# Snap-2-Ask

Technical Design Document

SMU | Bobby B. Lyle School of Engineering

# Table of Contents

# SOFTWARE REQUIREMENTS

* As used in these requirements, the term "student" is synonymous with "mobile interface user" and the term "tutor" is synonymous with "web interface user."

## FUNCTIONAL REQUIREMENTS:

### Requirement 1

Description: The web and mobile interface shall allow a tutor and a student to log into their accounts.

Rationale: Tutors and students need to be able to log into their specific accounts, in order to see their own profile information, questions/answer.

Testing Criteria: User validation and password authentication.

Data model: The database contains a user table with the information needed to validate the users.

### Requirement 2

Description: The web interface shall allow a tutor to answer questions by typing the answer or by posting a picture that answers the question.

Rationale: The tutor should be free to either type the answer to the question (since this is easier) or write the answer if it involves complex symbols or figures. If the tutor chooses the latter option, they can take a picture of their writing and upload it as an image.

Testing Criteria: make sure tutors are able to successfully submit answers that will be seen by the students.

Data model: The database has an answer and a question table. When a tutor answers a question the answers table will get populated. The answers table contains a foreign key with the id of the question it is answering (this is how we are relating both tables)

## Requirement 3

Description: The mobile interface shall allow a student to ask questions by posting pictures.

Rationale: The main advantage of Snap-2-Ask is giving students the ability to easily post their math and science questions without having to type out symbols, graphs, or figures. The easiest way for them to post is to directly take a picture of the question from their textbook.

Testing Criteria: The student is able to submit a picture and this will appear as an unanswered question for the tutors.

Data model: When a student adds a question the questions table gets populated.

## Requirement 4

Description: The system shall contain one type of account allowing both tutor access through the web interface and student access through the mobile interface.

Rationale: The first version of Snap-2-Ask will be based on the idea that most students study with their smartphones by their side, and thus taking a picture of a question is quick and natural. On the other hand, tutors should be able to see all of the questions laid out for them on a large interface and type lengthy answers with a computer keyboard.

Testing Criteria: Students will not be able to use the web interface, and tutors will not be able to use the IOS app.

Data model: Any user in the users table can access any of the two interfaces.

## Requirement 5

Description: The system shall allow creation, deletion, or editing of an account from both interfaces.

Rationale: Account creation is a multifaceted process of personal and monetary information submission and email verification that is easiest to implement first for a web interface. Eventually, this functionality may be added to the mobile interface as well.

Testing Criteria: When a user deletes his account there is no information about this tutor left in our database.

Data model: When any change is made to a user's account the user table in the data model will be updated. When an account is created or deleted, a row in the users table will be added or subtracted, respectively.

## Requirement 6

Description: The system shall allow creating an account by logging in through Facebook or Google on either interface, and by entering account information.

Rationale: If users can login through Facebook or Google they may be more motivated to get started as this would make the verification and account creation process simpler and faster.

Testing Criteria: The user of either interface is able to login by providing solely their Facebook username and password or Google username and password.

Data model: The table for registered users has a field for authentication mode, which specifies which method the user employed. Other than this, authentication for Google and Facebook is handled by the server without needing to query the database.

## Requirement 7

Description: The student may see previously discarded answers.

Rationale: If the student discarded a question accidentally or because he wanted to obtain some more information, it will be helpful to the student to be able to view the previous answers.

Testing Criteria: all the three answers (if the student discarded the first two) must be available for the student to see.

Data Model: The model allows a natural join of the questions table with the answers table and display all of them, regardless of the answer status.

## Requirement 8

Description: The mobile interface will open with a camera view and allow the user to take a picture of a question.

Rationale: Opening with a camera view will make the main function of the app, submitting a question, the easiest and quickest to use.

Testing Criteria: When opening the iOS app the camera must go on.

Data model: This requirement does not relate directly to the database. It is based merely on user convenience.

## Requirement 9

Description: The student and tutor shall have thorough description of all the monetary policies associated with Snap-2-Ask available from either interface for viewing.

Rationale: The student or tutor may want to quickly reference all the policies while they are using the mobile or web interface, so a textual reference will be available for each

Testing Criteria: test that a page with all the information is displayed when the user requests it

Data Model: This requirement isn't related to the data model.

## Requirement 10

Description: The student shall be able to send his question to the web server.

Rationale: The web server will handle processing of the question, including inserting it into the database and then responding with it to appropriate HTTP requests.

Testing Criteria: The web server will be able to predictably and consistently store the information received from a question in the database, and its associated image in the file system.

Data model: The description, category, subcategory, and path to the associated image will be stored for every question in the questions table.

## Requirement 11

Description: The student shall receive a notification when a question has been answered.

Rationale: After posting his question the student will likely have to wait minutes or hours for a response, and may even forget about it. For this reason, he will be given some notification from the app when an answer is ready.

Testing Criteria: Some kind of alert must be display in the student's phone when one of his questions has been answered.

Data model: The row for students has a field for "answered" questions.

## Requirement 12

Description: The student shall have the option of either rating the answer or discarding it, thus opening his question to new answers.

Rationale: The student's rating will be used for calculating the tutor's rating, and the student is thus encouraged to rate. If the answer is poor or insufficient, the student can discard it and wait for a new answer so that they feel like they can get their money's worth.

Testing Criteria: after a student discards an answer the question must go back to the unanswered questions and the tutor that replied previously must get his rating decreased.

Data model: the questions table has a field with status, as long as the value in times_answered is smaller than 3 (maximum of three answers per question), the question will be open for more answers.

## Requirement 13

Description: The student shall be able to navigate to a view showing them their entire history of questions asked and whether or not they have been answered.

Rationale: The student may have asked multiple questions and is waiting for notifications about multiple answers. Seeing this view would allow the student to determine which questions have or have not been answered.

Testing Criteria: The student shall be able to navigate to a page showing their entire list of questions asked, and each question will display a distinctive icon if it has been answered.

Data Model: The data model supports a "status" field for every question, which indicates whether or not it is open for answering.

## Requirement 14

Description: The student shall be able to view answers to their own old questions.

Rationale: Students may want to review answered questions from their past. In this case, they are free to navigate to any of the questions they have asked and see their answers and discarded answers.

Testing Criteria: When a student clicks on an old question for which they have already viewed the answer, they will still be able to view the answer.

Data Model: The answers table will permanently store all answers until the student deletes the associated question.

## Requirement 15

Description: The tutor shall be able to create an account without taking any tests, but in order to activate his account he must pass a test for each category he wants to be validated on.

Rationale: Ideally, the tutor would take a test for any category of question he would like to be able to answer. This ensures the students that their answers will be credible, and will motivate them to use Snap-2-Ask instead of asking their friends or posting questions for free on Facebook.

Testing Criteria: A non-validated tutor will not be able to submit answers to any questions, but will only be able to view the answers.

Data Model: Our table for users has a field indicating whether the user has been validated for any subjects.

## Requirement 16

Description: The student shall be able to provide their credit card information and make a deposit onto their account (simulated in feature sets 1 and 2). Initially, we will not deal with actual, but simulated financial information.

Rationale: The student must pay for being able to view answers to a question. The easiest way to do this would be taking the student's credit card information and having them make a deposit of an amount they choose to begin with.

Testing Criteria: the balance of the user increase by the amount they transfer.

Data Model: there is a field "balance" in the users table will get updated when the user transfers money.

## Requirement 17

Description: The tutor shall be able to see the unanswered questions in a grid-like arrangement on their screen, ordered by oldest unanswered submissions.

Rationale: The tutor should clearly see all the questions most recently asked in a dynamic interface that suggests that questions get asked frequently and that the site gets a lot of traffic. The grid-like form would also be the most clear visual organization.

Testing Criteria: make sure that the oldest question that has been waiting on the queue is the one that appears first in the grid-like arrangement.

Data Model: the field "data_created" in the answers table will be used to fulfill this requirement.

## Requirement 18

Description: The tutor shall be able to search for unanswered questions by category.

Rationale: The tutor might only be able to answer questions in a particular subject area, so he should be able to search for only questions of that sort.

Testing Criteria: when tutor click in order by category and selects a category, all the unanswered questions of that category will appear.

Data Model: in our data model each question is associated with a category id and status. Joining the questions table with the category table we can get the name of the question category.

## Requirement 19

Description: The tutor shall be able to search for unanswered questions by keyword. The result will be order by oldest unanswered question.

Rationale: The tutor may be able to answer only questions that are subtopics of subject areas, and he should be able to search for these.

Testing Criteria: When the tutor clicks on order by key word and enters the keyword, all the unanswered questions containing that keyword in the description must be display.

Data Model: the keyword must appear in the description field in the questions table, and the status must be false, in order for the questions to appear in the list.

## Requirement 20

Description: The tutor shall have a rating associated with their account.

Rationale: The tutor's compensation per question will be determined by his rating relative to all the other tutors in the system. In addition, it will clue students into the reliability of the tutor's answer and help them to decide whether to discard it or not.

Testing Criteria: there is no tutor without rating.

Data Model: there is a rating field in the users table. When the tutor is first created the rating is initialized to 0.

## Requirement 21

Description: The tutor rating will be derived from the number of questions they have answered and the average rating given by the students.

Rationale: The tutor rating should be some logical function of their individual ratings from all of the questions they have answered, and an average seems sound here. The main goal with the rating is to give students an idea of the tutor's experience.

Testing Criteria: rating gets calculated follow the same formula and criteria for every tutor.

Data Model: there is a field in the users table that holds the rating, but the data model is not involved in actually calculating the rating.

## Requirement 22

Description: The tutor shall be prohibited from answering their own question or rating their own answer.

Rationale: The tutor might take advantage of the system in this way so as to increase their rating or experience.

Testing Criteria: same user_id won't be able to answer a question posted by that user_id. A user with the same id that the one posting the answer won't be able to rate that answer.

Data Model: questions have a student id, since students/tutors are the same user, they will have the same user id; therefore, the student id in the questions table can't be equal to the tutor id answering the question.

## Requirement 23

Description: The tutor shall be able to link their bank account to their account on Snap-2-Ask and withdraw a specified amount of money less than or equal to their account balance. In feature sets 1 and 2, this process will be simulated.

Rationale: The only way the tutor's account balance will mean anything to them is if they have an associated bank account they can withdraw money to according to the amount in the balance.

Testing Criteria: for features 1 and 2, just test that the balance of the tutor is decreased according to the amount the transfer to their accounts.

Data Model: the balance field in the users table will be updated

## Requirement 24

Description: The tutor's balance shall increase for every answer that has not been discarded for one week, by an amount that is a function of their rating.

Rationale: Students are given a certain amount of time to discard poor answers. If they do not do so within a week, Snap-2-Ask assumes the answer was good and rewards the tutor. This is the basis of our compensation mechanism.

Testing Criteria: balance of the tutor increase for each answer that is being alive for over a week. Test that the amount the balance is increased is correct

Data Model: there is a field in the users table that will hold the balance, but nothing in the data model is actually involved with assigning the rating or determining if the question is being alive for more than a week.

## Requirement 25

Description: The student's balance shall decrease for every question that has been answered. If the student discards an answer, the balance will shall not decrease for every other answer of the same question. Maximum of three discards per question.

Rationale: students will pay a fix amount for question answered. Weather they use the 3 answers they can get or they just use one, they will be charged the same.

Testing Criteria: test that when a student discards an answer, he doesn't get charge more.

Data Model: the answers table has a field with its status.

## Requirement 26

Description: The student has a maximum of three discards before he must stick with the final answer or post the same question again.

Rationale: otherwise a student could get as many answers to his question as he wants, and the small amount they pay for a question will have to be divided among a large number of tutor. This won't be fare for tutors

Testing Criteria: after a questions has receive three answers it won't be visible for tutors to answer.

Data Model: the questions table has a field with the number of times it has been answered, as well as a status.

# NONFUNCTIONAL REQUIREMENTS:

## Requirement 27

Description: When creating an account through the web interface, the user will be given a thorough description of all monetary rules for their account that they must agree with in order to continue.

Rationale: The rules for making and paying money on the account are simple but it is important for all users to understand and be aware of them before interacting with Snap-2-Ask at all. This will ensure that they are kept happy using it.

Testing Criteria: test that an account cannot be created without accepting the terms conditions.

Data Model: the data model isn't involved in this requirements

## Requirement 28

Description: Each account will be an object of the Account class, which shall contain the user's first name, last name, email address, username, password, rating, and current balance as attributes. Both student and tutor will use the same account.

Rationale: Having one type of account that both students and tutors can use makes things easier for those users who would like to be both. It is only that the account will be presented differently depending on which user interface is being used.

Testing Criteria: it won't be more than one type of account.

Data model: this is all the information contained in the users table

## Requirement 29

Description: Each question submission is an object of the Question class initially containing an image, a category, an optional description, and a null answer pointer.

Rationale: Questions will be easier to organize if each one asked simply creates a new object. The student will be required to take a photo and name a category before submitting, although they can optionally include more keywords or description of the question that will make it easier to search for.

Testing Criteria: a minimum of a picture and a category must be entered when asking a question in order for the submission to be valid.

Data model: this information matches the information stored in the question table

## Requirement 30

Description: The web server shall accept new questions being submitted, recognizing the user submitting them, noting the time of submission, and updating the database accordingly (first come first serve).

Rationale: when a logged in user submits a question the servers is responsible to provide the information needed to add the question to the corresponded API function.

Testing Criteria: test that the submitted answered is related to the correct user id.

Data Model: users and questions table.

## Requirement 31

Description: The web server shall accept answers to questions, noting the user and time of the answer. It should update the database accordingly and then send the student a notification that an answer has been received.

Rationale: when an answer is submitted by a tutor the server needs to add the information to the database, and then notify the student that his answers has been answered.

Testing Criteria: When an answer is submitted the correct tutor id is linked to it, and the student who posted the question must be notified.

Data Model: the information in users, answers and questions will be used.

## Requirement 32

Description: The server shall be able to accept information for new accounts and create them in the database or edit or delete existing accounts.

Rationale: when the user enters information to create a new account or update an existing one the server should send this information to the corresponding php script, so the information gets updated in the database.

Testing Criteria: after an account has been created or edited, the proper php script is called with the information submitted.

Data Model: no use of the data model for this requirement.

## Requirement 33

Description: The server shall keep track of account balances by updating the database any time a monetized transaction has taken place.

Rationale: the server is the one in charge to call all the needed functions to update the database after a monetary change has been done in a user's account.

Testing Criteria: after changes in a user's monetary balance the database must be refreshed and the information updated.

Data Model: the data model isn't involved in this requirement.

## Requirement 34

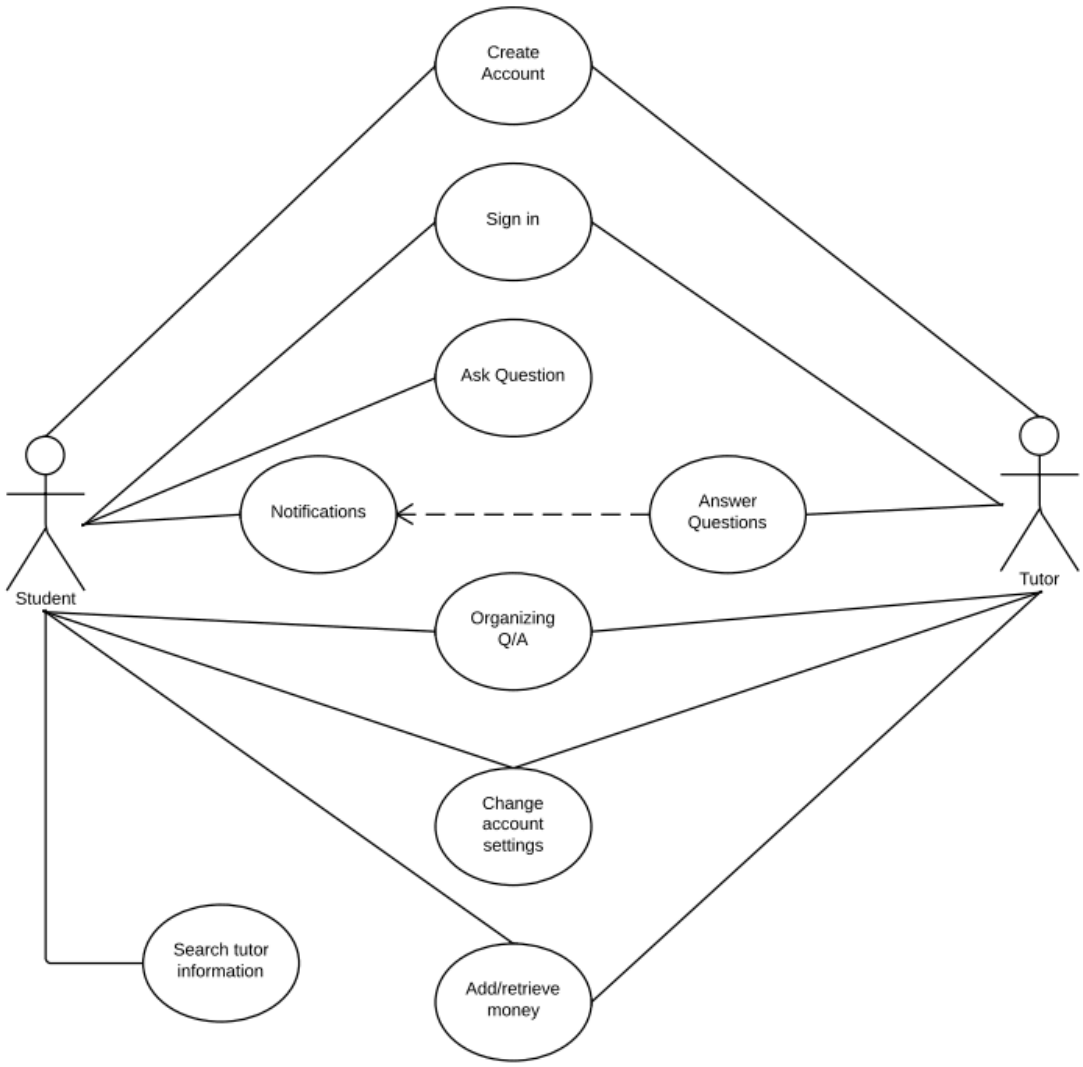
Description: The database shall be composed of five tables: answers, questions, users, categories, and subcategories.

Rationale: these are the tables needed to implement the user functionality that we had planned for our web page.

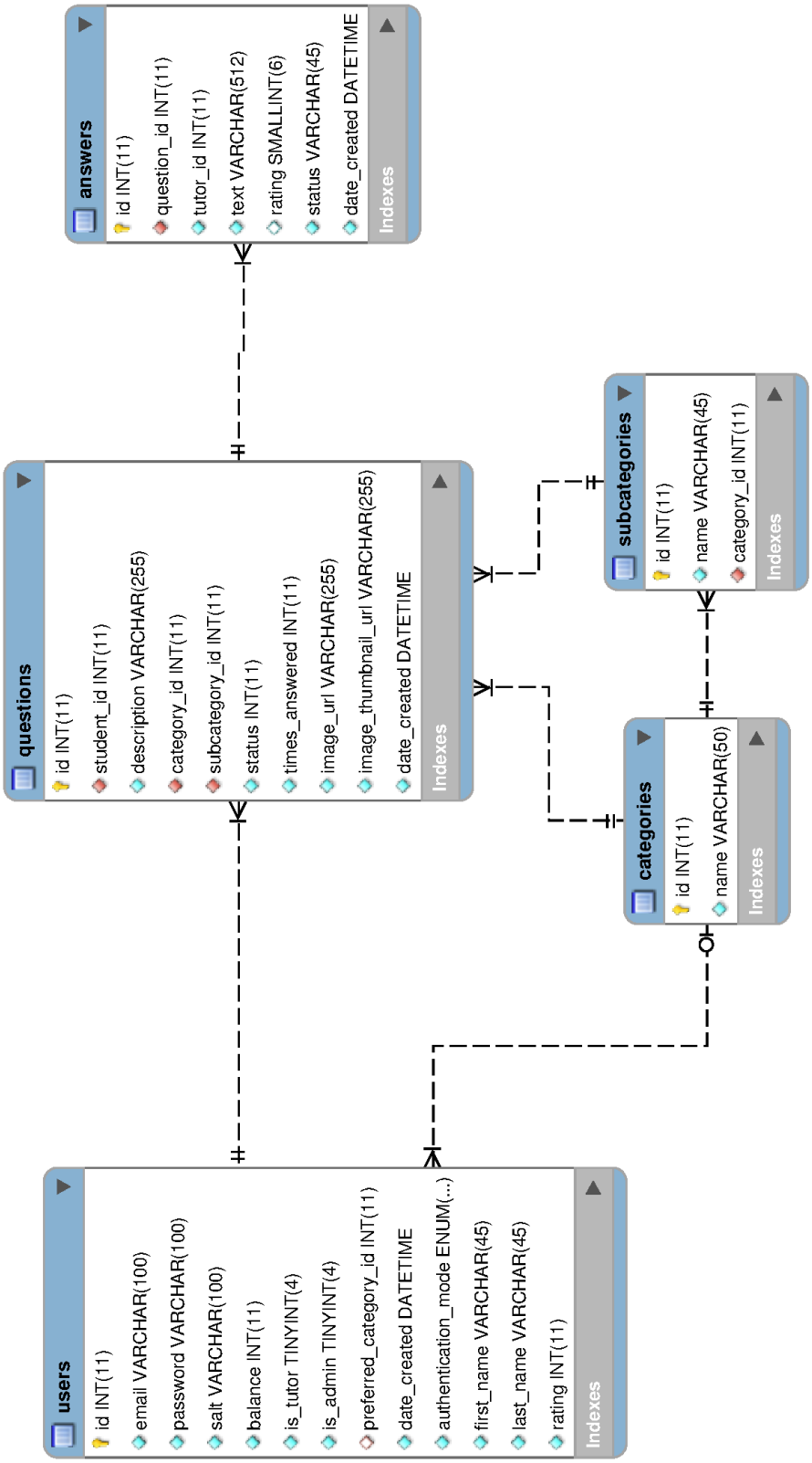
Testing Criteria: these 5 tables and their attributes must be part of our database

Data Model: our data model will have these 5 tables.

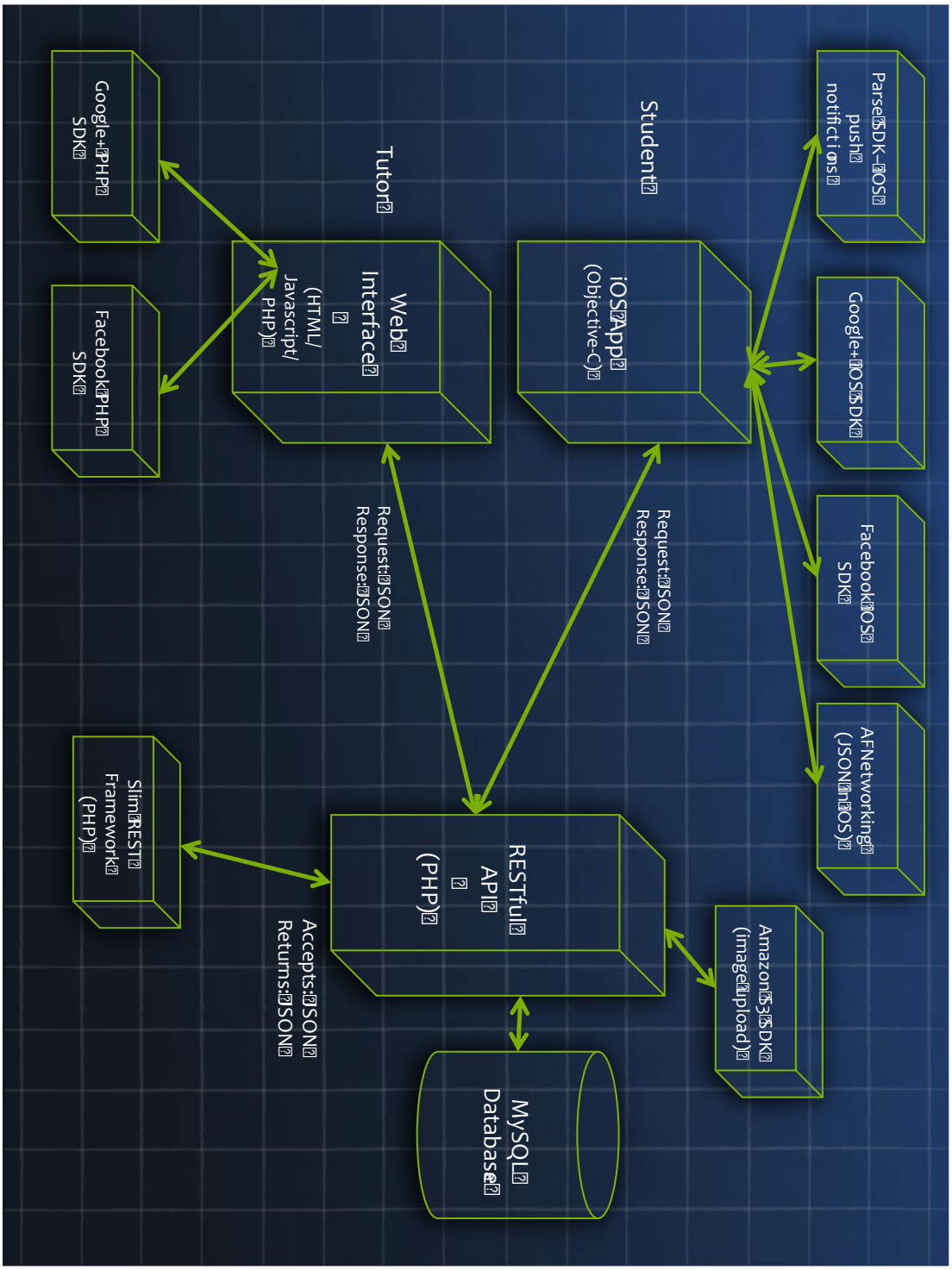
# USE CASE DIAGRAM



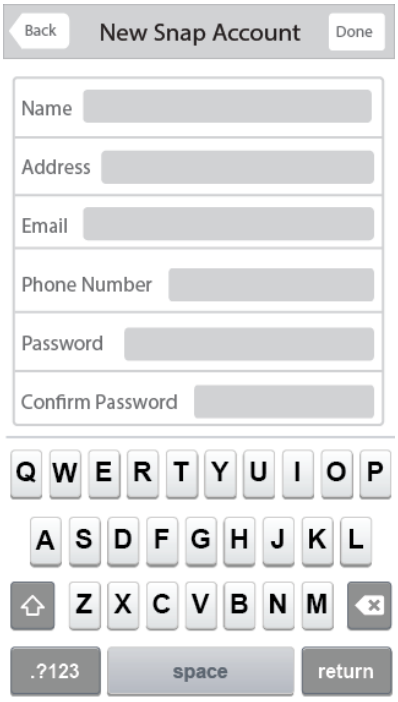
# DATABASE MODEL

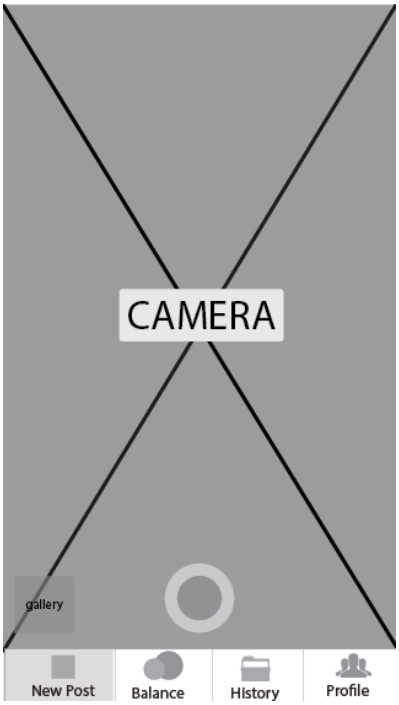


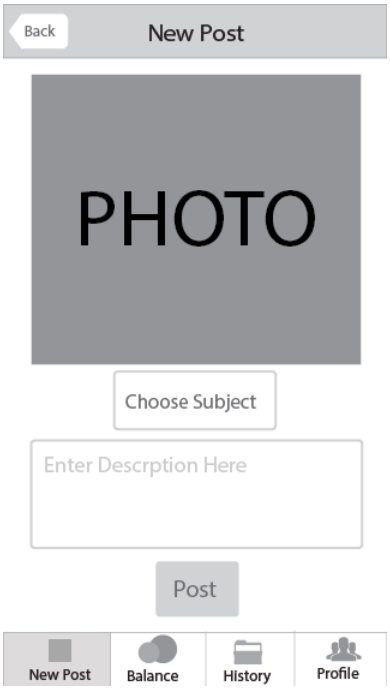
# SOFTWARE ARCHITECTURE DIAGRAM

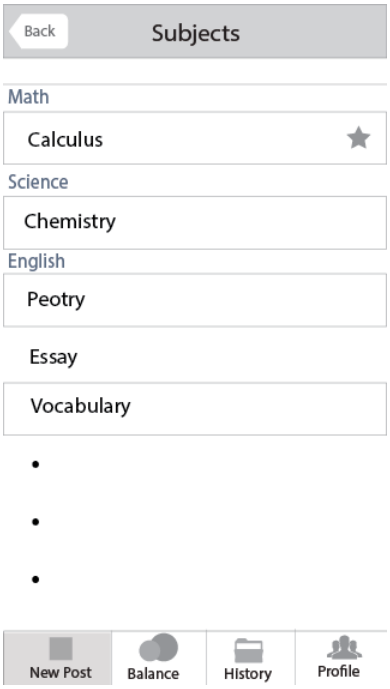


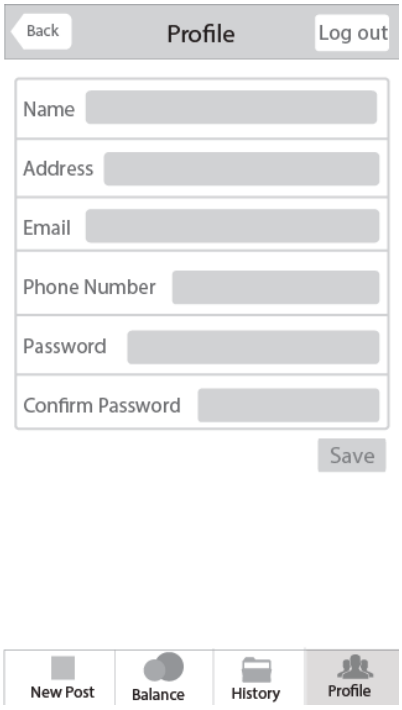
# IOS PAPER PROTOTYPES

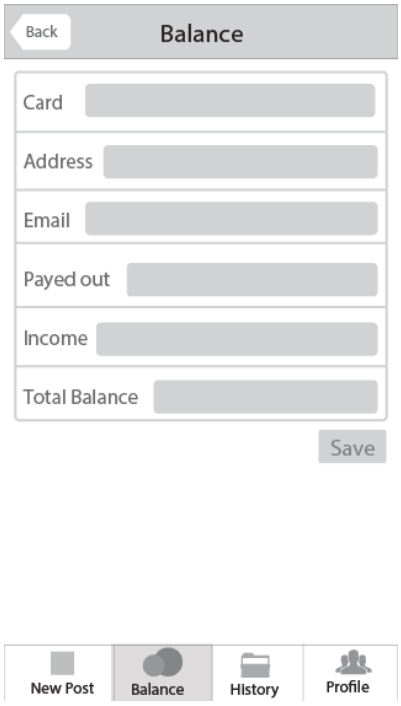



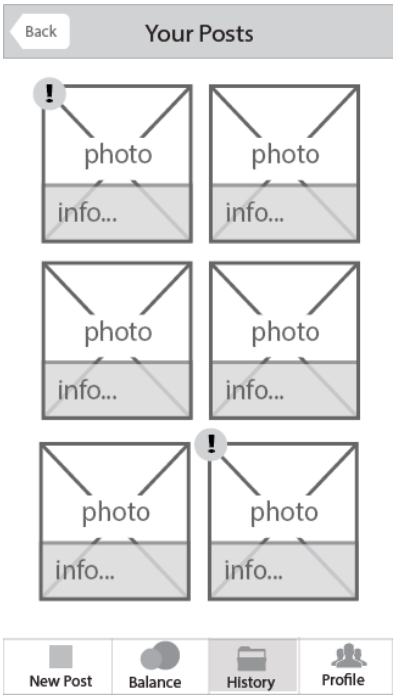



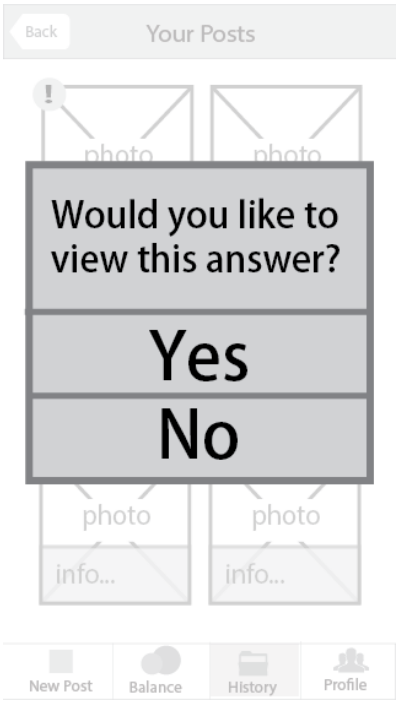



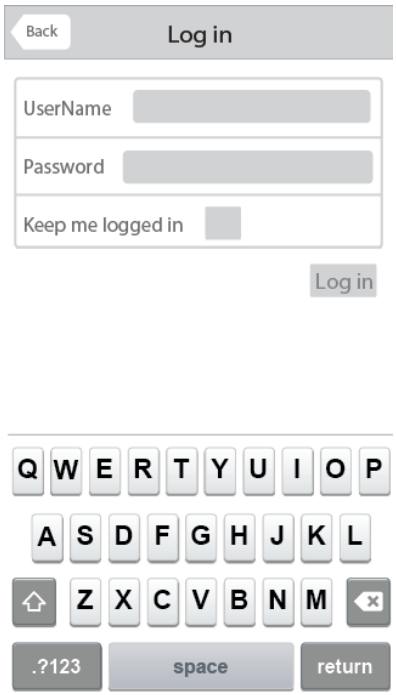


Back
Profile
Log out
Name
Address
Email
Phone Number
Password
Confirm Password
Save
New Post
Balance
History
Profile


Back
Balance
Card
Address
Email
Payed out
Income
Total Balance
Save
New Post
Balance
History
Profile

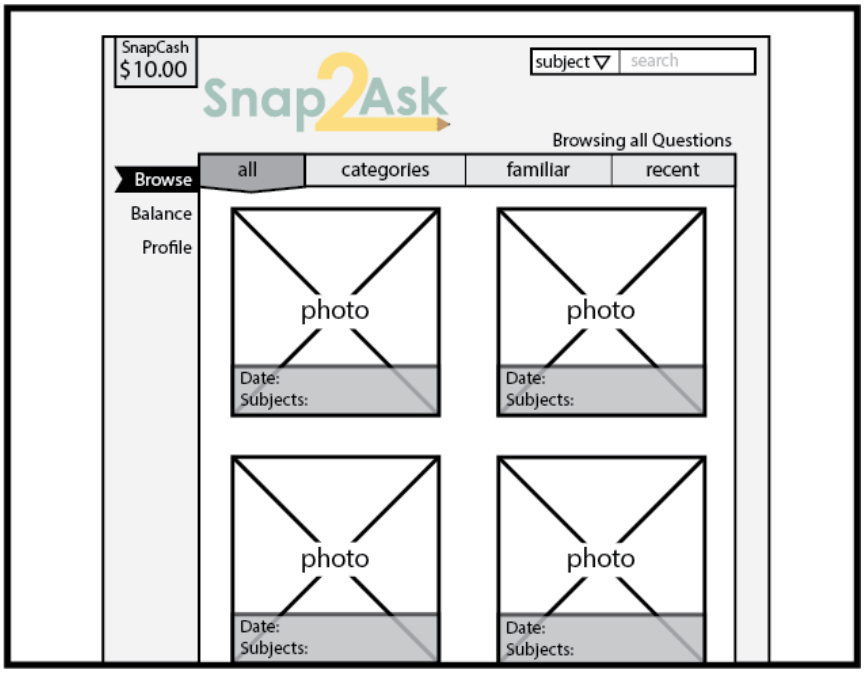

Back
Your Posts
photo
info...
New Post
Balance
History
Profile

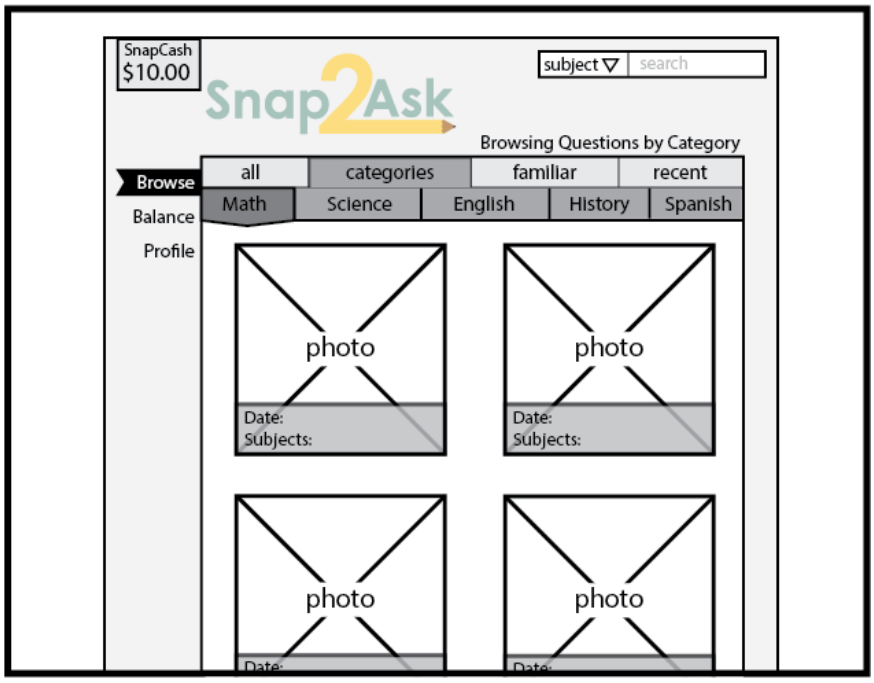

Back
Your Posts
Would you like to view this answer?
Yes
No
New Post
Balance
History
Profile

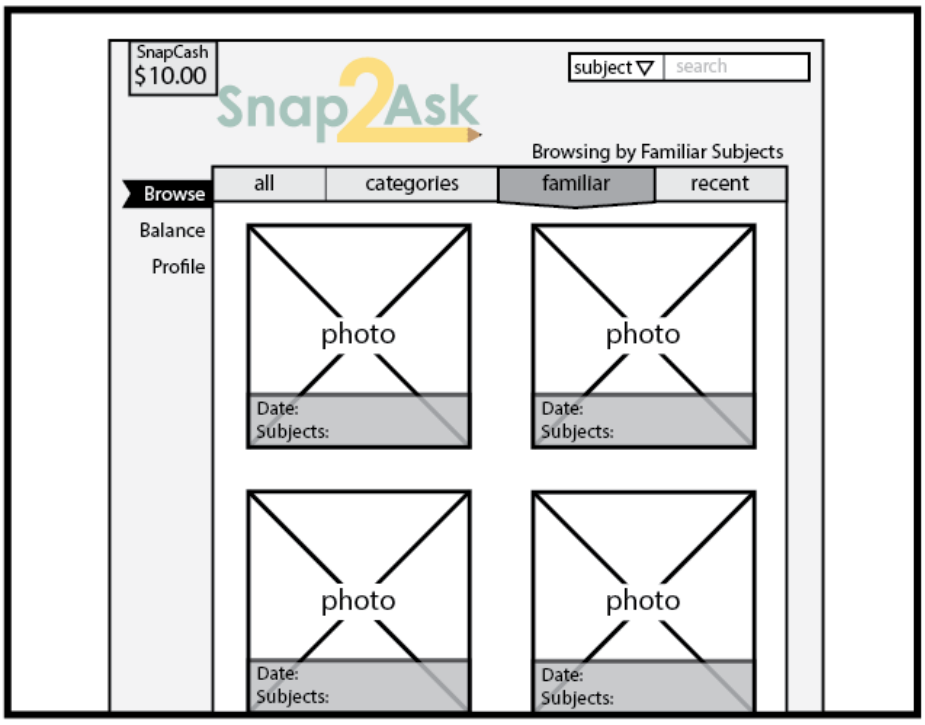

Back
Your Post
PHOTO
username
data about user and answer
answer
New Post
Balance
History
Profile

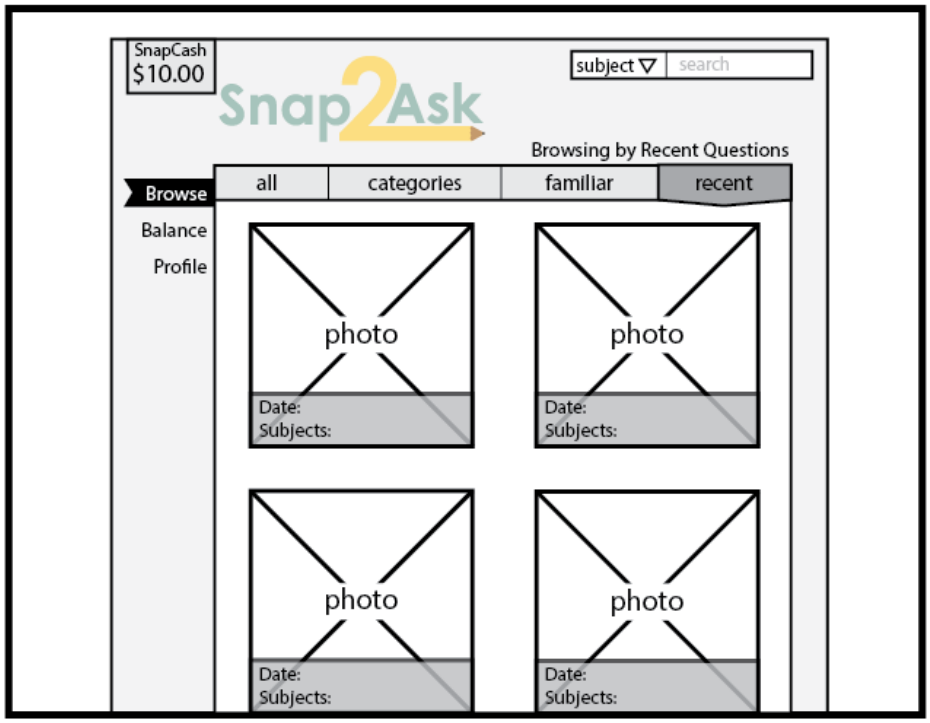

Back
Log in
UserName
Password
Keep me logged in
Log in
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
.?123
space
return

# WEB PAPER PROTOTYPES

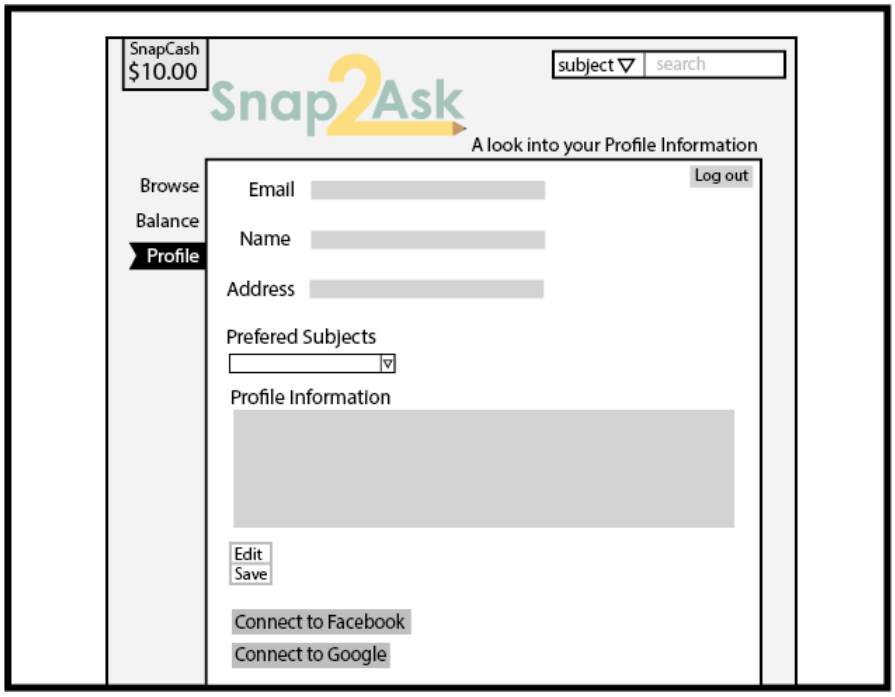



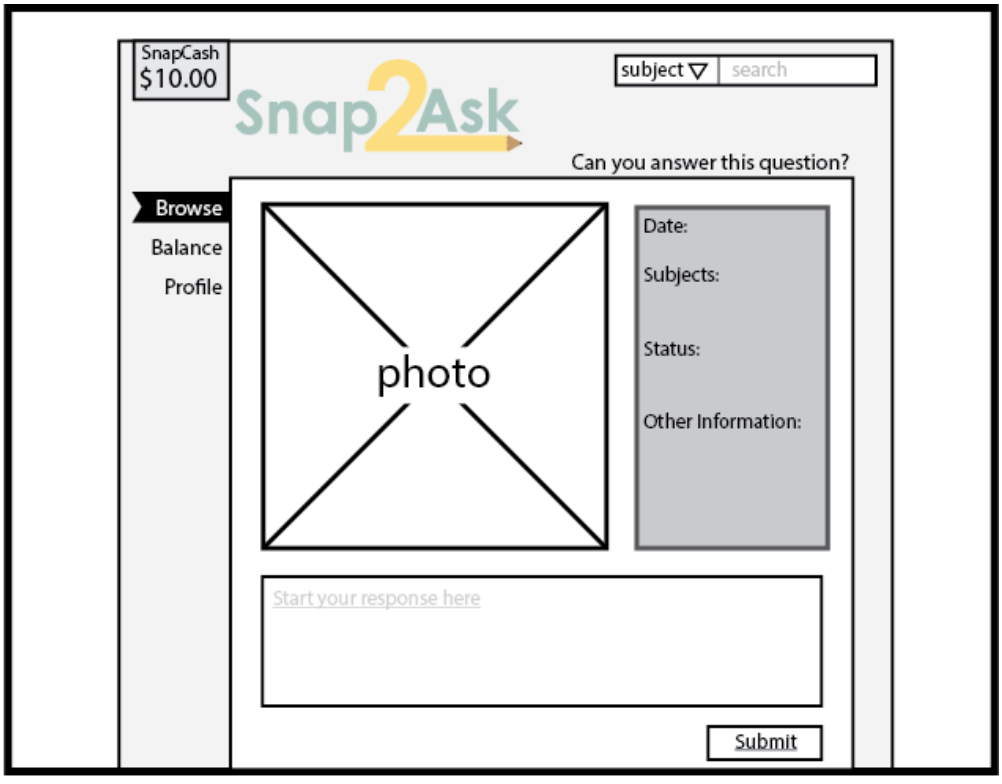

SnapCash
$10.00
Snap2Ask
subject
search
Browsing Questions by Category
Browse
Balance
Profile
all
categories
familiar
recent
Math
Science
English
History
Spanish
photo
Date:
Subjects:
photo
Date:
Subjects:
photo
photo


SnapCash
$10.00
Snap2Ask
subject
search
Browsing by Familiar Subjects
Browse
Balance
Profile
all
categories
familiar
recent
photo
Date:
Subjects:
photo
Date:
Subjects:
photo
Date:
Subjects:
photo
Date:
Subjects:


SnapCash
$10.00
Snap2Ask
subject
search
Browsing by Recent Questions
Browse
Balance
Profile
all
categories
familiar
recent
photo
Date:
Subjects:
photo
Date:
Subjects:
photo
Date:
Subjects:
photo
Date:
Subjects:


SnapCash
$10.00
Snap2Ask
subject
search
A look into your account balance
Browse
Balance
Profile
SnapCash
Address
Email
Total Balance
Questions or Concerns?
Call us at 123-456-7890
E-mail us at support@snap2ask.com
Visit our Help Page


SnapCash
$10.00
Snap2Ask
subject
search
A look into your Profile Information
Log out
Browse
Balance
Profile
Email
Name
Address
Prefered Subjects
Profile Information
Edit
Save
Connect to Facebook
Connect to Google


SnapCash
$10.00
Snap2Ask
subject
search
Can you answer this question?
Browse
Balance
Profile
photo
Date:
Subjects:
Status:
Other Information:
Start your response here
Submit

# USER PROFILES

| Name | Simar B. | Dr. L. Resendis | Chris A. |
|---|---|---|---|
| **Probable Relation to Snap2Ask** | Student Asking Questions | Professor answering Questions | Customer Service Employee |
| **Age** | 16 | 35 | 20 |
| **Gender** | Male | Male | Female |
| **Salary** | $2,400/yr | $100,000/yr | TBA |
| **Education** | 11th grade | Doctor/Professor | SMU |
| **Relevant Subjects** | H Physics, AP US History | Computer Science | Computer Science |
| **Tutoring Budget/Expected Pay** | $-20 a month | $+100 a month | $+60 a month |
| **Tech Experience** | Pro eBay-er, Tutor.com, Yahoo Answers | Dr. of Computer Science | Intern for Amazon.com, TA for Computer Labs at SMU |

Simar B.: He has really hard time in school. 11th grade is really hard especially with Honors Physics and AP US History on top of college placements exams such as the SAT and ACT. He does not that much time in his day with all the video games he has to play. He tried Skyping his friends with homework questions, but they would only give him an answer without any explanation. He is really good with technology. He has an Apple iPhone 5 Ghost Armor without any access to Android devices in his household. He loves his Windows 8 but prefers his MacBook Pro.

Chris A.: A graduate student at SMU. He works as a TA and grader for the computer science department, so he is fond of tutoring and experienced with managing a gradebook. Therefore, he is a good candidate to serve as an employee for Snap2Ask's customer service. He is always in search of paid positions that involve working on his MacBook executing simple tasks such as browsing member accounts and resetting their passwords.

Dr. L. Resendis: A Computer Science professor at SMU. Professor Resendis teaches many Computer Science courses at SMU and is very experienced in the field of technology. He is a good candidate for a tutor for Snap2Ask. With his knowledge from teaching classes, he will be able to help many students with their questions related to Computer Science.

**How often do you use the internet to solve homework questions?**

Simar B.: Googles every homework question he cannot figure out himself. Gets help from the Internet especially in US History because his teacher does not explain anything in his 1 hour a day class time, but he is given a lot of homework.

Dr. Resendis: I do not use the Internet that much except to look at publications by other professionals. I am a professor at SMU, so much of my focus goes into my career, but I would really be interested in having a place where my students can get professional help with their assignments. I give a lot of homework and programs, and not all my students can make it to my office hours to get their questions answered. A professional online resource would be VERY helpful!

Chris A.: I use search engines to look at code from other programmers that can help me with my own programs. I would like a system where I can get my code reviewed by professionals in the field. Snap2Ask should definitely have a college category. A lot of code is out there on the Internet, but there is no way to get help by professionals quickly if the code does not work for me.

**How would you feel about paying/getting when posting/receiving answers?**

Simar B.: I want tutors to answer my questions for free. I will only pay $20 a month maximum. I want a monthly subscription unlimited plan like Netflix as an option if I find I become a daily user. I want to be billed an insertion fee only when I receive an answer if I am on a per-question basis. I should be able to buy credits that never expire through your web site if I do not buy a monthly subscription.

Dr. Resendis: Provide an estimate of pay that the tutor will get before answering each question. Log all questions asked or answered and provide the total balance. For the first few questions, keep the funds pending so tutors can work to establish a good track

record. After a tutor has proven himself, allow for instant payment after an answer has been rated or been posted for 48 hours without being "discarded"

Chris A.: I want payment immediately after I help a customer. If my payment must be approved by my "boss" that is fine, but I hate when funds are pending for too long. I have no problem logging which inquiries and help I provided so my supervisors can assess my pay.

**What do you LIKE about Google, Yahoo Answers, and Tutor.com?**

Simar B.: Google and Yahoo Answers solve your answers quickly. No fees and no wait time. It's a community, so people just post questions and answers to COMMON PROBLEMS since they know they will help A LOT of people with their answers as well as get top rated on their answers. I like Tutor.com because I can chat one-on-one with QUALIFIED TUTORS. The tutors will try to help me with ANY question in their subject of expertise .Not every solution/explanation to every homework question can be found with a Google search or on Yahoo Answers. I can also select a favorite tutor and complete a survey to rate them.

Dr. Resendis: Provides examples of how to use basic programming techniques and equations to solve example situations or problems. Quick answers are good!

Chris A.: Quick and usually helpful answers. Correctness I must analyze on my own.

**What do you HATE about Google, Yahoo Answers, and Tutor.com?**

Simar B.: Tutor.com has long wait times, and tutors take forever to answer a simple question. They are more focused on making me "learn" rather than answer the question and let me learn at my own pace. I get the service through my library, so it is free for me, but only a small number of students can be logged it at one time. The site has a fixed time schedule, which means it is not 24/7 like Google or Yahoo Answers. I do not like Google or Yahoo Answers because amateurs could be answering and only starting an argument rather than coming to a decisive answer. There is no one to reliably validate the answers that are posted except by a trivial rating system. Sometimes answers with 0 rating are "voted best answer"

Dr. Resendis: Top rated answers could be outdated or incorrect. There is no way to distinguish between a qualified user and a random user. There is no motivation to post quality responses other than a "feedback" or "point" system.

Chris A.: Tutor.com? What is that? I never heard of it. Seems below college-level. Google and Yahoo answers do not seem to have professional maintenance. Just because I create an account on discussion forums and blogs does not mean I know how to use the account at first. A good customer service team is a must. I figure everything out at the end, but I should be asking and answering immediately after I create the account.

**Do you think it is a good idea only for tutors to answer questions?**

Simar B.: No because not all tutors are knowledgeable even though they pass the entrance exams. Students from the same school have more contexts of the questions, so I would rather go to our school's Facebook group. I might come to your web site as a last resort if only one answer is allowed at a time. I think having answers from several tutors is a good idea.

**Do you think our web site can compete with Yahoo Answers?**

Simar B.: "I do not like to discourage people, and I want them to go and reach for their goals. My motto is "Shoot for the moon, if you miss you'll land among the starts," so I actually encourage a new competitor and let's see where it goes.

**Would you prefer upload/type the question through an online form or upload through our Android app?**

Simar: I like the idea of mobile uploads, but I do not like that is only an Android app. You should have a mobile online form possibly for us iOS users.

**How do you like getting paid to answered questions? What pay do you expect per answer?**

Simar: I love getting paid for easy stuff online. Level 1 20 cents a question. Level 2 25 cents.

Dr. Resendis: College level questions should get $1 per answer! Based on the answer time relative to the posting time, the pay should be more. It should be like an answer "auction."

Chris A.: I will only answer e-mail or phone inquiries. $5 for each successful resolution. I might answer some computer science questions, but I want $1 per answer.

**How do you think we should resolve negative feedback?**

Simar B.: Every time an answerer gets a negative feedback, his pay should be deducted.

Dr. Resendis: Give a break to new tutors. Maybe give them a withdrawal or payment limit until they have achieved a certain status. Give all Level 1 tutors the same pay. As a certain feedback is reached, add to the pay scale!

Chris A.: I will not delete or dispute negative feedback. I will just watch the ratings, and warn the tutor that his/her negative track record could lead to tutor account deactivation or deletion.

**How long do you think our billing/payment cycles should be?**

Simar B.: Keep funds pending until the end of the month when the billing/payment cycle ends.

**What type of feedback system do you suggest?**

Simar B.: Each answer should be rated individually, but the student should know the tutor's username to add to a "block" list in case a bad answer is received.

Dr. Resendis: "Blocking" of tutors should not be allowed. A student continuously leaving negative feedback should be warned. Both tutor ratings and answer ratings are good, but allow tutors with higher ratings to get the first shot at answering questions! Let students rate.

Chris A.: Let a formula rate the questions and answers. Add multiple performance metrics. As a customer service agent, I should easily be able to gauge the accuracy of the tutor's pay to the rating of their answers.

# TASK ANALYSIS

## What goals do users want to achieve by using Snap-2-Ask?

Snap-2-Ask is designed to simplify the process of getting answers to homework problems. By quickly snapping a picture, students are able to get a response to their questions from our any of our certified tutors.

## What set of human tasks is the application intended to support?

Snap-2-Ask is intended to replace hiring tutors when you only have a few specific questions to ask. Snap-2-Ask makes asking questions very simple and easy.

## Which tasks are common, and which ones are rare?

The most common tasks are asking a question and answering a question. The rarest tasks are to delete your account and to sign out of your account.

## Which tasks are most important, and which ones are least important?

The most important tasks are those, which revolve around questions. Thus, the most important tasks are to ask questions, and to answer a question. Additionally, it is also important to allow the creation of an account and the logging in of an account. This is required so users can ask questions.

## What are the steps to asking a question and answering a question?

The process of asking a question is remarkably simple. A student takes a picture of the question using their iPhone. They give the question a description and a category. Finally, they upload it to the Snap-2-Ask servers using the Snap-2-Ask iPhone application. They will get a push notification as soon as their question is answered.

The process of answering a question is also simple. A tutor will browse all questions related to their specialized subjects. When they see a question they can answer, they just submit their answer in textual form and can continue answering other questions.

## Where does the information for each task come from?

The information about questions comes from the student and their iPhone application. Additionally, any information about answers comes directly from the tutor via the web interface.

## How is the information from each task used?

The information that the student publishes using the iPhone app is added to a database. This information is then dynamically displayed to all tutors on the web interface. This allows tutors to find questions and be able to answer them.

Additionally, any answer that the tutors submit gets added to the database. Also, a notice is sent directly to the user's phone, notifying them that they received an answer to one of their questions.

## Which people perform each task?

Students, who have questions, use the iPhone application to submit their question. On the other hand, Tutors answer all the questions. Thus each user has specific tasks that they can perform

## What tools are used to do each task?

The students ask questions via the iPhone application. The tutors answer questions via the web interface.

## What problems do people have performing each task?

Tutors might not know the answer to some questions; however, since there will be a really large number of tutors looking for questions to answer, this shouldn't be a problem.

## What sorts of mistakes are common? What causes them?

The answers that a tutor gives to a question might not be 100% accurate. Therefore, if a student finds an answer to be unacceptable, they can reject the answer and open their question back up for more answers.

## What terminology do people who do these tasks, use?

Question: a question asked by the student. This is usually a homework problem.

Answer: a response to a question submitted by a tutor. This response is sent to the user who asked the question.

# OBJECT/ACTION ANALYSIS AND MATRIX

| | Create | Edit | Delete | View | Search | Send/Submit | Pass | Fail | Rate |
|---|---|---|---|---|---|---|---|---|---|
| **Account** | WM | WM | WM | WM | | | | | |
| **Question** | M | M | M | M | W | M | | | |
| **Answer** | W | W | M | M | WM | W | | | M |
| **Test** | | W | W | W | | W | W | W | |

W = Web Interface

M = Mobile iOS Interface

# SOFTWARE LEXICON

**Account:** account based on email and password that only exists on the snap2ask database.

**Tutor:** a person who answers the questions or has the ability to answer them.

**Validated:** an adjective for a tutor who has been certified by the website to answer question in their realm of knowledge.

**Student:** a user who asks a question.

**Facebook Account:** an account linked to by Facebook.

**Google Account:** an account linked to by Google.

**Question:** a picture, with its certain subjects, that asks for a tutor to answer it.

**Subject:** a tag on a question that pertains to the questions subject matter.

**History:** the previous questions one has posted, including the answers for those questions.

**Profile:** a user's information (both tutor and student information), which pertains to sign up and identification.

**Balance:** the pending amount of SnapCash in a user's account.

**Answer:** a picture or text based response that answers a user posted question.

**SnapCash:** pseudo money in the database that users will be able to use and transfer based on questions answered and questions posted.

**Answer Rating:** determination of how well an answer comprehensively fulfills a question.

**Tutor Rating:** a calculation that determines the validity and dependency of the answers submitted by a tutor.

# USABILITY TEST OUTLINE

**3 USERS WILL BE TESTED:** 1 high school student, 1 college student, and 1 professor.

All high school and college participants will try our mobile app and web site because they are of age to be a paid tutor while they have questions of their own from school.

Doctor/Professor Resends will try our web site during his office hours since he likes to tutor students on computer science during his spare time.

Professor Raley will try our products as a professional usability tester when it comes to mobile app and web site design.

2 high school students and 1 college student will be selected from Vipul's Skype contacts so that we can observe only what they are doing in one session.

In a second session, we will ask the users to think allowed in order to better understand usability issues. We will ask these Skype users to retry the software throughout the development cycle as newer features are added.

Other group members can perform the "Hallway Testing" routine by showing our product to a random classmate they find in the hallway, residence hall, or Hughes-Trigg Student Center.

We will develop a survey for our testers to complete after each evaluation.

These usability tests will primarily focus on the structure and functionality of our web site as far as relevance to our users.

The goal should be to fix all bugs and missing functionality that our users find as well as task failures because our users find a "dead end" in our software where they have no clue how to proceed.

For the initial testing, we will focus on our users being able to successfully, sign up, login, post a question, and view account statistics. At this stage, users should be able to log in through their Facebook or Google Accounts.

Afterward, usability testing will be focused on allowing members to pay and get paid as well as feedback on the "looks" of our web site.

# USABILITY REPORT

## Usability Test with Dr. L. Resendis:

**MINOR PROBLEM:** Why does a balance page appear when account is created? Maybe change title to "Balance and Payment."

**MISSING FUNCTIONALITY &TASK FAILURE:** User enters credit card and profile information and leaves mobile app screen thinking info was saved or submitted However, the reality is that no "save" or "submit" feature was actually provided.

**MISSING FUNCTIONALITY:** No logout feature once user has viewed answer or submitted question.

**TASK FAILURE:** User think iPhone app has exited once hitting the home button, but this is not true. Also, user expects to be automatically logged out once exiting the program, but no log out feature exists. Make sure there is a clear "close" feature on our app so that user does not have to go their devices task manager to end the application.

**ANNOYING:** User has to click on specific question link in history to see if it has been answered rather than receiving a notification as a header in the history

**MISSING FUNCTIONALITY:** No button provided to activate tutor account.

## Usability Test with student Simar B.

**MISSING FUNCTIONALITY:** Add description field to our mobile app beneath picture

**MISSING FUNCTIONALITY:** give feedback through app!

**MISSING FUNCTIONALITY:** close app and log out

## Usability test with Chris A.

**MISSING FUNCTIONALITY:** In tutor profile page, tutor is only allowed to select one preferred subject. Rather than having a drop-down, we should have a "mark all that apply" feature.

**MINOR PROBLEM:** The iPhone app Profile view/edit page has a confirm password field. It is better to have a change password link.

**MINOR PROBLEM:** The "Income" and "Payed out" features of Balance should be removed or replaced. User should only see Card and Balance.

**ANNOYING:** On the "Success Your Question has been posted!" page, there should either be a link to view history or an arrow pointing to the "History" button on the bottom task bar.

**TASK FAILURE:** User thinks he provided one subject associated with his question in "Choose Subjects" when he actually associated his questions with 4 unrelated subjects. Let user first select from English, Math, Science, Social Studies, Other. Once user selects broad category let them scroll subcategories.

**MISSING FUNCTIONALITY:** If tutor's answer has a lot of "description" text student should be able to zoom in on the text and easily read the text answer. If a zoom in feature is not provided, there should be a scrolling feature. Mobile student should also be able to save and enlarge picture answers.

**ANNOYING:** The Web home page has too much text. There should have an "I am a:" heading followed by a Tutor button and a Student button. Next should be our video and footer. A lot of text and too many wordy options, which is very annoying to the user, so please edit.  Compatibility does not need to be displayed on our web site app download page since the App Store should take care of that for us.