign in.
- Facebook Login: Users can sign in via their Facebook account instead of having to register for a new account.
- Google+ Login: Users can sign in via their Google+ account instead of having to register for a new account.
- Filtering Unanswered Questions: Tutors will be able to sort and filter unanswered questions. Tutors can filter questions by category, by all of the their preferred categories, or by searching for specific keywords within a question.
- Filtering Previous Answers: Tutors will be able to sort all of their previous answers by category, pay amount, date, or rating. This can help tutors look back and see how happy students have been with their answers and where they can improve.
- Edit Profile: Using the web application, users can change their profile information. This is as simple as clicking on the "Profile" link , clicking on "Update Profile", and changing any desired field.
- Deposit SnapCash: Students can deposit SnapCash into their account in order to ask questions. Each time a student asks a question, SnapCash is deducted from their account.



- Withdraw SnapCash: Tutors can withdraw their earned SnapCash. Tutors earn SnapCash by answered questions.
- Delete account: Accounts can be deleted from our database at any time. However, unclaimed SnapCash will be lost if the account is deleted. Therefore, Snap-2-Ask displays a warning before deleting the account, and if the user proceeds to delete the account, all information associated with the account is deleted.

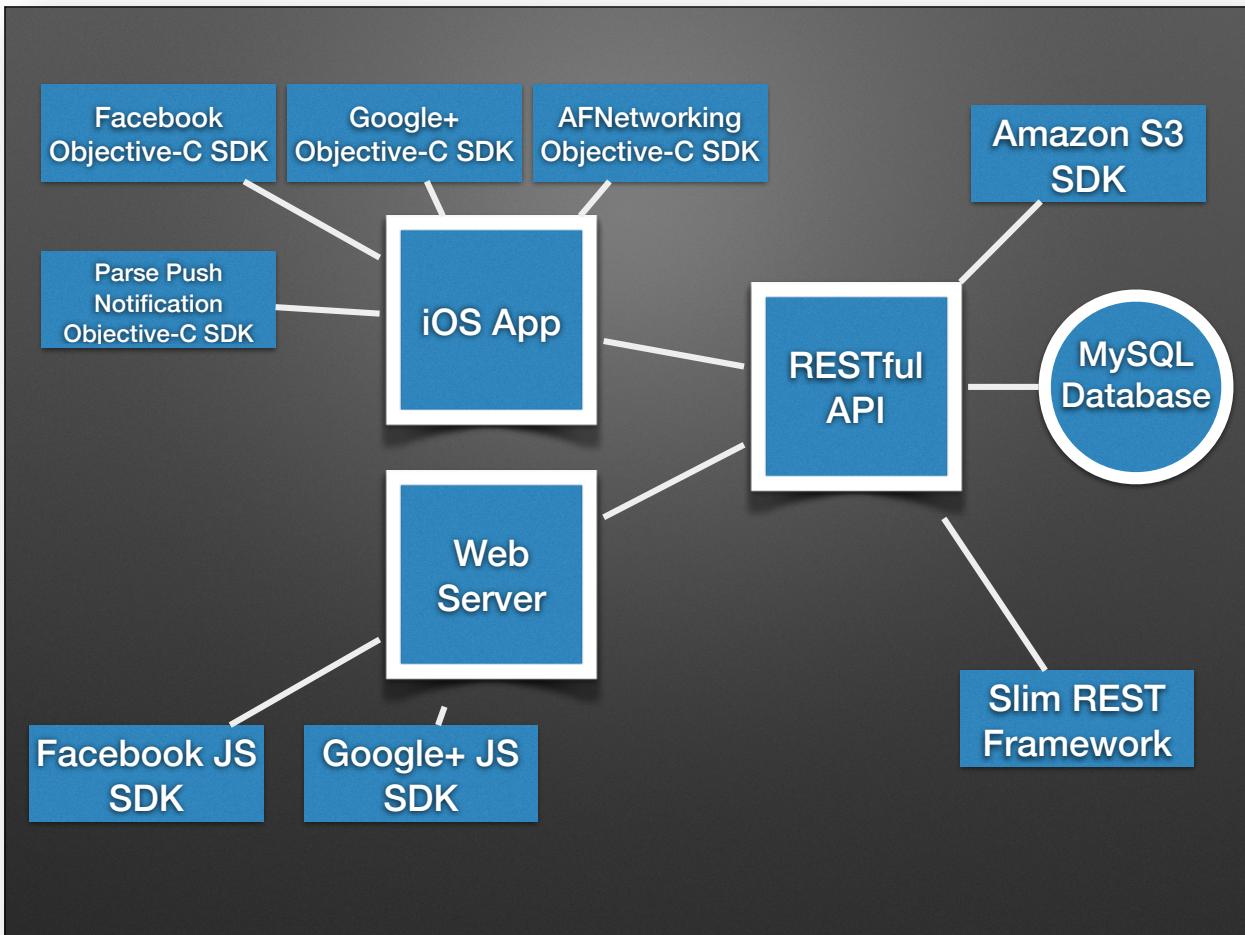
TUTORS

- Answer Questions: Tutors will go to Snap-2-Ask website to browse unanswered questions. Tutors simply need to select a question they can answer, type out the answer, and submit the answer to the student.
- Validation: Tutors can pass a validation test for each category they would like to be verified in. This rewards the tutor with extra SnapCash for every answer in that category.

STUDENTS

- Ask Questions: Students simply need to snap a picture of the question they are struggling with, enter the category the question belongs to, optionally enter a description of the question, and click submit.
- Push Notifications: Students will have the option to receive push notifications from Snap-2-Ask every time one of their questions gets answered.

SOFTWARE ARCHITECTURE



Our architecture consists of 4 main parts.

MySQL database (Backend Layer)

- Serves as the backend for the whole architecture
- Responsible for all persistent data

RESTful API (Business Logic Layer)

- Written in PHP
- Contains all of the business logic
 - This is the only place where MySQL statements are prepared and executed
- SLIM REST framework
 - Maps PHP functions to a specific route for the REST API
- Amazon S3 SDK



- This SDK is used in order to upload question images to an Amazon S3 Bucket which hosts all of the images used by Snap-2-Ask

Web server (Presentation Layer)

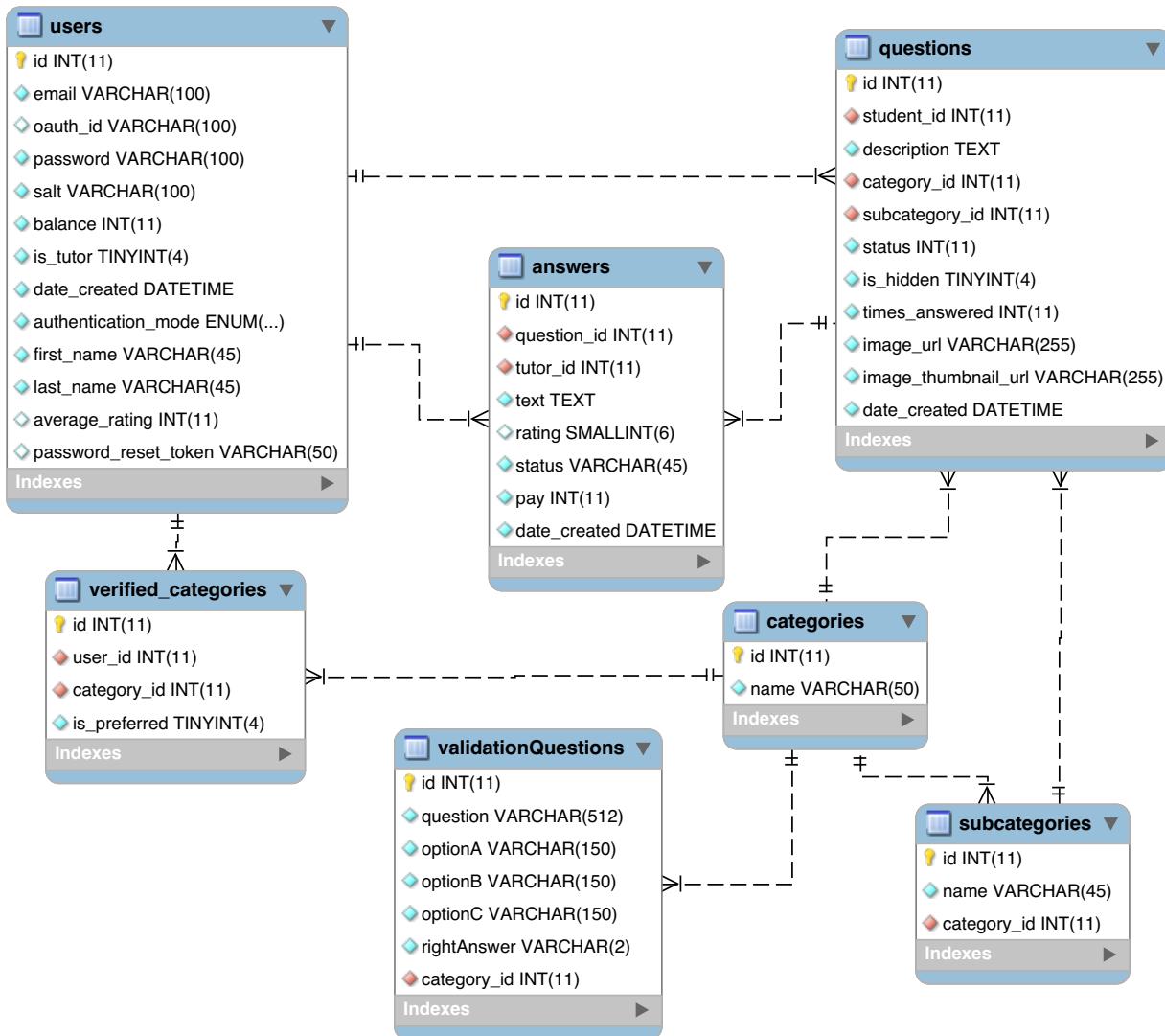
- This is the front-end that tutors use while answering questions.
- jQuery
 - Small, feature rich library that extends the general abilities of Javascript.
 - Helps significantly with DOM traversal
 - <http://jquery.com>
- jQuery-Validation
 - Library that can dynamically validate any HTML form
 - Used everywhere that user input is required in the website
 - Helps ensure the input is valid
 - Provides real-time warnings and messages to the user depending on whether their input is valid or not for a specific form
 - <http://jqueryvalidation.org>
- jPages
 - Library that can paginate a list of HTML elements dynamically
 - Used when browsing questions to ensure the user is not overloaded with 100's of questions being loaded on the page at the same time
 - <http://luis-almeida.github.io/jPages/>
- jQuery-Tinysort
 - Library that can dynamically sort a list of HTML elements based on any specified attribute or property
 - Used on the "My Answers" page to help sort and filter answers based on the user specified criteria
 - <http://tinysort.sjeiti.com>
- Lightbox2
 - Library that can overlay images on top of the current page
 - This is used anywhere that question images are displayed
 - Users can click on an image to display a larger, more detailed version of that picture
 - <http://lokeshdhakar.com/projects/lightbox2/>
- Facebook JS SDK
 - Library used to authenticate user via Facebook's authentication system
 - <https://developers.facebook.com/docs/javascript/gettingstarted/>
- Google+ JS SDK
 - Library used to authenticate user via Google+'s authentication system
 - <https://developers.google.com/+/web/api/javascript>
- Parse REST API
 - Not specifically a library, but the web server calls Parse's REST API in order to request a Push Notification to be sent to the iOS user

iOS application (Presentation Layer)



- front-end layer that students use to ask questions and receive answers.
- AFNetworking
 - Library that helps everything related to networking and networks requests
 - Automatically parses an HTTP Response into JSON
 - Used for all HTTP requests within the application
 - <http://afnetworking.com>
- Facebook Objective-C SDK
 - Library used to authenticate user via Facebook's authentication system
 - <https://developers.facebook.com/docs/ios/>
- Google+ Objective-C SDK
 - Library used to authenticate user via Google+'s authentication system
 - <https://developers.google.com/+/mobile/ios/api/>
- Parse SDK
 - Library used to register each iOS device for Push Notifications
 - <https://parse.com>

DATABASE MODEL



See the Data Dictionary appendix for more details.



USER INTERFACE

IOS MOBILE UI

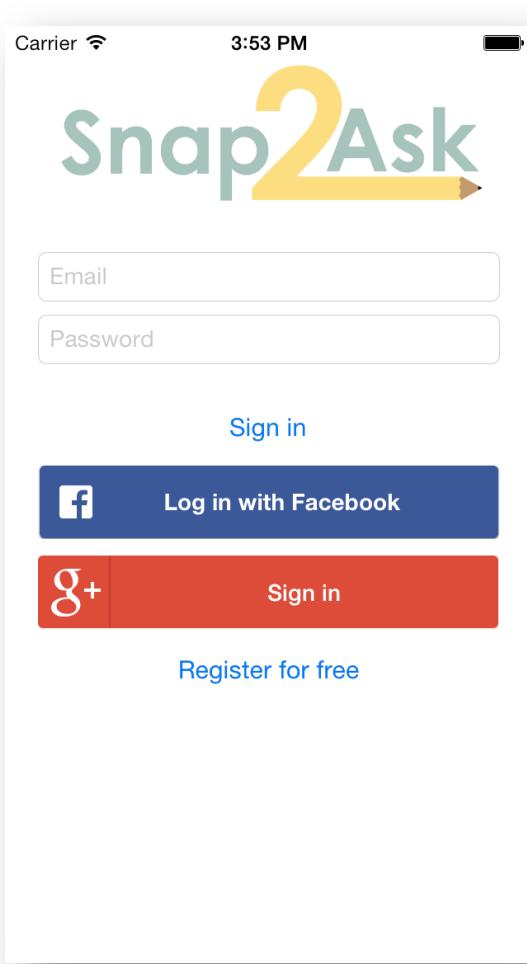


FIGURE 1: IOS HOME PAGE. FROM HERE, USERS CAN EITHER LOGIN WITH AN EXISTING ACCOUNT, OR CREATE A NEW ACCOUNT

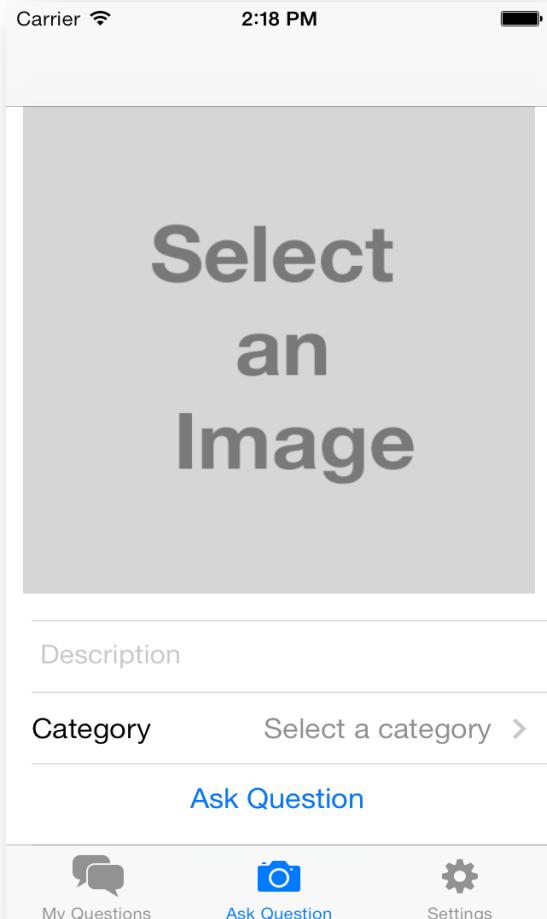


FIGURE 2: THE MAIN PAGE FOR ASKING A QUESTION. HERE, YOU CAN TAKE A PHOTO OF YOUR QUESTION, ADD A DESCRIPTION, AND SELECT A CATEGORY FOR YOUR QUESTION. THEN, SIMPLY CLICK THE "ASK QUESTION" BUTTON TO SUBMIT THE QUESTION TO THE SNAP-2-ASK TUTORS

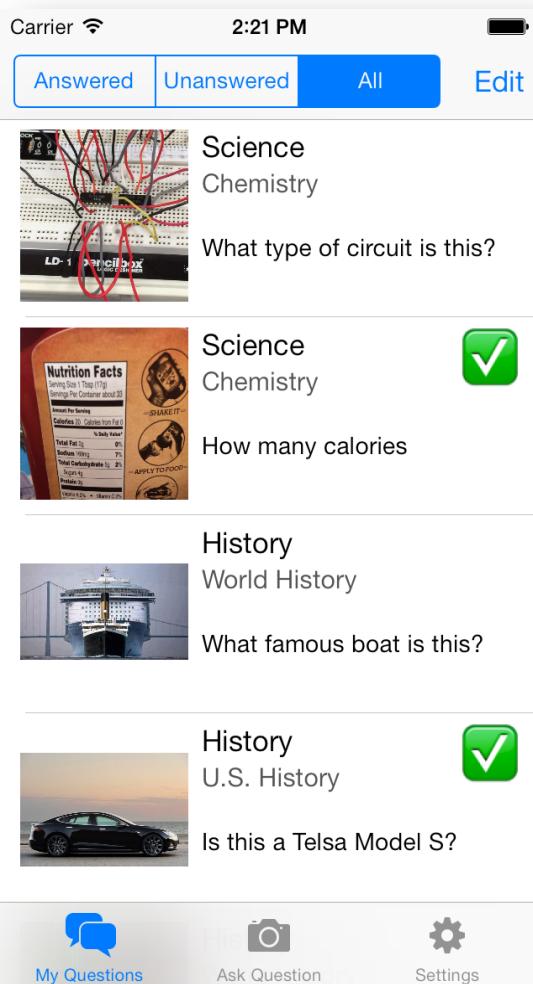


FIGURE 1: VIEW ALL YOUR QUESTIONS. THE GREEN CHECK MARK NEXT TO A QUESTIONS INDICATES THAT IT HAS BEEN ANSWERED. ADDITIONALLY, YOU CAN FILTER OUT ANSWERED OR UNANSWERED QUESTIONS IN ADDITION TO VIEWING ALL OF YOUR QUESTIONS

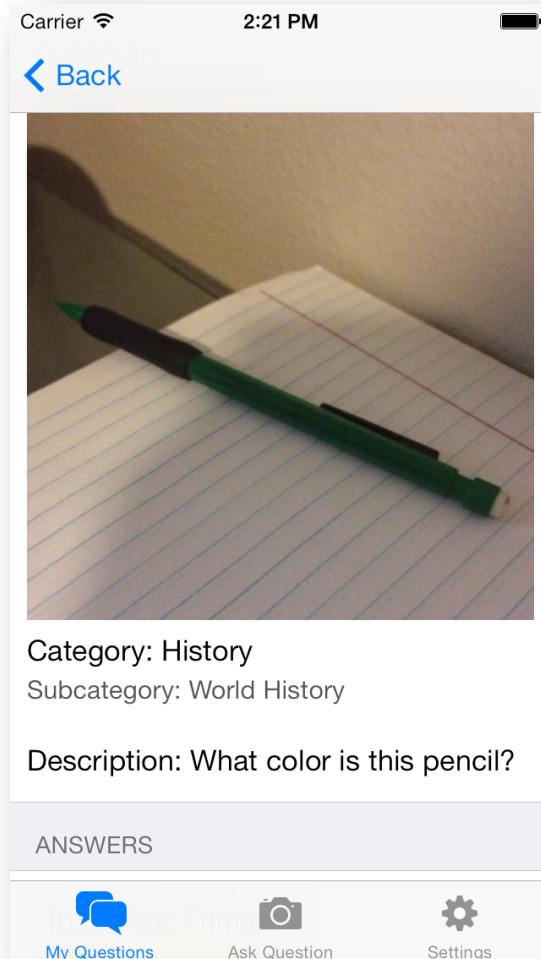


FIGURE 4: A DETAILED VIEW OF A QUESTION. THIS INCLUDES SPECIFIC DETAILS OF THE QUESTION. ADDITIONALLY, THE ANSWER TO THE QUESTION WILL APPEAR HERE, IF IT IS AVAILABLE

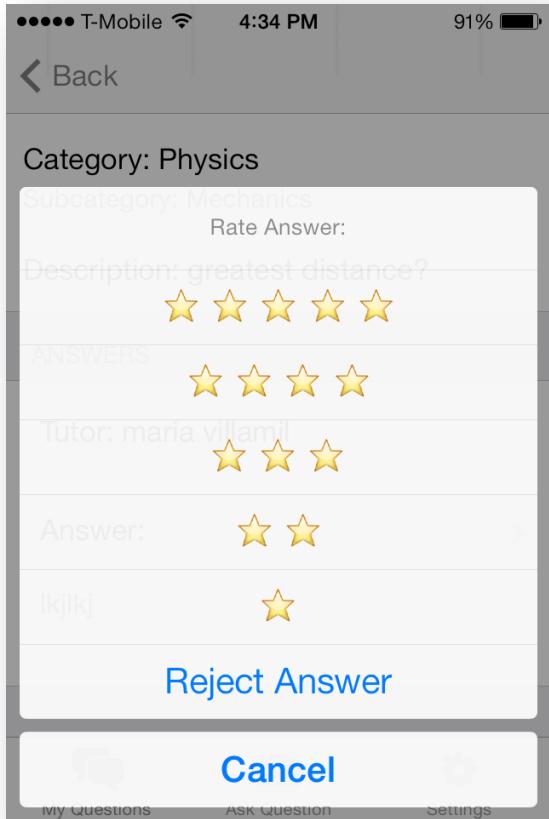


FIGURE 5 – THE UI FOR RATING THE ANSWER TO QUESTION THAT A STUDENT POSTED

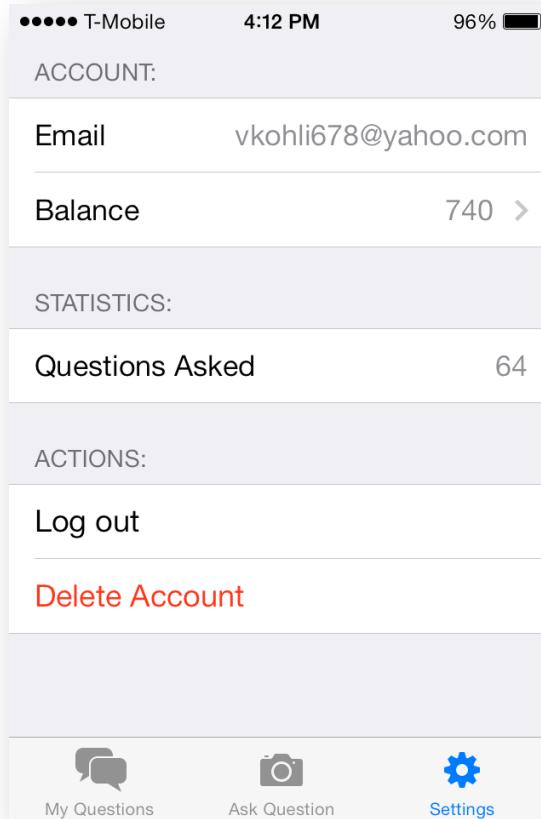


FIGURE 6 – THE SETTINGS PAGE WHERE YOU CAN VIEW DETAILS ABOUT YOUR ACCOUNT, AS WELL AS LOG OUT AND DELETE YOUR ACCOUNT



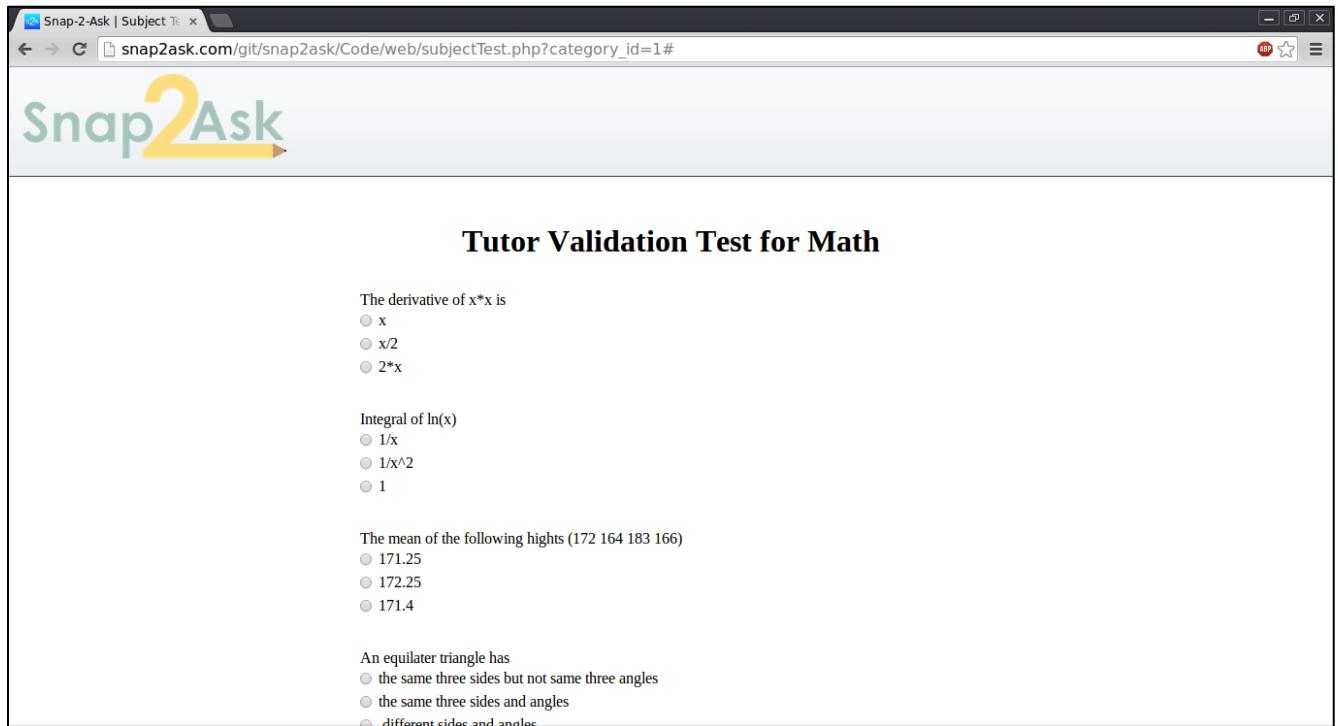
WEB UI

The screenshot shows the homepage of the Snap2Ask website. At the top right, there is a login form with fields for email (rfriman@smu.edu) and password, and buttons for "Log in", "Sign in", and "Forgot Password". Below the login, there are social media links for Google+ and Facebook. The main content area has three bullet points: "Answer picture-based questions.", "Get paid for answering questions.", and "Certify yourself in your favorite subjects for extra bonuses.". A horizontal line separates this from another section: "Are you a student? [Click Here](#) for more information.". To the right, there is a "Create an Account" form with fields for First Name, Last Name, Email, Password, and Confirm Password, followed by a "Sign Up" button. At the bottom of the page, a footer bar contains the text "Snap-2-Ask 2013 | About Us".

FIGURE 7 - THE SNAP-2-ASK HOMEPAGE OF THE WEBSITE

The screenshot shows a test choice page. The browser address bar shows "snap2ask.com/git/snap2ask/Code/web/testChoice.php". The main content area displays a message: "Our records indicate you are a first time tutor. In order to start answering questions and get paid, you have to pass a test. You can choose to take it now or later." Below this is a dropdown menu labeled "Select Category" with a downward arrow. Underneath the dropdown are two buttons: "Take Now" and "Take Later". At the bottom of the page, a footer bar contains the text "Snap-2-Ask 2013 | About Us".

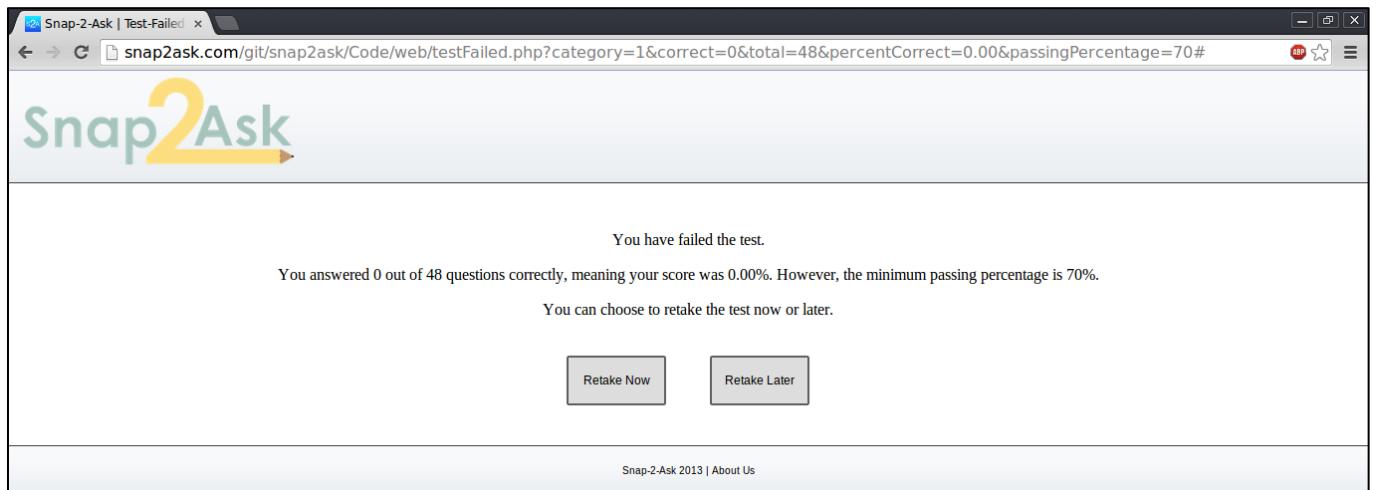
FIGURE 8 - AFTER CREATING AN ACCOUNT FOR THE FIRST TIME, YOU ARE PRESENTED WITH AN OPTION TO VERIFY YOURSELF IN A SPECIFIC CATEGORY.



The screenshot shows a web browser window with the title "Snap-2-Ask | Subject Test". The URL is "snap2ask.com/git/snap2ask/Code/web/subjectTest.php?category_id=1#". The page content is titled "Tutor Validation Test for Math". It contains the following questions:

- The derivative of $x*x$ is
 - x
 - x/2
 - 2*x
- Integral of $\ln(x)$
 - 1/x
 - 1/ x^2
 - 1
- The mean of the following heights (172 164 183 166)
 - 171.25
 - 172.25
 - 171.4
- An equilateral triangle has
 - the same three sides but not same three angles
 - the same three sides and angles
 - different sides and angles

FIGURE 9 - AN EXAMPLE VALIDATION TEST WITH SAMPLE QUESTIONS FOR THE MATH CATEGORY



The screenshot shows a web browser window with the title "Snap-2-Ask | Test Failed". The URL is "snap2ask.com/git/snap2ask/Code/web/testFailed.php?category=1&correct=0&total=48&percentCorrect=0.00&passingPercentage=70#". The page content is titled "Snap2Ask". It displays the message "You have failed the test. You answered 0 out of 48 questions correctly, meaning your score was 0.00%. However, the minimum passing percentage is 70%. You can choose to retake the test now or later." Below this message are two buttons: "Retake Now" and "Retake Later". At the bottom of the page, there is a footer with the text "Snap-2-Ask 2013 | About Us".

FIGURE 10 - AFTER FAILING A TEST, THE USER IS PRESENTED WITH THE ABILITY TO RETAKE A TEST OR TAKE THE VALIDATION TEST LATER

The screenshot shows the homepage after logging in. At the top, there's a navigation bar with links for 'Browse', 'Categories', and 'Familiar'. Below this, a search bar and a 'Log-out' button are visible. The main content area is titled 'UNANSWERED QUESTIONS' and features a grid of question cards. The cards include various subjects like Science, Chemistry, SAT Math, Geography, Physics Mechanics, and History. Each card displays a thumbnail image, the question title, and a date. Some cards also show small diagrams or graphs related to the questions.

FIGURE 11 - THE HOMEPAGE AFTER LOGGING IN. THIS PAGE SHOWS ALL OF THE UNANSWERED QUESTIONS AVAILABLE FOR TUTORS TO ANSWER. YOU CAN SORT AND FILTER THESE QUESTIONS BY CATEGORIES

The screenshot shows the 'Balance' page. At the top, it displays 'Available SnapCash: 40'. Below this, there's a 'Withdraw Amount' input field and a 'Withdraw SnapCash' button. The page also includes links for 'Browse', 'Balance', 'Profile', and 'My Answers'. At the bottom, there are links for 'Snap-2-Ask 2013 | About Us'.

FIGURE 12 - THE BALANCE PAGE WHERE YOU CAN VIEW YOUR CURRENT SNAPCASH BALANCE. YOU CAN WITHDRAW YOUR SNAPCASH EARNED FROM ANSWERING QUESTIONS HERE

The screenshot shows the 'Profile' page of the Snap2Ask website. At the top, there's a navigation bar with 'Welcome, Raymond' and links for 'Search' and 'Log-out'. On the left, a sidebar lists 'Browse', 'Balance', 'Profile' (which is underlined), and 'My Answers'. The main content area has a title 'YOUR PROFILE' and displays the following information:

- First Name: Raymond
- Last Name: Martin
- Email: a@a.com
- Preferred Categories: Add preferred categories [here](#)
- Certified Categories: (empty)

Below this is a section titled 'GET CERTIFIED' with the text: 'When you pass our the subject tests, you will be paid more for answering questions in that category.' A dropdown menu labeled 'Select Category' is shown.

FIGURE 13 - THE PROFILE PAGE WHERE YOU CAN VIEW ALL OF THE DETAILS OF YOUR ACCOUNT. HERE YOU CAN EDIT YOUR PROFILE INFORMATION OR VALIDATE YOURSELF IN OTHER CATEGORIES

The screenshot shows the 'View Answers' page of the Snap2Ask website. The interface is similar to Figure 13, with a sidebar for 'Browse', 'Balance', 'Profile', and 'My Answers'. The main content area has a title 'MY ANSWERS' and includes a date range filter ('mm/dd/yyyy') and sorting options ('Most Recent First', 'All'). Below this is a table listing answers:

Question	Answer	Rating
which has higher refraction index? 	glass	5

At the bottom, there's a footer with 'Snap-2-Ask 2013 | About Us'.

FIGURE 14 - THE ANSWERS PAGE WHERE YOU CAN VIEW ALL OF YOUR PREVIOUSLY POSTED ANSWERS. YOU CAN SORT AND FILTER THESE ANSWERS IN ORDER TO SEE HOW WELL STUDENTS HAVE RATED YOUR ANSWERS



TESTING

USABILITY TESTING

Other than the Feature Set testing done with our testing team, we also showed the production web site to Dr. Luis Resendis during his office hours and some other personal friends outside of school. These testers gave us valuable input that we, as developers, would not intuitively see as issues to new users because we are fiddling with site on a daily basis. Since we are focused on implementing features to make the site functional in several aspects, we do not always notice when user info is not properly propagated to all pages.

Dr. Resendis gave us a great deal of very informative feedback about our website. For example, Dr. Resendis stated: "You need to subdivide the familiar page with a header because tutors cannot distinguish between familiar categories". We appreciated this feedback and implemented a new feature which addresses this issue immediately afterwards.

Another personal friend, Michael H. had this to say after using Snap-2-Ask: "I think it will be of good use to people who have questions... but from the front page, I don't really get exactly what it does". In response to this, we modified our home page to have a more clear and direct explanation of our product. Since then, every new user that we have talked to has understood the general concept behind our product.

REVIEW OF OUR TEST TEAM

Our testing team was a huge resource to us. They spent quality time finding important bugs that we overlooked while developing Snap-2-Ask. Thankfully, their bug reports on Github were very detailed and helped us pinpoint exactly where the bugs were occurring. We took the time to examine every issue opened on our project's Github page and we were able to solve each issue. Some of their feedback included making certain pages easier to understand for users not very familiar with the website. This makes sense, given that we, as a team, understand our product very well. However, all users need to be able to understand our product, so the testing team's feedback was greatly appreciated.

OUR PERFORMANCE AS A TEST TEAM

Our team split up the software architecture into five different areas.

- Web Authentication/Registration
- Web Usage



- Android Authentication/Registration
- Android Usage
- Android and Web Consistency

We then each undertook our part of the software testing. When we thought we had tested the software area thoroughly and had found most of the bugs, we went on to test other areas of the software for consistent implementation and usage on the mobile app and the web app.

Our bugs were all documented on GitHub with detailed information. This allowed the test team to examine and evaluate their bugs in order to solve them.

Communication was kept open, even after the bug-testing period, in order to ensure full disclosure of bugs.

General feedback was also given in the form of an “issue” on GitHub. Stylistic and more subjective opinions were offered in order to improve user interface. Additionally, vocabulary and design suggestions were also offered through GitHub.

All of the documented bugs were taken into consideration by the test team and improved/fixed if needed. The test team was very quick to resolve bugs and even quicker in responding to questions in feedback. An open channel for communication was kept through GitHub until all bugs for the test team were resolved. This whole process happened very smoothly and efficiently. Having a test team was definitely a win-win situation for both our test team and us.



TEAM REFLECTION

TECHNICAL CHALLENGES

One of the technical challenges we faced was how to store all of the images for the questions posted to Snap-2-Ask. We wanted a fast, efficient, reliable method to store and retrieve these image files. After debating whether to store the images as a BLOB in the database, or on the webserver's local file system, we could not come to a perfect solution. However, later we discovered Amazon's S3 service. This service lets you upload files (photos in our case) to Amazon's servers using a bucket file system. By using Amazon's simple API's, we were able to efficiently and reliably store all of our image files and simply refer to them with a URL on our webserver. Using Amazon's S3 service not only solved our challenge, but also proved to be a great experience for learning how to integrate another company's API's into our own product

EXTRACURRICULAR CONCEPTS

CREATING A REST API

We had to research how to create a REST API in PHP that can be accessed by both the web server and the iOS application. After doing research, we came across the SLIM PHP Framework, which allows you to map PHP functions to a REST API route.

CREATING AN IOS APPLICATION

Raz, our mobile developer researched how to design and develop iOS applications. After reading many articles and browsing the Apple documentation, he was able to learn how to develop an iOS application and created the Snap-2-Ask mobile application.

WHAT WE WOULD DO DIFFERENTLY

One of the hardest aspects of this project was to communicate clearly with the whole team. From the beginning of the project, we created a group text-message using GroupMe. However, communicating between five people via text-messages was



not very effective. We would choose a more effective method of communication to ensure that all of our team members are always on the same page. However, in the end, we all came together and accomplished everything we set out to do.



SUMMARY AND FUTURE FEATURES

Overall, Snap-2-Ask provides a huge set of features to assist both students with asking questions and tutors with answering these questions. However, we still have many more features and ideas in mind for the future that have not been developed yet.

One of the most important goals for the future is to support more mobile platforms. This includes Android and Windows Phone. The reason for this, is because not all students have an iOS device, and therefore cannot ask their questions. By supporting more mobile platforms we will increase the availability of our application to many more users.

Another very important feature, that is not complete, is a real payment system. Currently Snap-2-Ask uses a virtual currency called SnapCash that is earned when answering questions and deducted when asking questions. In the future, we will integrate this system directly with real money. So students will be able to deposit money into their account and earn a specific amount of SnapCash. On the other hand, tutors will be able to withdraw their earned SnapCash back out to real money. This is important because Snap-2-Ask must give real incentives for tutors to answer questions, and money is one of the best incentives available.

DATA DICTIONARY APPENDIX

ORGANIZED BY TABLE

Users

This table contains the user data information for every Snap-2-Ask account created.

FIELD NAME	DESCRIPTION
id	Integer made up of up to 11 numbers. It is the primary key of the table and auto increments whenever a new user is created.
email	String of 100 characters. It is the username the user will use to log in, therefore it is unique for the user.
oauth_id	A unique token used to identify users while using 3 rd party authentication such as Facebook or Google+.
password	String of 100 characters. It is a hashed version of the password the user entered when creating the account.
salt	String of 100 characters. It contains the salt used to hash the user's password. It is also used to verify the hashed password stored in the database when a user logs in.
balance	Integer of up to 11 digits. Contains the amount of SnapCash for the user. This is initially set to 50 to give users a free chance to ask questions.
is_tutor	Tinyint (Boolean). Not every Snap-2-Ask user is a tutor. It is initially set to false, and when the user gets validated it is set to true.
date_created	Date-time value. This holds the date and time when the account was created.
authentication_mode	Enumeration specifying what method of authentication the user's account uses. The enumeration members include: "custom", "facebook" and "google".
first_name	String of up to 45 characters. It contains the user's first name.
last_name	String of up to 45 characters. It contains the user's last name.
average_rating	Integer of up to 11 digits. It contains an average of all the rating the user has been given.
password_reset_token	A unique token that is generated when the user requests to reset their password. This token is emailed to the user to verify the identity of the user when resetting the password.

Indices and Relationships

- Primary Key – id
- Unique Key – email, authentication_mode
 - This set of tuples determines a unique user account

answers

This table contains the information stored for each answer of every question.

FIELD NAME	DESCRIPTION
id	Integer made up of up to 11 numbers. It is the primary key of the table, and auto increments whenever a new answer is posted.
question_id	Integer made up of up to 11 numbers. It is used to uniquely identify the question that the answer is referring to.
tutor_id	Integer made up of up to 11 numbers. It is used to uniquely identify the tutor that posted the answer.
text	Text object. Used to store the actual answer given by the tutor.
rating	Small integer made up to 6 numbers. It is used to hold the rate that the student gives to this answer. This is null until a student rates the answer.
status	String of 45 characters. It is used to indicate if the answer has already been answered or not. Possible values include "Pending", "Answered", and "Rejected"
pay	Integer made up of up to 11 numbers. It is used to represent the amount of SnapCash that the tutor earned by answering the question.
date_created	Date-time value. This holds the date and time when the answer was posted.

Indices and Relationships

- Primary Key – id
- Foreign keys:
 - question_id references a unique question from the question table.
 - tutor_id references a unique tutor from the users table.

questions

This table contains the information stored for every question on Snap-2-Ask. This includes unanswered and answered questions

FIELD NAME	DESCRIPTION
id	Integer made up of up to 11 numbers. It is the primary key of the table,

	and it auto increments whenever a new user is created.
student_id	Integer made up of up to 11 numbers. Used to uniquely identify the student who posted the question.
description	Text object used to hold the description of the picture given by the student.
category_id	Integer made up of up to 11 numbers. Used to uniquely identify the category the question belongs to.
subcategory_id	Integer made up of up to 11 numbers. Used to uniquely identify the subcategory the question belongs to.
status	Integer made up of up to 11 numbers. Used to indicate if the question is unanswered, has been answered but still can be answered, or is has been answered and it doesn't need more answers.
is_hidden	Tinyint made of up to 4 integers. It is used to indicate if the question has gotten a successful answer and it can be mark as answered.
times_answered	Integer made up of up to 11 numbers. Used to indicate how many answers this question has received.
image_url	String of 255 characters. Used to store the URL to reach the picture of the question.
image_thumbnail_url	String of 255 characters. Used to store the URL for a scaled down version of the picture for a question.
date_created	Date-time value. This holds the date and time when the question was posted.

Indices and Relationships

- Primary Key – id
- Unique key: (description, image_url)
- Full-text Index – description
 - A full-text index is used to take advantage of MySQL full-text searching.
 - Allows using complex text searching queries to find questions by their text instead of by a regular MySQL SELECT statement
- Foreign keys:
 - subcategory_id references a unique subcategory in the subcategory tables.
 - categories_id references a unique category in the categories table.
 - user_id references a unique user in the users table.

categories

This table contains the names and id of every category that Snap-2-Ask recognizes.

FIELD NAME	DESCRIPTION
------------	-------------

id	Integer made up of up to 11 numbers. It is the primary key of the table, and it auto increments whenever a new user is created.
name	String of 255 characters. Used to indicate the name of the category

Indices and Relationships

- Primary Key – id
- Unique Key – name
- Full-text Index – name
 - Using a full-text index allows Snap-2-Ask to search for questions by category name

subcategories

This table contains the names and id of every subcategory that Snap-2-Ask recognizes.

FIELD NAME	DESCRIPTION
id	Integer made up of up to 11 numbers. It is the primary key of the table, and it auto increments whenever a new user is created.
name	String of 255 characters. Used to indicated the name of the subcategory
category_id	Integer made up of up to 11 numbers. Used to uniquely identify the category to which this subcategory belongs.

Indices and Relationships

- Primary Key – id
- Unique Key – (name, id)
- Foreign Key – category_id
- Full-text Index – name
 - Using a full-text index allows Snap-2-Ask to search for questions by subcategory name

validationQuestions

This table contains the information for each validation question used to validate the tutors in specific subjects

FIELD NAME	DESCRIPTION
id	Integer made up of up to 11 numbers. It is the primary key of the table, and it auto increments whenever a new user is created.
question	Array of characters made of up to 512 characters. It contains the header of the question.
optionA	Array of characters made of up to 512 characters. It contains the first option to answer the question.
optionB	Array of characters made of up to 512 characters. It contains the second option to answer the question.
optionC	Array of characters made of up to 512 characters. It contains the third option to answer the question.
rightAnswer	Array of characters made of up to 2 characters. It contains the letter of the correct answer to the letter. Thus it can be either A, B or C.
category_id	Integer made up of up to 11 numbers. It used to uniquely identify the category this question is testing.

Indices and Relationships

- Primary Key – id
- Foreign key: category_id references a unique category in the categories table.
- Unique Key – (question, optionA, optionB, optionC, rightAnswer, category_id)
 - This ensures that the same question cannot appear more than once in a validation test

verified_categories

This is a bridge table the combines the categories that a tutor has been validated in.

FIELD NAME	DESCRIPTION
id	integer made of up to 11 number. It is the primary key. Each tuple has a unique number assigned to it.
user_id	integer of up to 11 digits. It contains the id of the tutor we want to refer. It is used when a join with the user table is need. It cannot be null.
category_id	integer made of up to 11 digit. It contains the id of the category this tutor is verified for. It is used when joining with the categories table. It cannot be null.

Indices and Relationships

- Primary Key – id
- Foreign key:
 - user_id references a unique user in the users table.
 - Category_id references a unique category in the categories table.
- Unique Key – user_id, category_id