Marisa Steans

Android CM Intern @ Udacity

Project: Office Hours App

Overview

June 2015

Table of Contents

1. Background, Purpose and Outcomes
2. Project Details
   1. Project Schedule
   2. General Overview of the Project
      1. Software Requirements
   3. Integrating the Project into Office Hours
3. Implementation Details
   1. UI Layout (independent web-app version)
   2. Feature Notes
   3. Implementation Notes

(1) **Background, Purpose and Outcomes**

The course managers, during an office hours session, often have to wait for students to submit their questions in the chat with supporting text or a forum post. The purpose of this project would be to streamline the question asking process for the course managers and help students learn how to provide all the information necessary to solve a problem. For course managers, they will have a list of submitted problems to move onto inside a web-app or Google Hangout app. This will help them know when to give general tips to the students or know when to move onto another question. For students, this will condition them to post more properly informative forum posts by requiring a post for each question submitted during office hours. The course managers can give feedback on what the post is missing and this will lead to more clean and hopefully easier to understand problems.

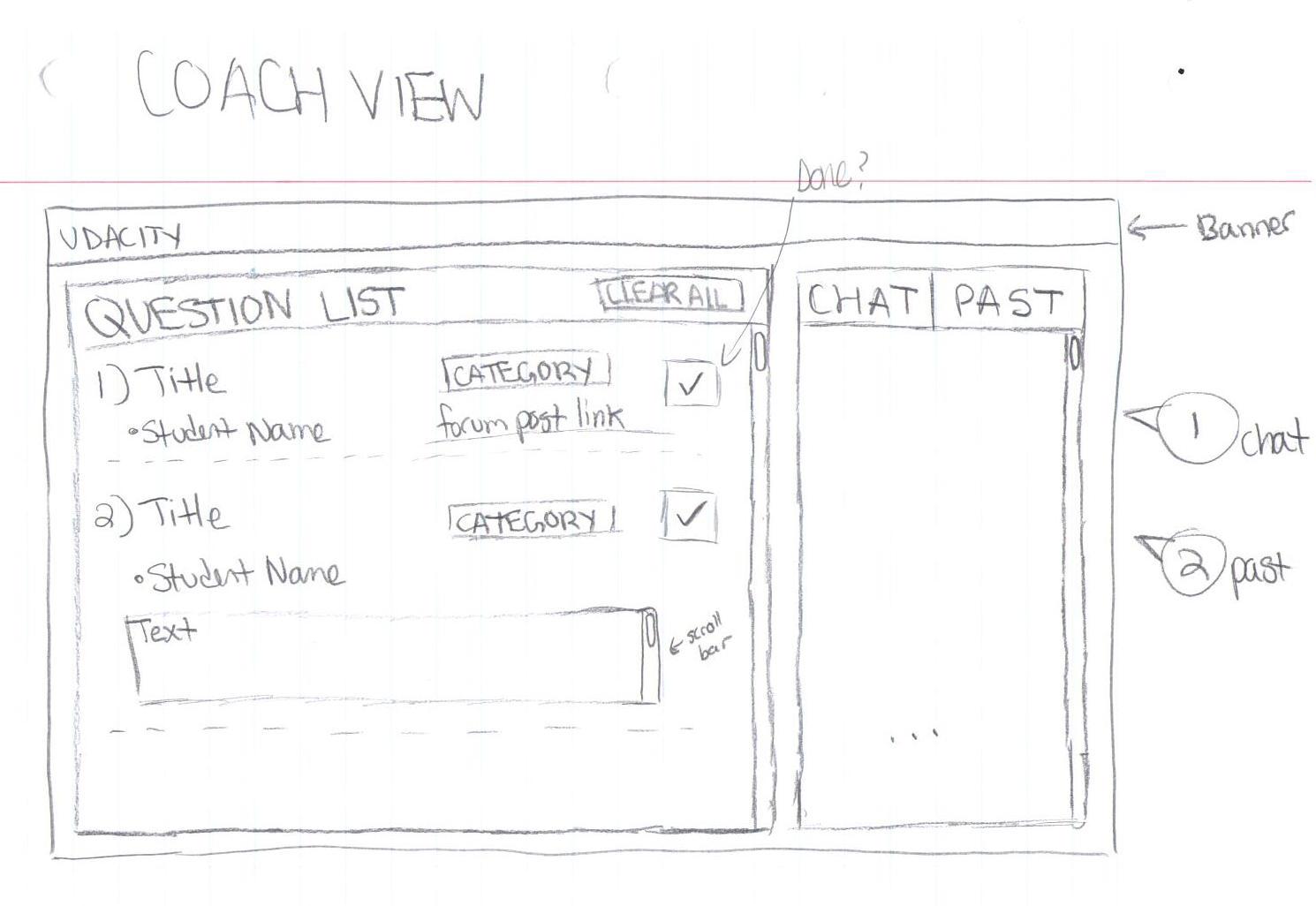
As for my background, I am a computer science undergraduate student and rising senior. During my tenure at the National Center for Supercomputing Applications I worked on an open-source web-app called Medici for the Innovative Software and Data Analysis Division. I also have some experience making websites in some courses as well as for personal purposes. I have the confidence to complete this web-app given some training and assistance from helpful Udacians.

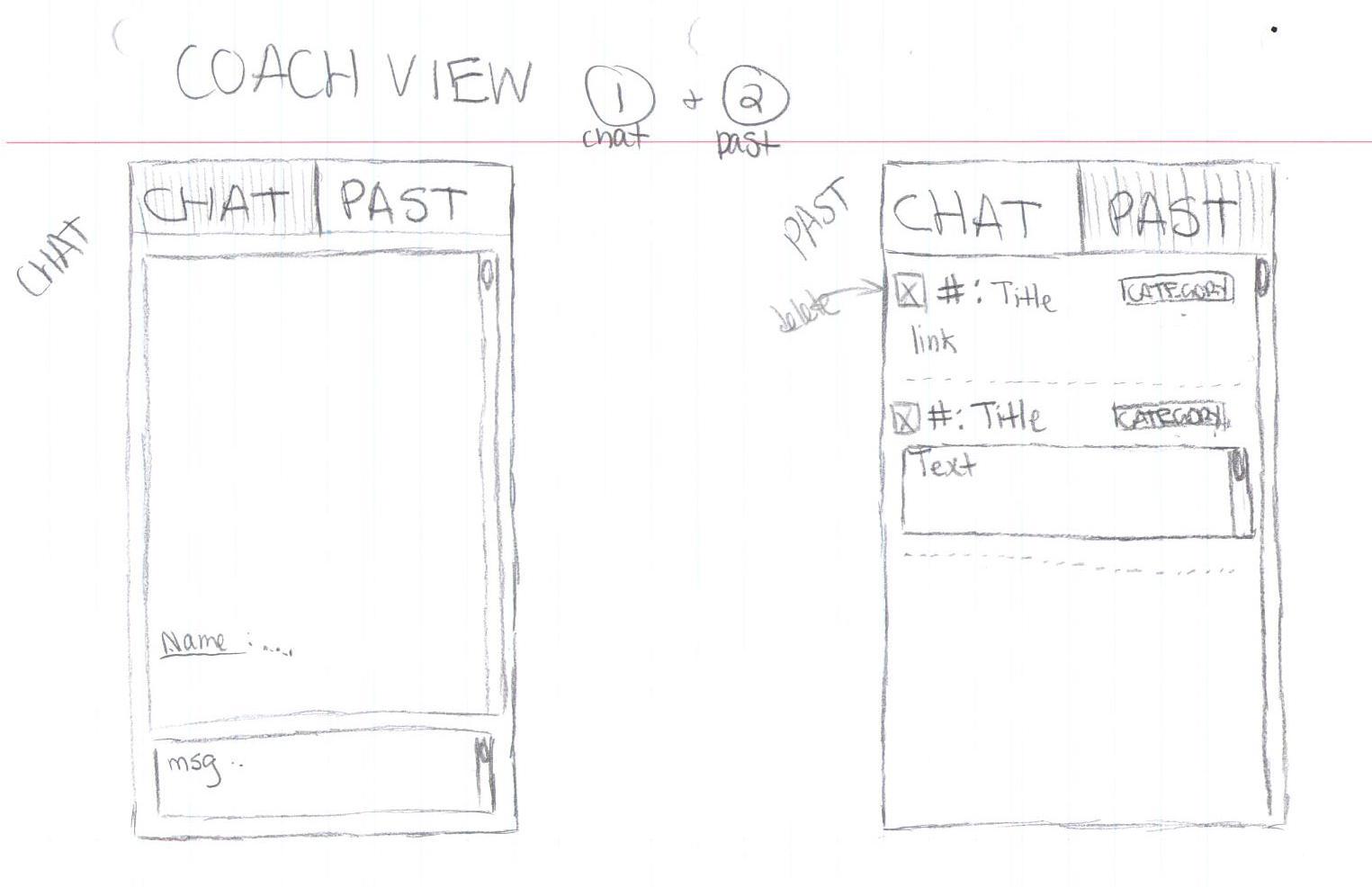
**(2) Project Details**

1. Project Schedule:
   * June 23rd – Determine necessary technology to complete project
   * July 13th – Check-In with Mentor about progress
   * July 30th – August 3rd – Complete Project and Prepare Presentation
   * August 5th – Presentation
2. General Overview of the Project
   * Separate Web-app or Google Hangouts Extension/App?
     1. Google Hangouts:
        1. Google Hangouts has an API for creating extensions and apps.
     2. Separate Web-app:
        1. There might be a change to the video service used for Office Hours. It is better to prepare to make this available in case Udacity switches from Google Hangouts.
   * Software Requirements
     1. Google App Engine, Python, HTML/CSS, JavaScript, ….
3. Integrating the Project into Office Hours
   * Process for Students:
     1. If students have a question: They will be directed to the web-app session upon entering office hours. There they can fill out the form to submit a question for the course managers to address.
     2. If students do not have a question: When there are no questions to address in the web-app the course managers can address more general questions via chat/voice.
   * Process for Course Managers:
     1. They start office hours and direct students to the app for question submission. When opening the details to a question there will be a forum post to review. Go through the forum post and address the student on what they did well and what needs clarification in their forum post. Answer the question if it is able to be answered then or request a few changes and ask the student to submit the question with added info. Then move onto the next question.

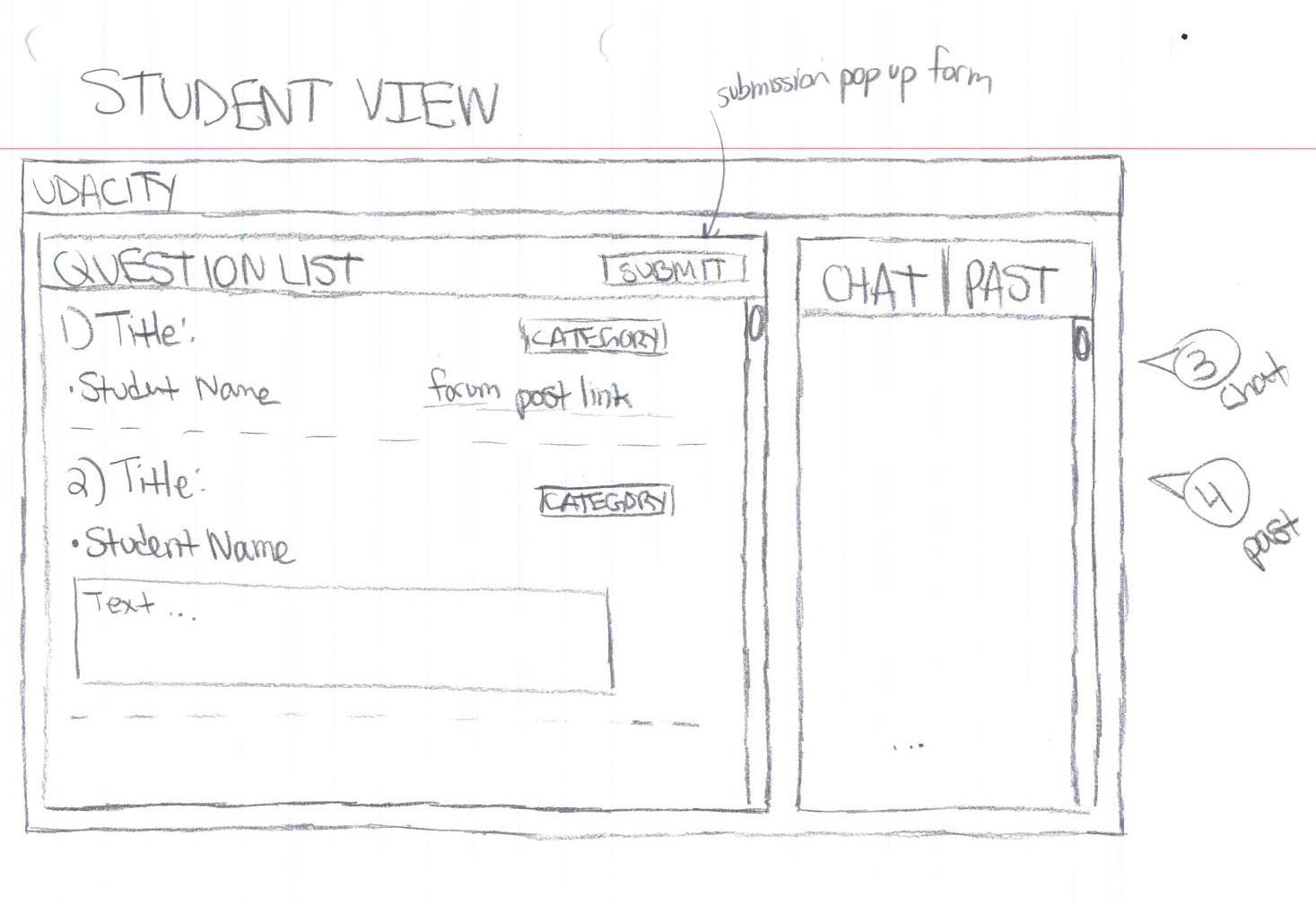
**(3) Implementation Details**

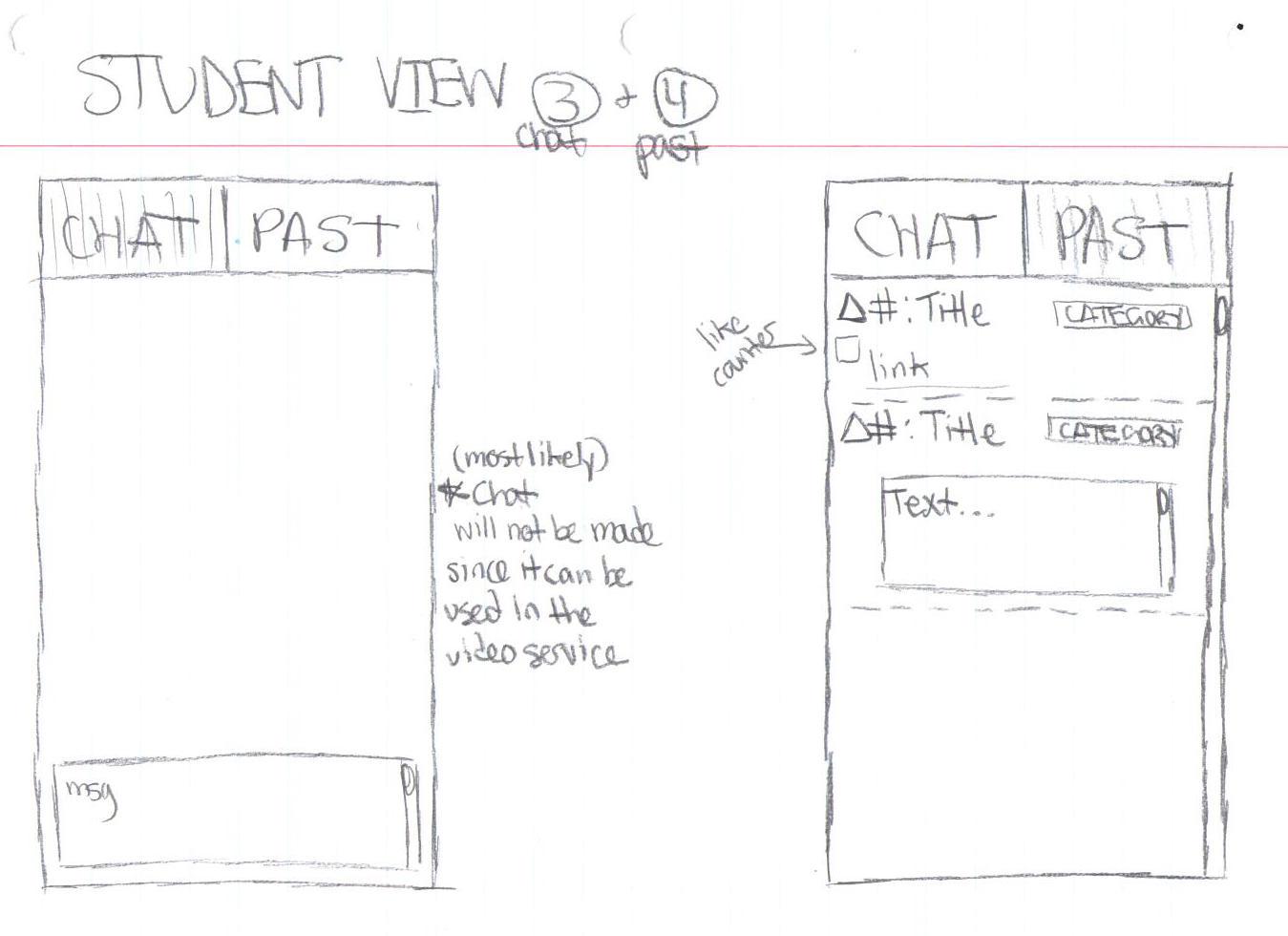
1. UI Layout Outline
   1. General Layout
      1. Possible Pseudo-Login (since they have to login with a Google account anyways; if their Udacity email is different from the Google email then it might be necessary)
      2. Certain elements in the view can be determined by the type of user.
         1. Have some “display:none” and some “display:inline”
      3. Coach View



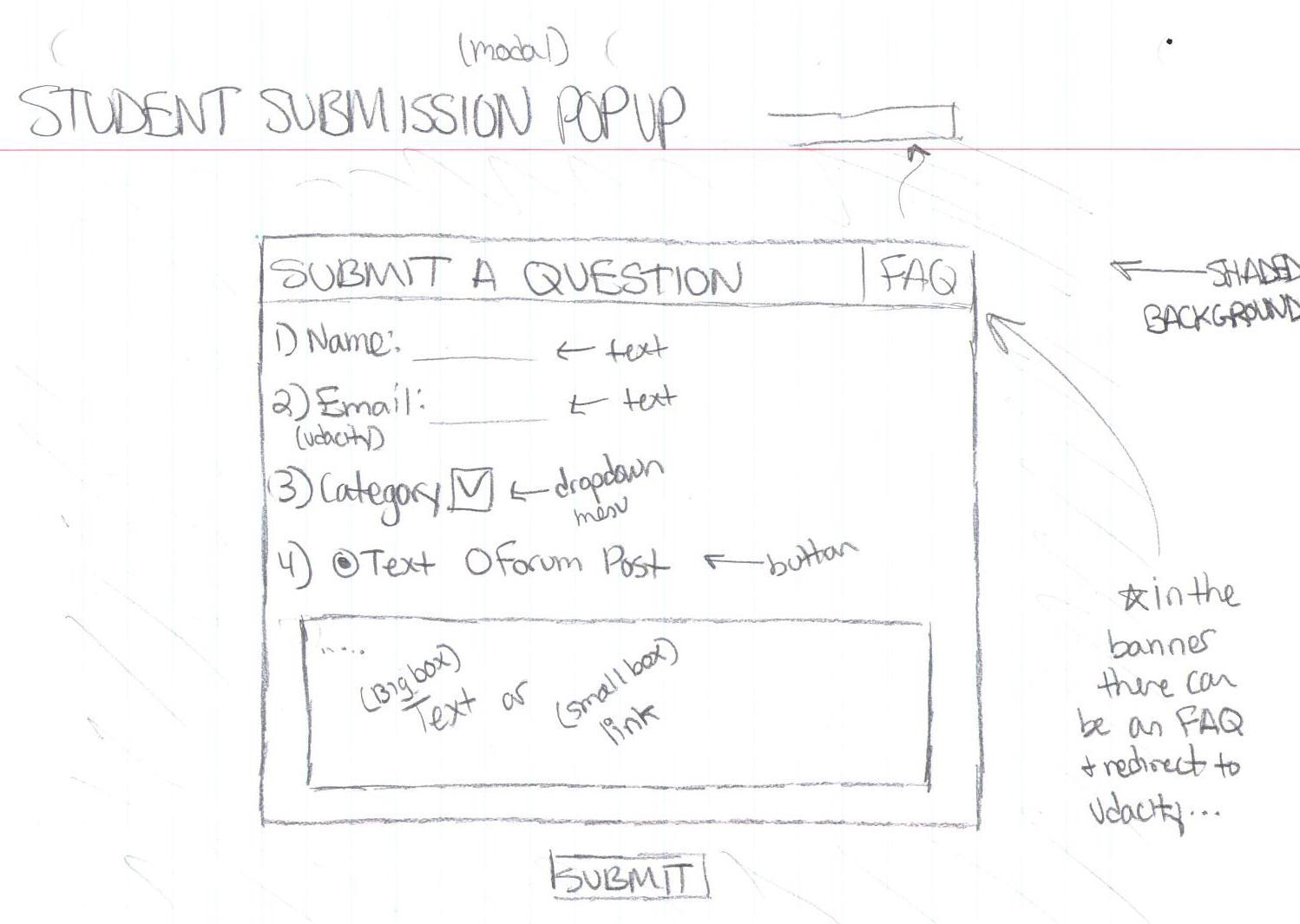


* + 1. Student View





* + - 1. Submit a Question (student view popup window)



1. Feature Notes
   1. Chat is to be made from previous code if necessary but otherwise the chat from the video service will be used (since two chats are confusing).
   2. How to share the webapp with students?
      1. The link to the app will be known by students and redirected to on the office hours schedule. The link will always be the same therefore the question submissions will have to be taken care of previous to the starting of the event. We can also take down the app when it is not in use.
   3. What to do with unaddressed questions made before Office Hours start?
      1. Strategy 1: Clear all since they have not been addressed yet
   4. Limit students to re-submitting 1 question from Past Questions and submitting a new question to Current Questions.
2. Implementation Notes
   1. GAE Entities Needed
      1. Total Question Container
         1. curr\_q and past\_q
            * Contents (columns)

Integer upvotes

Link q\_link OR Text String q\_description

Text String student\_name

Text String q\_title

Category q\_category

Date date\_submitted

Boolean current\_q (if true, curr\_q; if false, past\_q)

id

* + - * + Functions

construct\_q

Function: makes new question and assigns values of question with given data

upvotes starts at 0; category is chosen from dropdown

called when a question is submitted

move\_to\_past

function: move a curr\_q to the past\_q table

move\_to\_curr

function: move a past\_q to the curr\_q table

delete\_q

function: remove question from the table of questions and sends them to the past (if curr\_q) or permanently deletes them (if past\_q)

called when the X button is pressed (admin-only)

increase\_upvote

function: increase the question’s upvote count and reload the page

called when the upvote of a question is pushed

reask\_q

function: moves past\_q to curr\_q window

called when student clicks Re-Ask button in past questions window