Snap-2-Ask

Final Technical Report

Raz Friman

Raymond Martin

Vipul Kohli

Roman Stolyarov

Elena Villamil

Prepared for:

Mark Fontenot and Chris Raley

CSE3330/3345 Professors

December 8, 2013

Snap2Ask

Table of Contents

EXECUTIVE SUMMARY 2
USE CASE DIAGRAM 3
1.0 FEATURE SUMMARY 4
1.1 BOTH TUTORS AND STUDENTS 4
1.2 TUTORS ONLY 5
1.3 STUDENTS ONLY 5
2.0 SOFTWARE ARCHITECTURE 6
3.0 DATABASE MODEL 9
4.0 USER INTERFACES 11
4.1 iOS 7 MOBILE APP FOR STUDENTS 11
4.2 WEB SITE UI FOR TUTORS 15
5.0 TESTING 19
5.1 USABILITY TESTING 19
5.2 REVIEW OF OUR TEST TEAM 20
5.3 OUR PERFORMANCE AS A TEST TEAM 20
6.0 TEAM REFLECTION 21
6.1 TECHNICAL CHALLENGES 21
6.2 EXTRACURRICULAR CONCEPTS 21
6.3 WHAT WE WOULD DO DIFFERENTLY 22
7.0 SUMMARY AND FUTURE FEATURES 22

DATA DICTIONARY APPENDIX 24
users 24
answers 25
questions 25
categories 26
subcategories 27
validationQuestions 28
verified_categories 29

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

Main Contribution: Build Manager & Efficiency Researcher mobile & web

Raymond Martin – Website designer

Main Contribution: Tutor Profile & Accounts and Web site stylist

Vipul Kohli – Website developer

Main Contribution: Tutor's My Answers and Outside Usability Testing

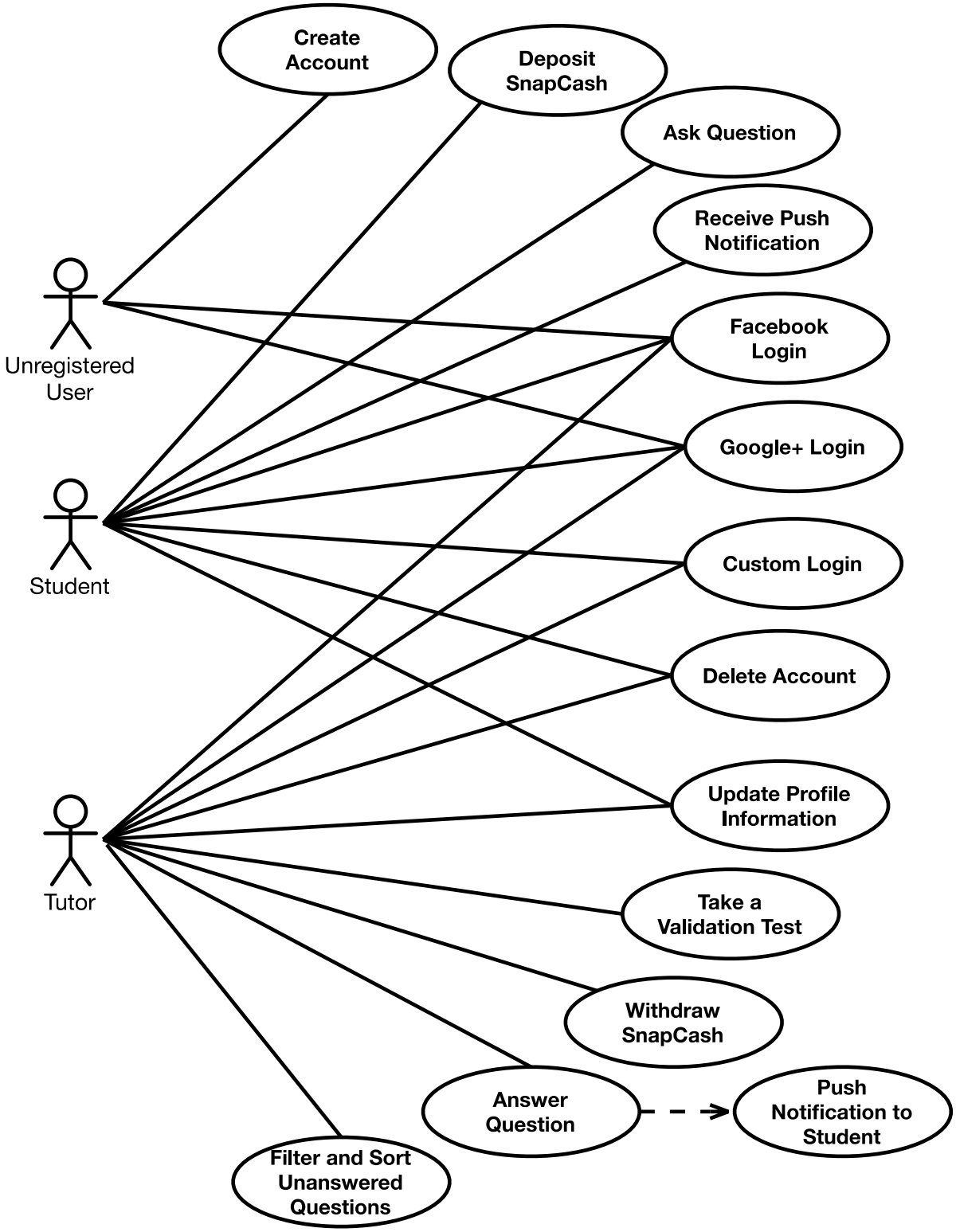
Roman Stolyarov – Website and backend API developer

Main Contribution: Tutor Validation Tests

Elena Villamil – Database manager

Main Contribution: Question and Answer Insertion and Lookup

USE CASE DIAGRAM



1.0 FEATURE SUMMARY

1.1 BOTH 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 warming before deleting the account, and if the user proceeds to delete the account, all information associated with the account is deleted.

1.2 TUTORS ONLY

- 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.

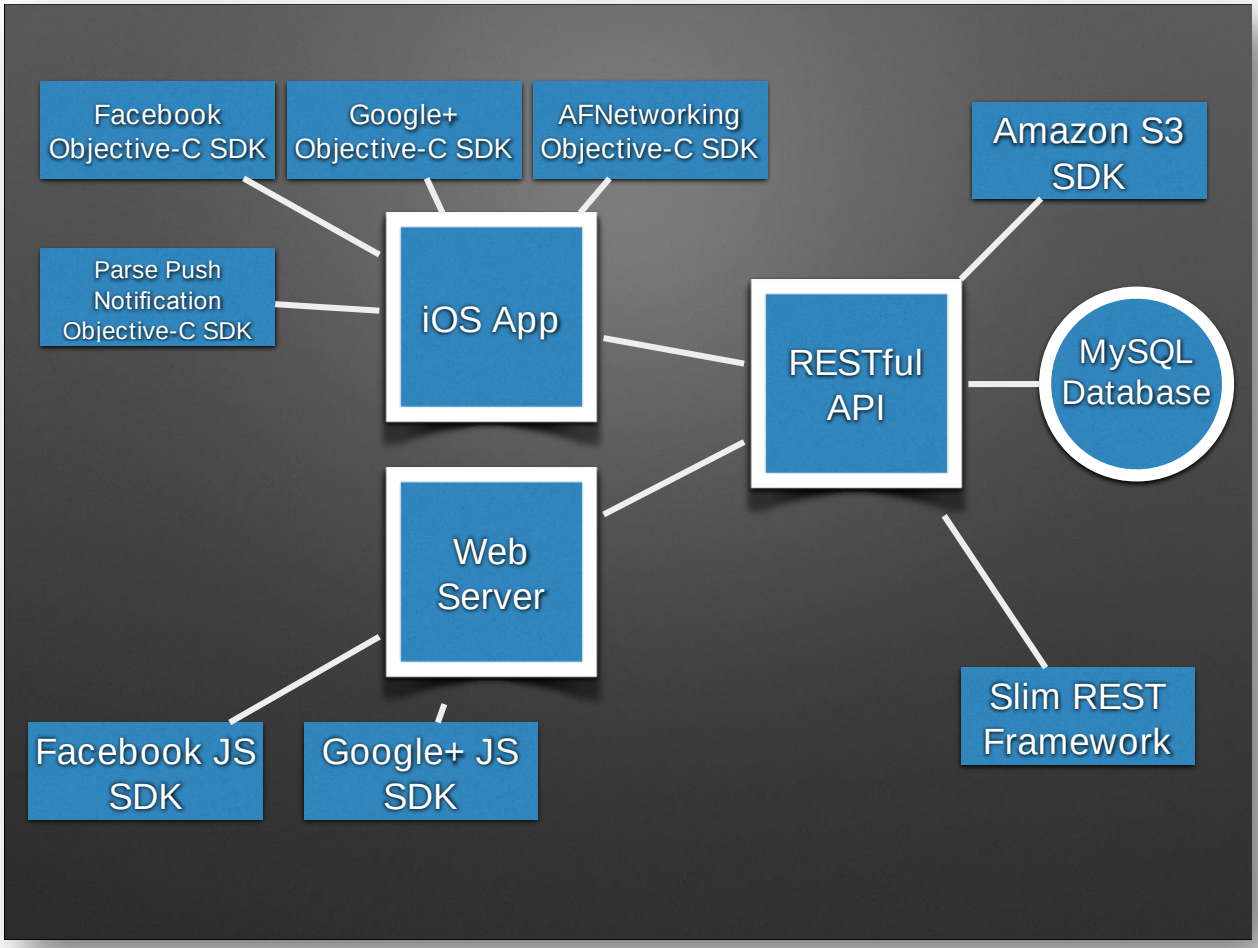
- Validation: Tutors can pass a validation test for each category they would like to be verified in. This rewards the tutor with extra SnapCash for every answer in that category.

1.3 STUDENTS ONLY

- 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.

2.0 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

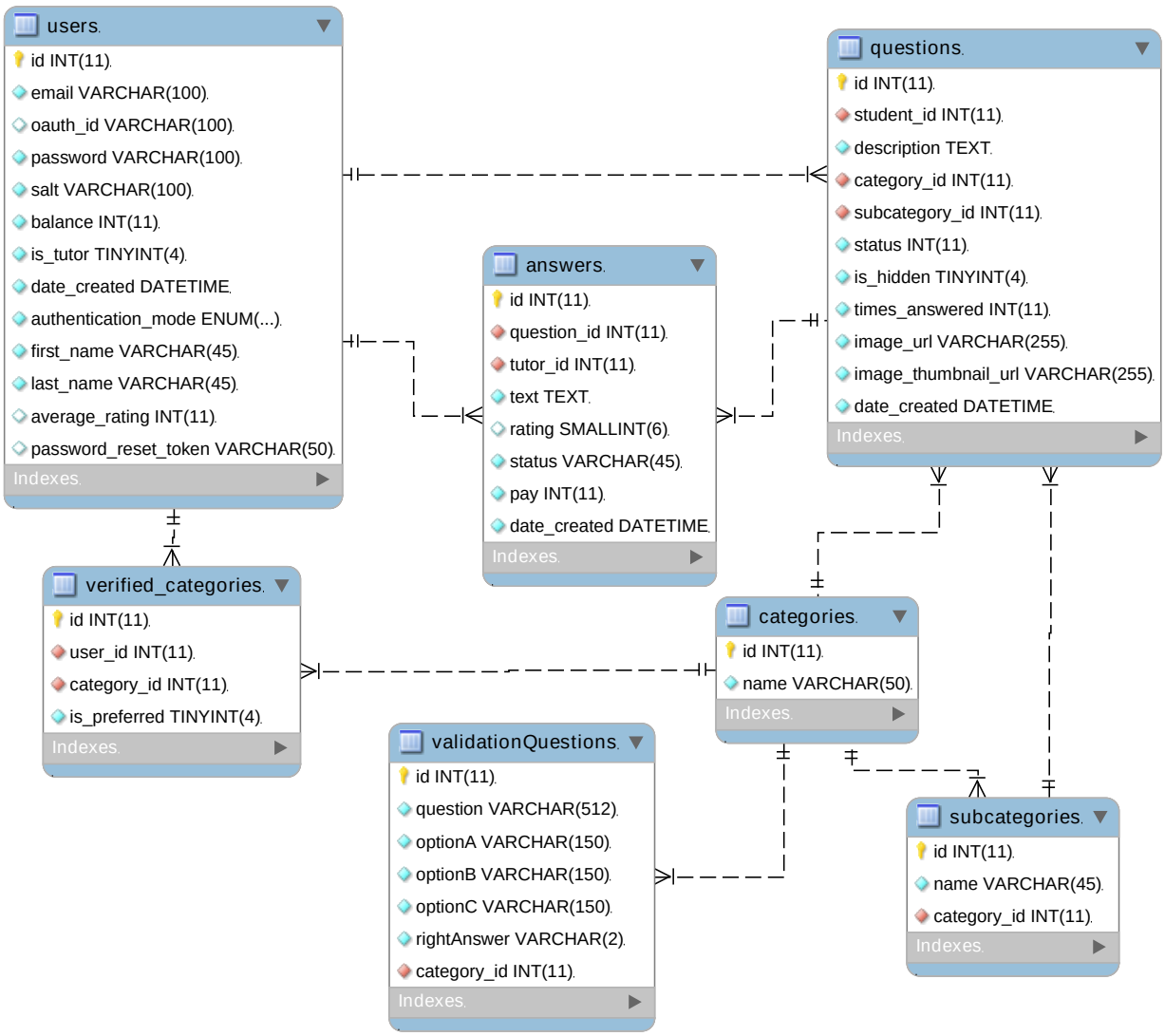
Web server (Presentation Layer)

- 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 specically 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

3.0 DATABASE MODEL

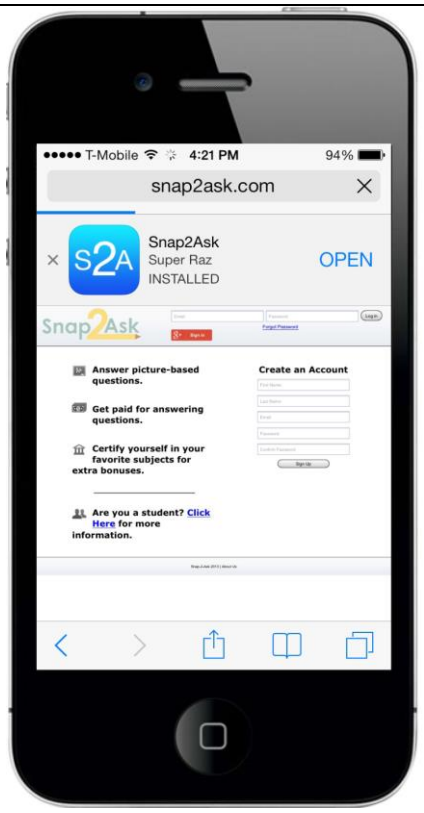


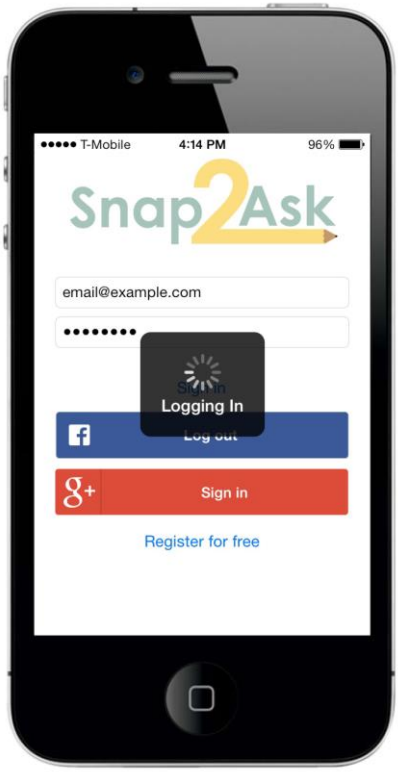
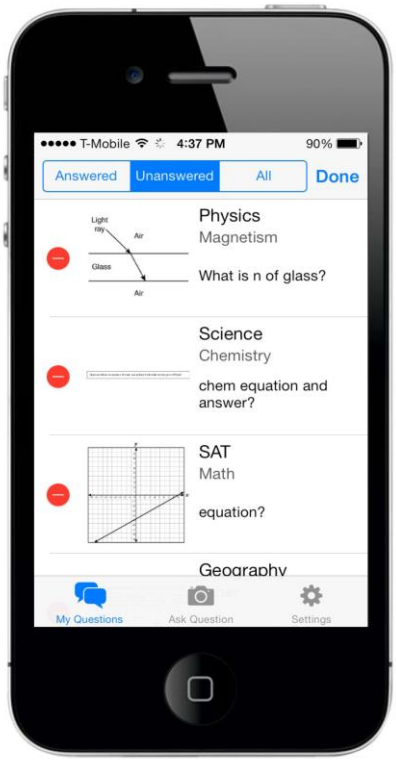
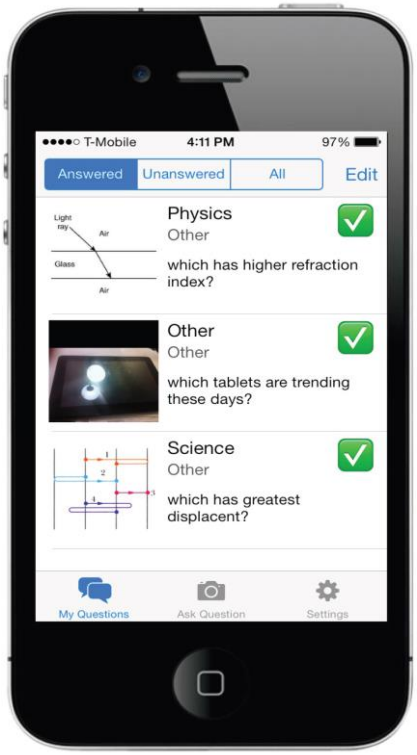
See the Data Dictionary appendix for more details.

4.0 USER INTERFACES

4.1 iOS 7.0+ Mobile App for Students

3 WAYS TO LAUNCH

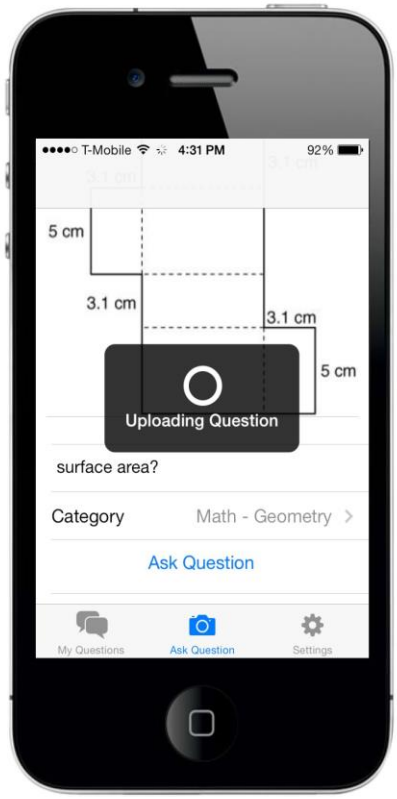
1. App Launcher	2. Safari Web Site	3. Task Manager
Once the app has been downloaded, it can be opened by touching the S2A icon on the Launcher.	New students can open or install the app via the banner when visiting snap2ask.com on an iPhone	Students can return to the app via the Task Manager.
		

STUDENT LOGIN AND QUESTION HISTORY		
1. Login/Register Students can login/register with a Facebook or Google+ account. They can also register an account with their first/last name, e-mail, password, and matching retyped password. A similar progress wheel is used when uploading questions.	**2. Hiding Questions** Students can view unanswered questions. The word "Edit" appears in the top right. By touching "Edit," a red circle icon, and "Done," students can hide old questions in all 3 "Unanswered," "Answered," and "All" tabs.	**3. Answered Questions** Questions that have been answered by a tutor are indicated by a green check.
		

UPLOADING QUESTIONS & RATING ANSWERS

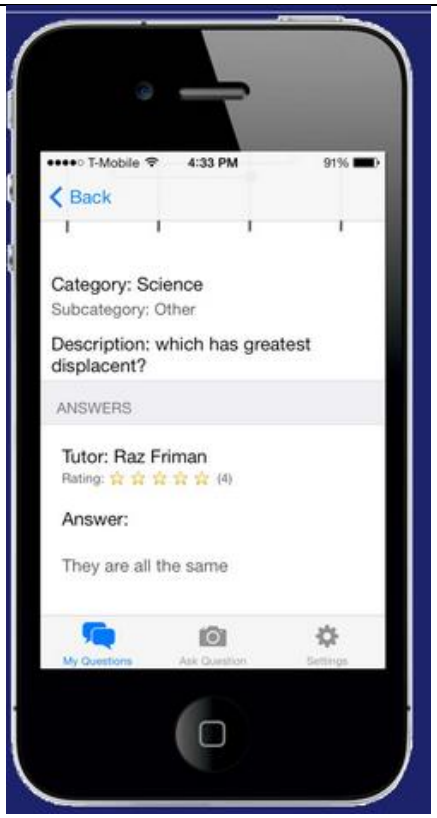
1. Add Picture, Category & Subcategory

Students can "Choose Existing" picture or Take a Picture by touching the "picture area. By touching the "Category" line, a list view opens up to choose a Category with an arrow to choose a from a second list of subcategories.



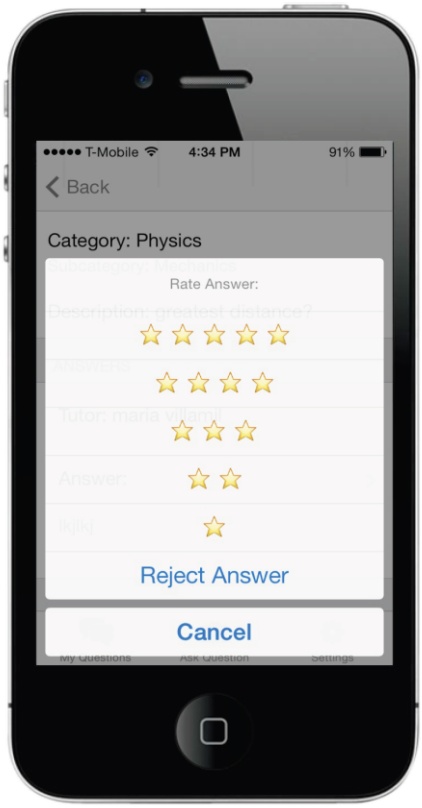
2. Answer Received

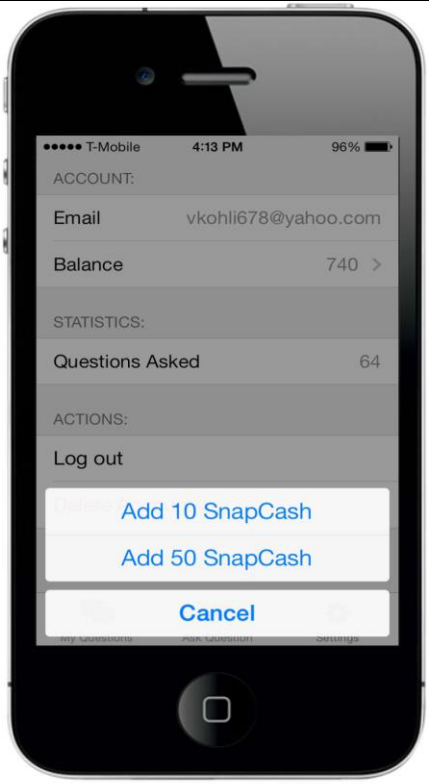
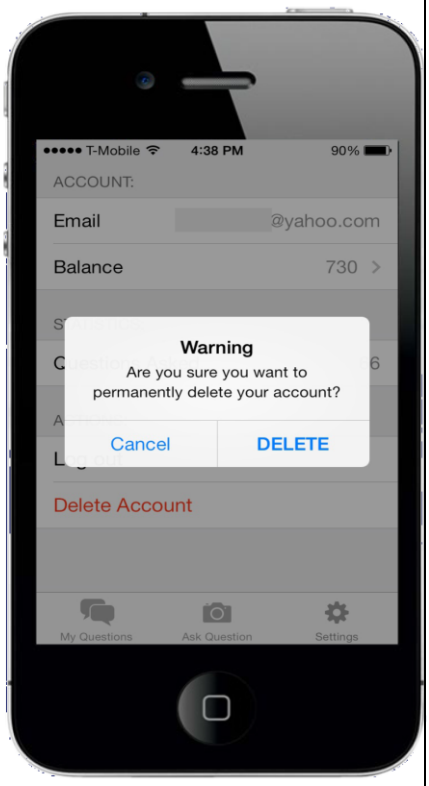
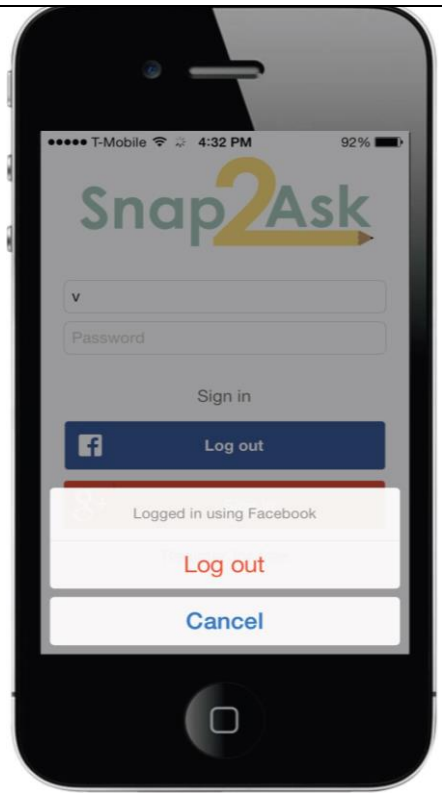
When an answer is received, the student gets a push notification on the iOS 7 lock screen as well as a popup on the app screen. They can view the new answered question in their "Answered" questions. The name and overall rating (out of 5 stars) is shown below the question description. Back goes to the main answer list view.



3. Rate the tutor

The student has a chance to rate the answer out of 5 stars. Students can reject off-topic answers. This rating effects the tutors overall rating. The tutor can view this rating in their "My Answers" Web page.



ACCOUNT MANAGEMENT		
1. Add SnapCash Students are debited SnapCash when they post a question. Tutors are paid SnapCash for answering questions. Students can view and add to their SnapCash balance by touching "Balance" in the "Settings" screen. They can also check their e-mail address and number of questions asked to see if the "Balance" is reasonable.	**2. Logout/Resign** Students can logout from the "Settings" screen as well as delete their account. Similar popup messages appear whenever the student enters an incorrect value at login or registration as well as when the student successfully posts a question.	**2. Facebook Logout** Facebook and Google+ Logout are linked to the device rather than the app so students can log out of these 3rd
		

4.2 WEB SITE UI FOR TUTORS (SNAP2ASK.COM)

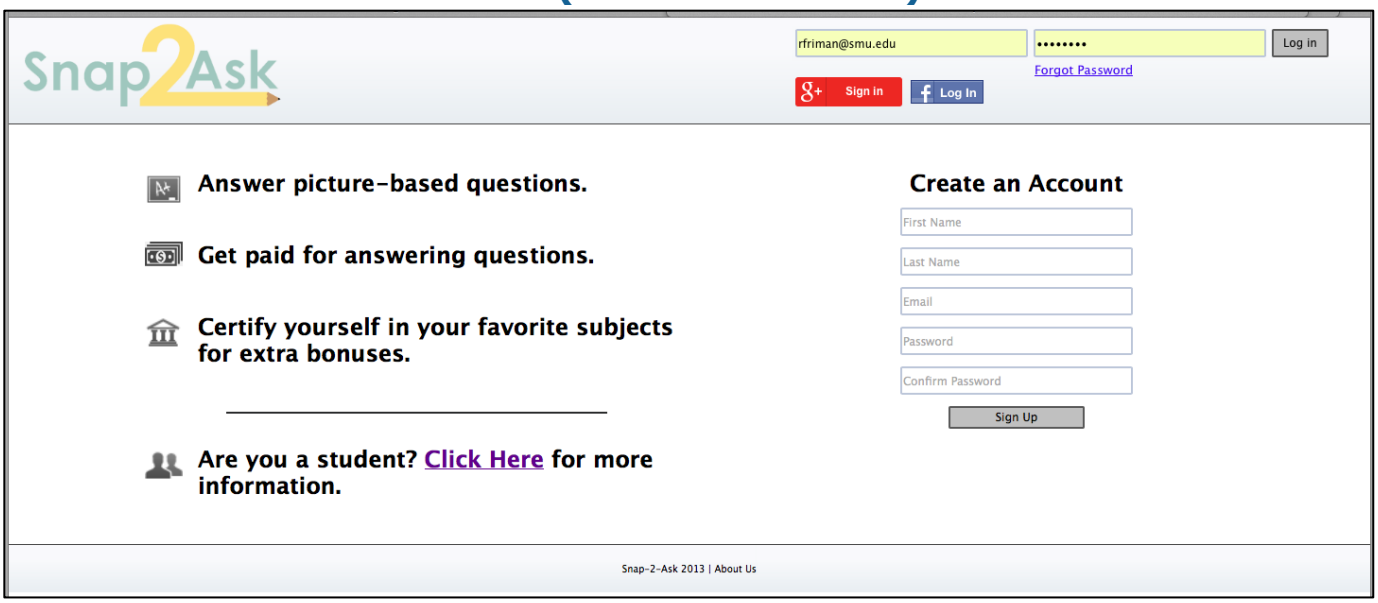


FIGURE 1.2.1 - THE SNAP-2-ASK HOMEPAGE FOR TUTOR REGISTRATION AND LOGIN

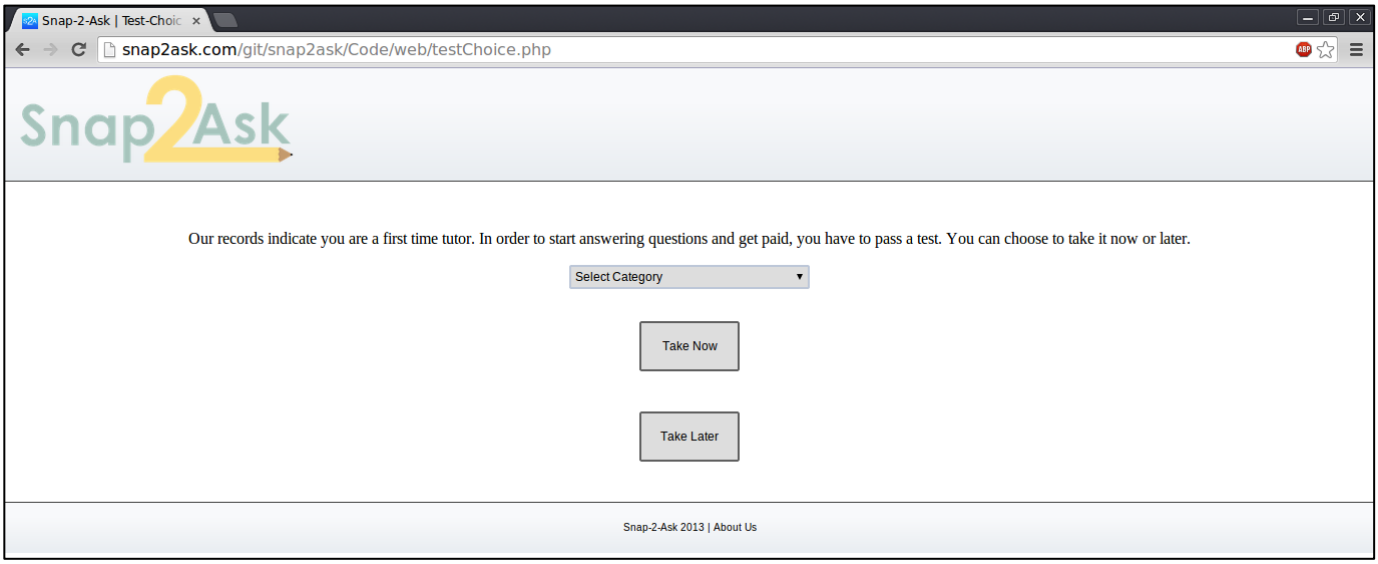


FIGURE 4.2.2 - AFTER CREATING AN ACCOUNT FOR THE FIRST TIME, TUTORS ARE PRESENTED WITH AN OPTION TO VERIFY YOURSELF IN A SPECIFIC CATEGORY.

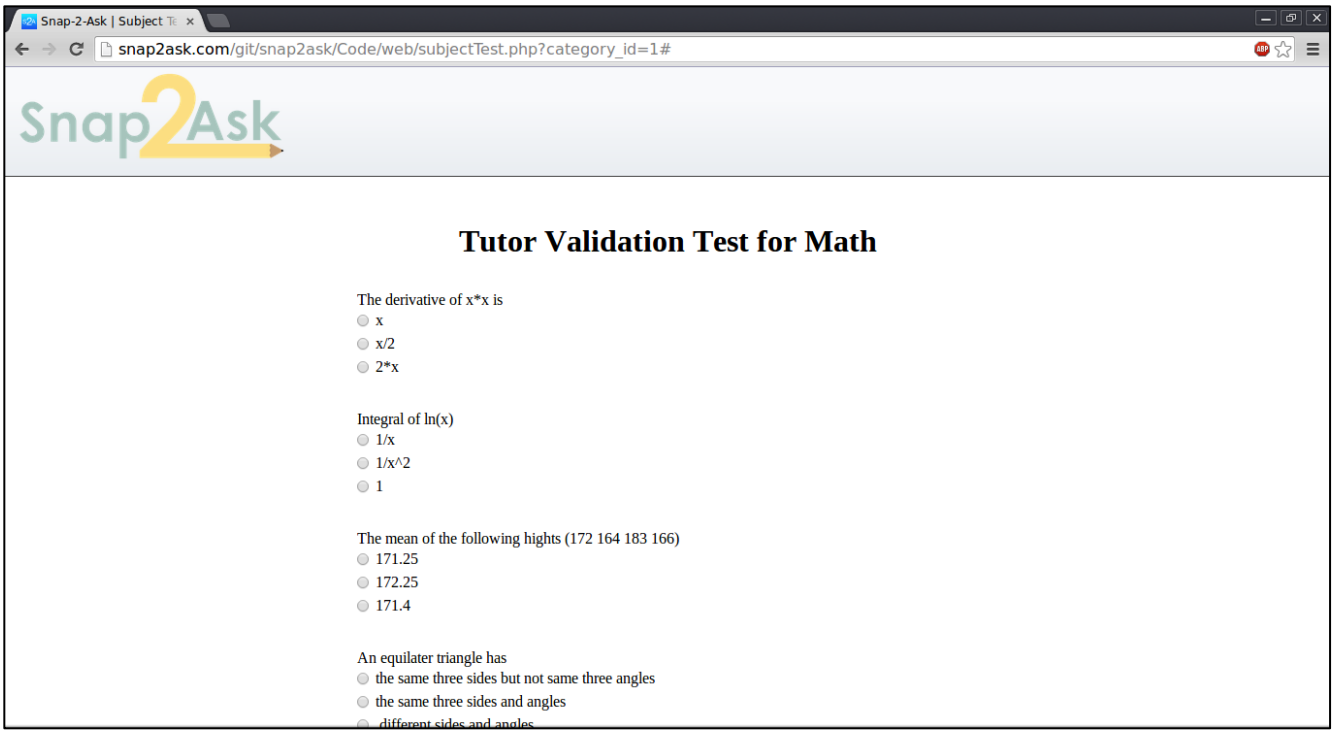


FIGURE 4.2.3 - AN EXAMPLE VALIDATION TEST WITH SAMPLE QUESTIONS FOR THE MATH CATEGORY

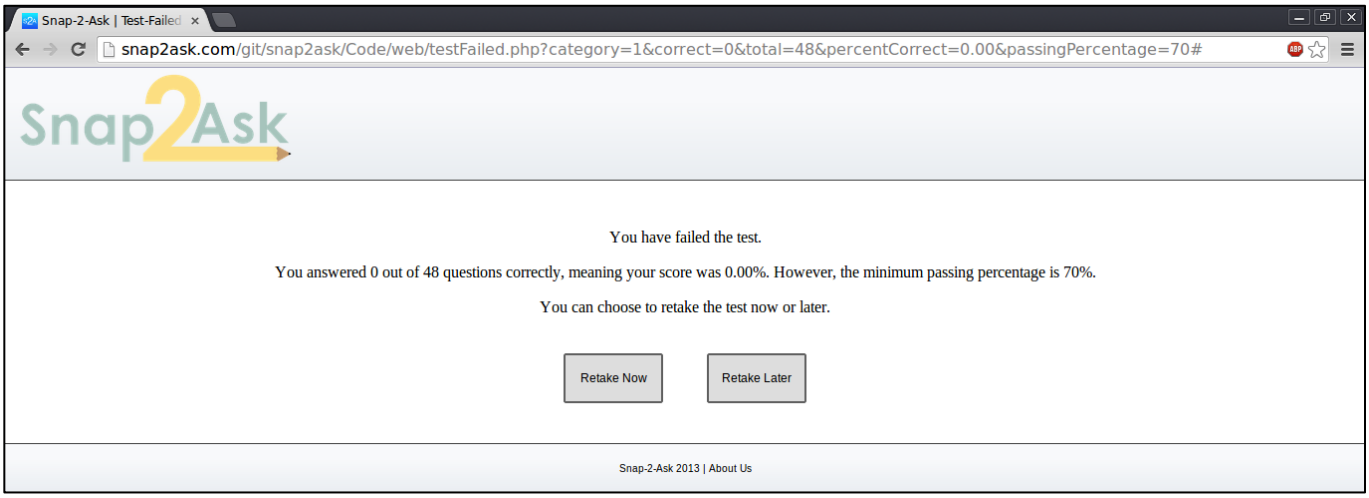


FIGURE 4.2.4 - AFTER FAILING A TEST, THE USER IS PRESENTED WITH THE ABILITY TO RETAKE A TEST OR TAKE THE VALIDATION TEST LATER

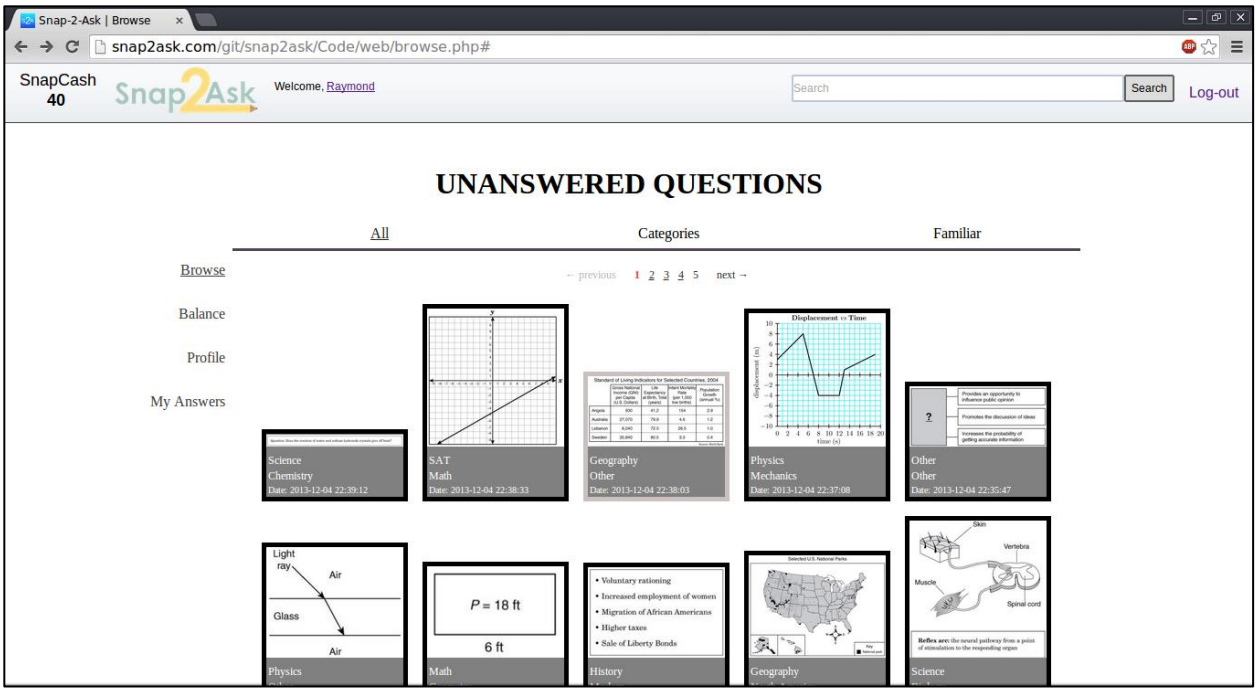


FIGURE 4.2.5 - THE BROWSE PAGE AFTER LOGGING IN. THIS PAGE SHOWS 20 UNANSWERED QUESTIONS AVAILABLE FOR TUTORS TO ANSWER. TUTORS CAN FILTER THESE QUESTIONS BY CATEGORIES OR GO TO THE NEXT PAGE TO SEE MORE UNANSWERED QUESTIONS TO ANSWER AND GET PAID

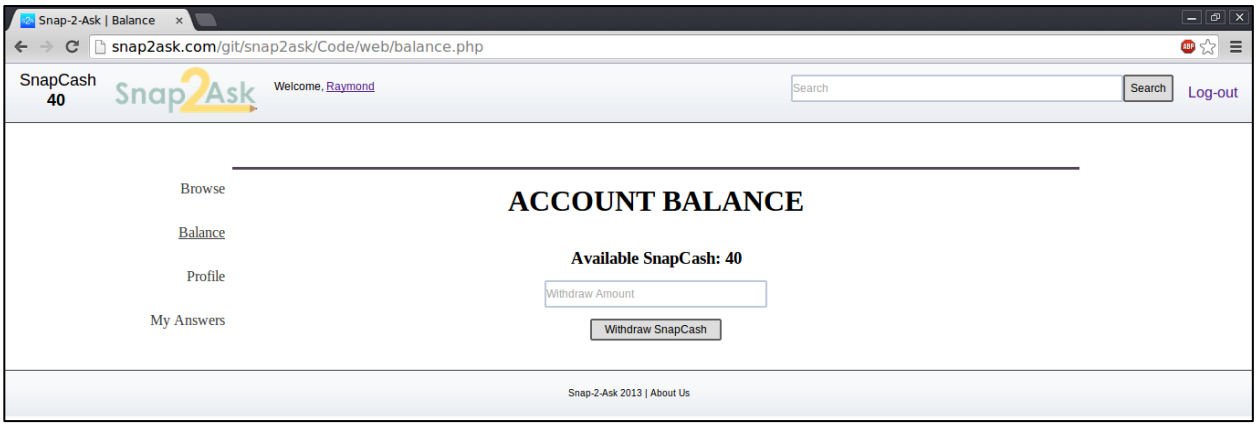


FIGURE 4.2.6 - THE BALANCE PAGE WHERE TUTORS CAN VIEW THEIR CURRENT SNAPCASH BALANCE AND WITHDRAW SNAPCASH EARNED FROM ANSWERING QUESTIONS HERE

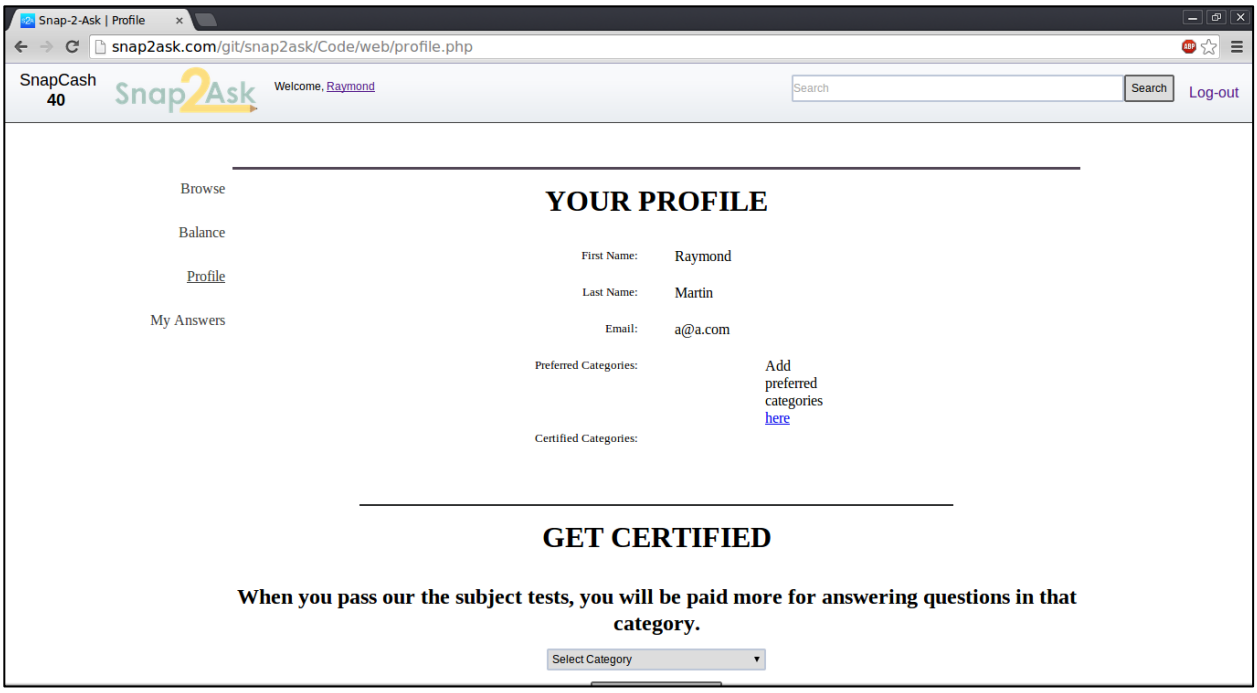


FIGURE 4.2.7 - THE PROFILE PAGE WHERE TUTORS CAN VIEW ALL OF THE DETAILS OF YOUR ACCOUNT. THEY CAN EDIT THEIR PROFILE INFORMATION OR TAKE VALIDATION TESTS IN OTHER CATEGORIES

.

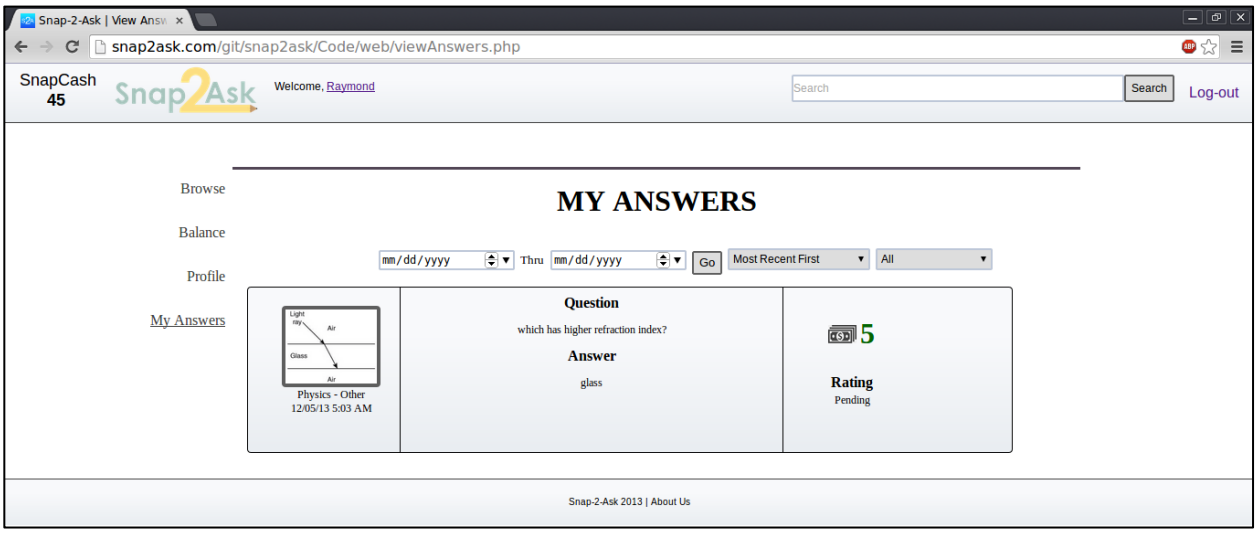


FIGURE 4.2.8 - THE ANSWERS PAGE WHERE THE TUTOR CAN VIEW ALL OF THEIR PREVIOUSLY POSTED ANSWERS AND RATINGS. TUTORS CAN SORT AND FILTER BY DATE, MOST/LEAST RECENT FIRST, HIGHEST/LOWEST RATING FIRST, HIGHEST/LOWEST PAY

5.0 TESTING

5.1USABILITY TESTING

Other than the Feature Set testing done with our assigned 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 first-time users because we have been fiddling with the web 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 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 his suggestion 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 that the web site was developed for tutors while the mobile app was developed for students.. After speaking with these critical usability testers, newcomers to our web site are very optimistic with few minor complaints about the web site although they wait for the day when SnapCash can be turned into real cash.

5.2 REVIEW OF OUR TEST TEAM

Our testing team was an essential resource to the improvement of Snap-2-Ask. 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 every issue accordingly and promptly.

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 developer team, understand our product very well, so we my overlook design elements that are meaningful to our users. However, all users need to be able to understand our product, especially new users, so the testing team's feedback was greatly appreciated. Story Zanetti reported an example: "When registering an account with the iPhone app, the confirm password is hidden from view when the soft keyboard appears." As developers,

we just focused on the input values being processed, overlooking that the input field was not very pleasing to the registrant.

Several of the bugs the team reported were login/logout loopholes that would be obvious to the user, but not easily seen by the developer because we are mainly focused on developing features while logged into one account. For example, one of the bugs Jessica Yeh found was tutors logging in through Snap-2-Ask were prompted to take a validation test while tutors using the Facebook or Google+ Login skipped the validation step. Evan Kohn found that on the iOS 7 App students could log out and create a new account, but they were logged back into their old account as if they never logged out. These

5.3 OUR PERFORMANCE AS A TEST TEAM FOR PONYPARK

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. We communicated back and forth using the "Comments" feature of GitHub, clarifying any misunderstandings as well as offering suggestions to what improvements could be made to the PonyPark web site and Android application.

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 need be. 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.

6.0 TEAM REFLECTION

6.1 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.

Presentation on our web site via CSS was especially a challenge. Understanding why a CSS property is not executing like expected invoked our understanding of the precedence of CSS selectors.

Maintaining JavaScript search filtration of the Tutor's My Answers page was also a challenge because event handlers would not always behave because the internal HTML has been altered and the event handlers were then rendered ineffective. Several logic errors were apparent as well and took quite a bit of debugging to achieve the expected output.

6.2 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 7 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.

JAVA TEXT & IMAGE FILE CONVERSION

Vipul extracted several images and test questions from several online tests that were used as validation questions for tutors in the subjects of Science, History, and Art. Snap-2-Ask only supported 3 answer choices on validation tests, so Vipul had to use Java to eliminate a wrong answer choice. The Java program had to convert HTML from online tests to CSV to insert validation questions to our database.

Several of the excellent question images on the Snap-2-Ask tutor browse page were originally transparent GIF image files. Vipul had to build a Java program to white-out all transparent regions of these images to make the images presentable on our web site.

6.4 WHAT WE WOULD HAVE DONE DIFFERENTLY

We depended on our usability test team, but they were not enough. By working with random users, we found several loopholes in our validation mechanisms as well as complaints about some bugs or usability issues within our site and mobile app. Different usability testers gave different impressions of our web site that other usability testers just ignored because we were guiding them rather than observing them. In future projects, I will be sure to do more usability tests as new features are implemented, and let the user naturally run through our web site or mobile app complimenting or complaining at the end of the test. Finding bugs at the end is not very helpful because several components are integrated, so fixing bugs across the entire site is difficult to do.

Another important difficulty was communicating 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 because only two people were paying attention at a time. We discussed different features and divided roles via GitHub, but not everyone actively participated in the discussions. We would like to have chosen a better method of communication to ensure that all of our team members are always on the same page as far as contributing their part as well as actively improving other members' work. However, in the end, we all came together and accomplished everything we set out to do in terms of our Feature Set Plans.

7.0 SUMMARY AND FUTURE FEATURES

Overall, Snap-2-Ask provides a huge set of features to assists 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 avaiable.

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 3rd 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 use 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 use 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 questions 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