Class.ly: Helping Students Collaborate

Data Analysis & Prototyping Plan

CS 3892 Human-Computer Interaction
Vanderbilt University Fall 2015

By: Parker Klein & Mitchell Masia

I pledge my honor that I have neither given nor received aid on this work.
Mitchell J. Masia & Parker A. Klein

I. Introduction

Class.ly is a system meant to reduce the inherent complexity in academic collaboration among Vanderbilt undergraduate students. Quite often, students and groups of students miss important class details leading to confusion, extraneous communication between students and professors, and stress for all involved parties. When working on group projects, one of the initial hurdles to overcome is getting to know one's group members and acquire their contact information- this is the originating painpoint addressed by Class.ly. Ideally, Class.ly will be used by students, professors, and administrators, each of whom is presented a custom interface with pertinent information to them.

Upon a student user logging into the system, Class.ly integrates with the Vanderbilt Your Enrollment System (YES) to gather their course enrollment information. The system uses this information to automatically create a chatroom and meetup scheduler with other students enrolled in the course. This system removes many of the avoidable and annoying parts of communicating with classmates and groups because no contact information exchange is necessary. Contrarily, faculty users of Class.ly will presented this same chatroom interface as students as well as an interface to gather statistics about student collaboration on a per-class basis and aggregate basis. A system of this nature is not difficult to implement programmatically; however, the key to success is to create a large user base such that communication via the application reaches as wide an audience as possible. This document outlines initial efforts to qualitatively expose the value Class.ly would create for Vanderbilt undergrads.

II. System Stakeholders

There are three groups with large vested interest in an implementation of the Class.ly system. Vanderbilt undergraduate students, professors, and registrar personnel could all gain enormous value from a large network of system users.

A. Vanderbilt Undergraduate Students

Vanderbilt undergraduate students have a potentially huge value-add from Class.ly due to the immediate network with their classmates and group members. Instead of spending valuable time collecting contact information or finding a time to meet, students will have more time for friendship, video games, and the like. This stakeholder group is likely to be the most active, frequent users. They act primarily on their mobile devices, and must be able to use the application in less-than-ideal conditions in high-stress, time-constrained situations. This leads the simplicity in the Class.ly interface, brilliant use of color, and simple navigation.

B. Vanderbilt Professors

Vanderbilt professors have much to gain from their students using Class.ly as well. Instead of being bombarded by emails or struggling with the email-like communication interface in Blackboard, professors will immediately have a secure instant-message texting-like interface to communicate with their students. Professors are only privy to the chatrooms of their specific courses. This is a group of less frequent users who divide their usage between mobile and desktop Class.ly interfaces. They act in far less time-constrained situations, but are still offered the same seamless, easy-to-navigate application.

C. Vanderbilt Registrar Personnel

Vanderbilt Registrar personnel have substantially less vested stake in the capabilities of Class.ly but are much more concerned about system implementation and the data the application can uncover and visualize. Protecting students' academic data is a serious business. That's why one of the main concerns of the Class.ly developers is data privacy and security. With government regulations protecting the rights of students, it is essential that Class.ly is thoroughly penetration tests to ensure leaks are not abound. Second, the registrar personnel can benefit from collaboration data acquired via the application. On an aggregate level, administrators can view statistics about student collaboration, which can be advertised, and act as a huge draw for socially-inclined prospective students.

III. Ideal Implementation

The ideal implementation of Class.ly provides customized interfaces to each user role (reflective of key stakeholders), is strongly-secure, fault-tolerant, integrated with Vanderbilt YES, and integrated with native and 3rd-party calendar services. This backend functionality is paired with a simple, yet elegant user interface which abstracts the backend services into an instant-message-like chatroom, custom chatrooms, a "smart" meetup scheduler, and easy way to visualize and export collaboration statistics.

IV. Initial Implementation

Due to time and resource constraints, the Class.ly developers are implementing a very small subset of these features in the version 0.0.1 high-fidelity prototype release for the Human-Compute Interaction course. First and foremost, only the interface for the "student" role will be implemented. This initial implementation will not be integrated with Vanderbilt YES. Developers will use a non-relational database to mock up fake student, course, chat, and meetup data. This also takes care of the data privacy and liability concern as the system will not contain real, protected student data. Class.ly will also not immediately integrate with any 3rd-party calendar systems, leading to a less “smart” meetup scheduler. The chatroom interface will be fully implemented; however, the backend driving the functionality will be extremely limited. The purpose of this implementation is not to have a fully-functioning system; however, it is to act as a proof of concept, collect usability testing data, and demonstrate human-computer interaction principles in the design and use of the user interface.

V. Data Collection

Due to the limited scope of implementation of Class.ly for this course, the faculty stakeholders were foregone for data collection and analysis as their primary interfaces will not be implemented. This allowed the data collection focus to be the key stakeholder: vanderbilt undergraduate students. The chosen methods of data collection were a short questionnaire and individual interviews. All participants were Vanderbilt undergraduate students and before collection began, they were informed that identities associated with their answers would be kept anonymous and they were free to stop the interviews at any time.

The first data collection method was a short questionnaire which focused on the “Functional” requirements outlined in the Class.ly Requirements Documents. For reference, the “Functional” requirement include:

1. Automatically creating a chatroom
2. Automatically creating a meetup scheduler
3. A side-drawer for high-level navigation
4. A tabbed navigation panel
5. Large selection areas for buttons
6. Ability to opt out of the application
7. Creating custom chatrooms

The questionnaire was supervised by a creator of Class.ly and participants in the survey were informed as to the purpose and functionality of each requirement. The questionnaire asked individuals to rate the importance of each functionality to them on a scale 1-5, 5 being most important. The results of this survey were then normalized in order to figure out the relative importance weights of each functionality. The results of this questionnaire are shown in the “Analysis & Results” section below.

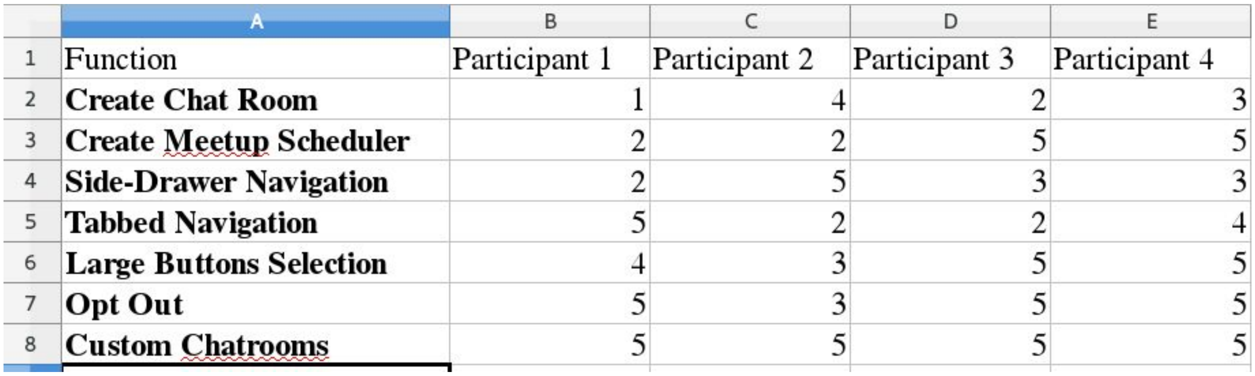
The other data collection technique employed in this study was the individual interview. In these sessions, participants were asked several questions, and then heard a pitch about the Class.ly application. The interviewer then asked for their gut reactions to the system, any concerns they may have, and whether they would use the application. This was a far more qualitative measure of excitement about the potential system development. Although this method is purely qualitative, it contains one of the most important data points gleaned during all of the collection: whether the interviewee would use the application. Because the value in Class.ly lies in a large user base for widespread communication and statistical analysis, finding out if students would use the application is extremely important to determine before building fully-fledged implementation.Qualitiative results of these interviews is explained in the “Analysis & Results” section below.

VI. Analysis & Results

A. Questionnaire

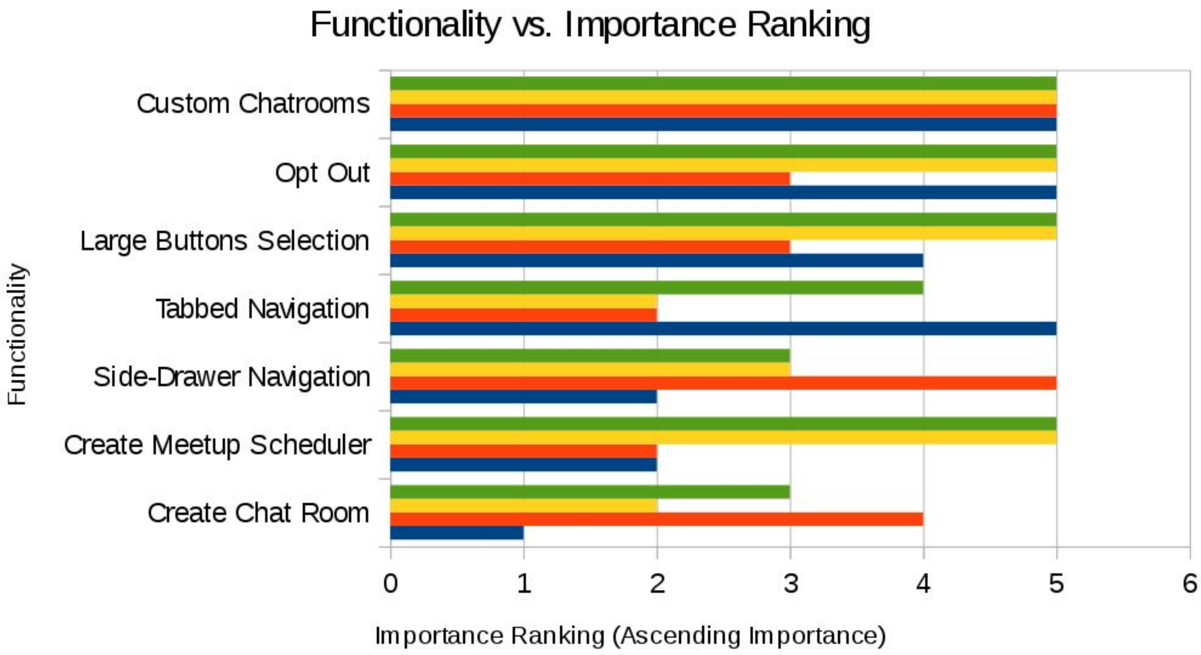
The questionnaire was administered to four Vanderbilt undergraduate students. Each proceeding was entirely isolated from other participants such that no answers were influenced by those of others, resulting in confounding results.

The initial questionnaire results tabulated in spreadsheet format are shown below:

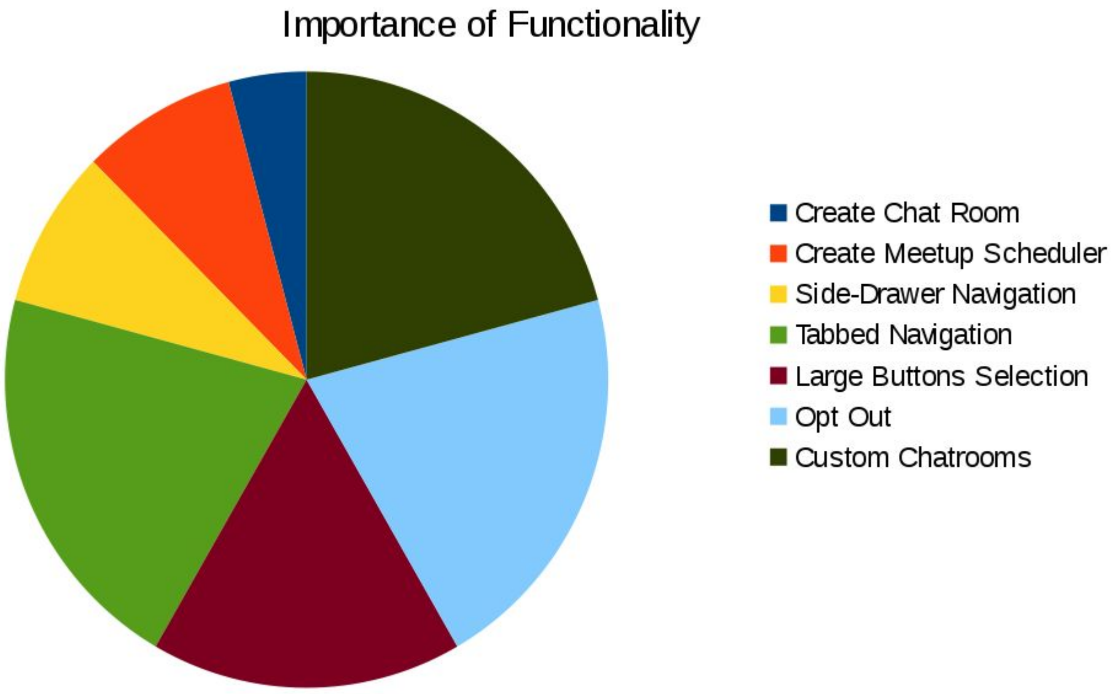


	A	B	C	D	E
1	Function	Participant 1	Participant 2	Participant 3	Participant 4
2	**Create Chat Room**	1	4	2	3
3	**Create Meetup Scheduler**	2	2	5	5
4	**Side-Drawer Navigation**	2	5	3	3
5	**Tabbed Navigation**	5	2	2	4
6	**Large Buttons Selection**	4	3	5	5
7	**Opt Out**	5	3	5	5
8	**Custom Chatrooms**	5	5	5	5

The answers, by participant, are shown in graphical format below:



The importance of each functionality, aggregate, is shown below:



As evident from the results above, the most important factors to the questionnaire participants are the ability to opt out of an individual class, create custom chatrooms, and that the application includes a tabbed-navigation interface. All of these rankings are very easy to trace back to originating stakeholder requirements. First, because a user can easily have upwards of 10 class and group chatrooms, the notifications from the application can feel like an all-out attack. This leads to four possible options for Class.ly designers: do not notify the user upon new chat messages, do nothing and possibly lose users, allow users to disable notifications on individual chats, and allow users to opt out of certain classes and groups all together. The option here is clear: allow users to enable and disable notifications on a class/group-by-group basis.

The next most important requirement according to users is the ability to create custom chatrooms. This is absolutely essential, even in the first implementation of Class.ly as it is the mechanism allowing communication with custom groups and class projects. The other large, important requirement by users is that the application use tabbed navigation for high-level system navigation. After consideration, the reason for this becomes clear. Because this group of users are prevalent smartphone users, they are used to the tab-style navigation interfaces in their favorite applications (e.g. Facebook, Instagram, YikYak, etc.). This helps them cleanly and quickly navigate from page to page, helping facilitate communication even under conditions of high-stress and limited time.

B. Individual Interviews

Contrary to the quantitative analysis performed with the questionnaire, six individual interviews were conducted, leading to some interesting new knowledge about potential users and their gut evaluations on the Class.ly concept. The interviews began not by introducing Class.ly, but by asking a series of exploratory questions. First, the interviewer leads with “What are the main problems you experience when working on group projects?” Frequent responses to this question include, but are not limited to: roles of individuals in the group (leaders vs. followers), **acquiring contact info**, **finding times to meet**, not being given time in class, not being on the same page, and **group members missing working sessions**. While Class.ly cannot help in balancing the roles of individual team members, it address three of the main concerns (in bold above) of interviewees. Naturally, Class.ly can help with contact information and scheduling meetings. Additionally, to account for missing persons, groups can insert notes and pictures from the meetup, which are stored by the application and can be reviewed by anyone invited to the meetup. This was very useful information to gather as it validated the Class.ly concept and identified a clear problem common amongst many Vanderbilt undergrads.

After gathering this input and hearing short pitch about Class.ly, students were asked for their concerns about the application. Most of these concerns reflect those from the questionnaire- mainly that there would be a constant stream of notifications berating users at all times. As stated, this problems, identified via data analysis, can be treated by allowing users to enable and disable notifications on a group-by-group basis. In this medium, most students had no problem with the application creating whole-class chatrooms, and thought that the meetup scheduler was equally, if not more, important than the chatroom functionality. This was surprising as it identifies finding meeting times as a larger consistent challenge than communication. From this discussion stemmed a few interesting ideas as well. First, multiple interviewees suggested that Class.ly integrate with native and 3rd-party calendar apps of all group members (not a trivial task), and actually suggest meeting times based on the respective schedules. This functionality will not implemented in the initial application; however, it will be taken into consideration for future work.

The last and probably most important information gathered from the interviews is whether or not students would actually use the Class.ly application. A few of the students refused to do so merely because of the plethora of systems already in use to facilitate professor-student communication (Blackboard, Piazza, etc.), and they do not want to deal with another foreign system. This was rather unexpected, but understandable response as many students are not particularly tech-savvy and may want to avoid using more and more networking systems. Contrarily, the majority of interviewees said that they would gladly use the Class.ly system, given a few constraints. First, a major requirement for them is the ability to “opt out,” or at least disable notifications on a group-by-group basis. This will be implemented in the initial version of Class.ly. Second, these participants state that the “smart” meetup scheduler provides immense value, and without it, the application is not much more useful than other communication methods. Clearly, this was valuable information to gather, as it means there are some completely necessary features that must be implemented in order to achieve widespread adoption among students, and increase the intrinsic value of Class.ly.

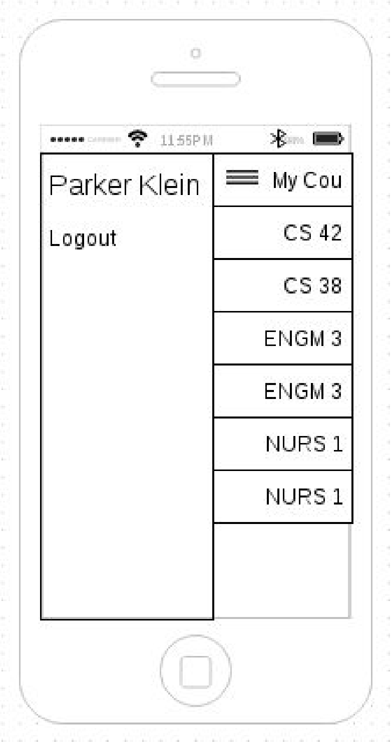
VII. Prototyping Plan

Scenarios for potential user interactions with Class.ly can be found in the requirements documents, available on our webpage.

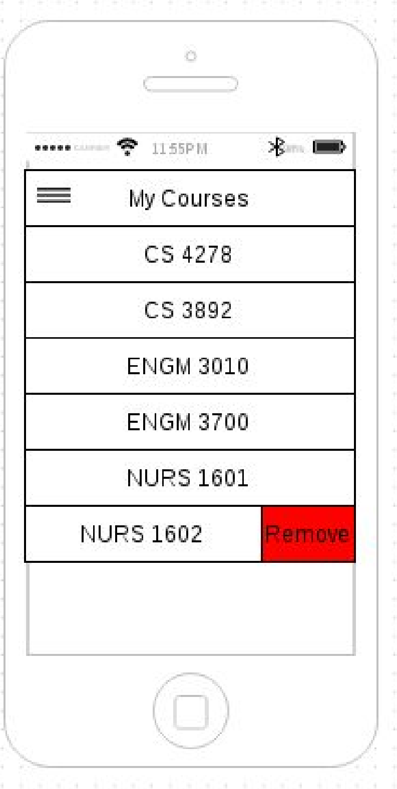
A. Low Fidelity Prototypes

The low fidelity prototypes for Class.ly began as hand-drawn images. Via tracing the outline of a smartphone and utilizing a ruler, Class.ly designers were able to mock up some temporary wireframes for several different pages of the application exhibiting the chat and meetup functionality. From there, designers utilized the web application ‘Draw.io” to create wireframe mockups mimicing the hand-drawn wireframes. This did not take into account any of the data collected, analyzed, and displayed in this document. Computer generated wireframes (via Draw.io) are shown below:

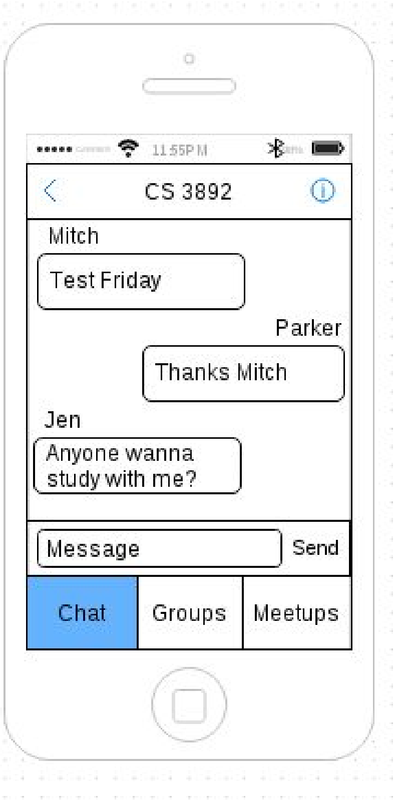
Side-Drawer High-Level Navigation



My Courses



Chat For Course



Mitch

Test Friday

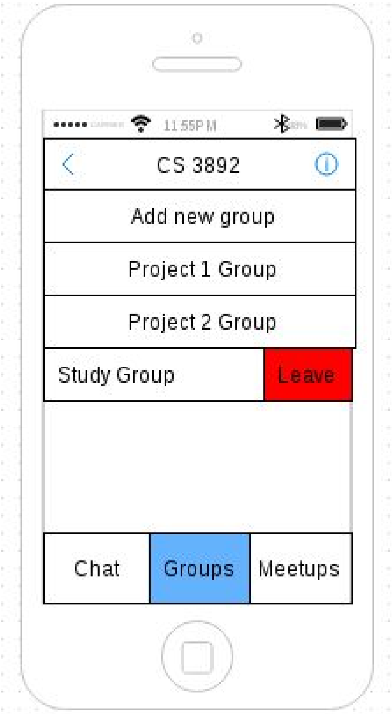
Parker

Thanks Mitch

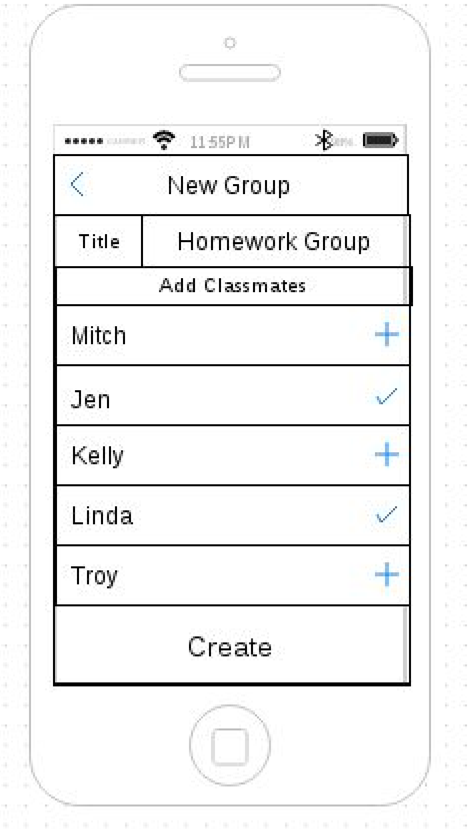
Jen

Anyone wanna study with me?

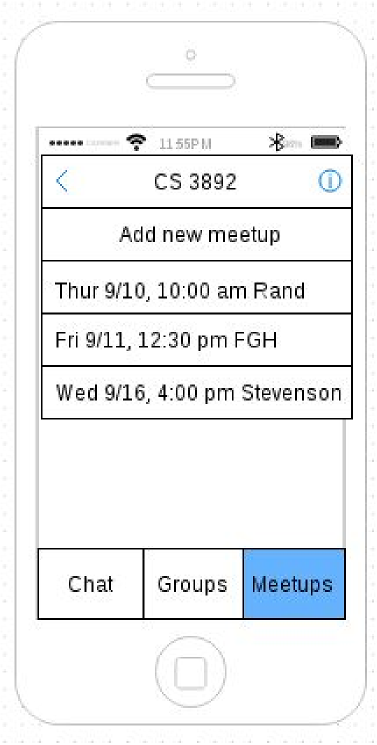
Group For Course



Create A New Group



New Meetup For Group

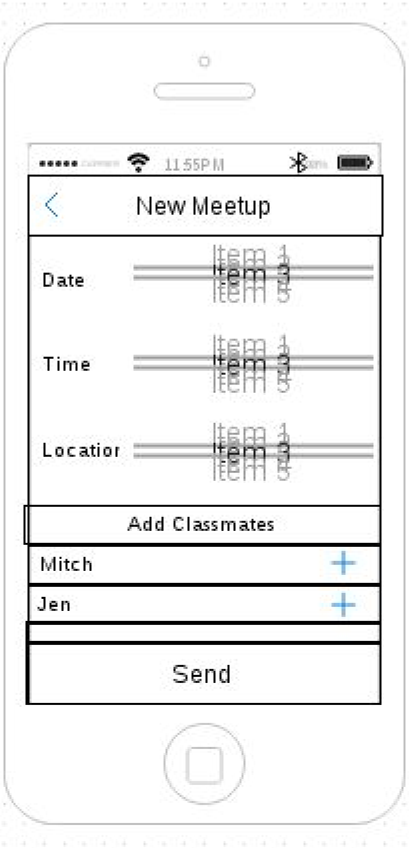


Thur 9/10, 10:00 am Rand

Fri 9/11, 12:30 pm FGH

Wed 9/16, 4:00 pm Stevenson

Meetup Details



B. High Fidelity Prototypes

To continue the prototyping process, we are using the Ionic Framework, a software development framework supporting the “write-once, run-anywhere” paradigm. With this framework, we utilize the MEAN Stack (MongoDB, ExpressJS, AngularJS, and NodeJS) to write a Model-View-Controller (MVC) web application representing Class.ly. This application can be used in any modern web browser, or on any iOS or Android device via the Ionic Framework, built on top of Apache Cordova (a native API for Android). We will use HTML and CSS to display dynamic content and create custom styling in the web application, which will be converted to web views fit for iOS and Android style guidelines. Eventually, the high fidelity prototype will be a full-fledged web application that exposes a user interface to the chat functionality, the custom chat functionality, the not “smart” meetup scheduler, and an option to enable and disable notification on a given group. The application will utilize both a side navigation drawer and a tabbed navigation interface, as per the quantitative data analysis results.

The prototypes will be fueled by mock data for students, courses, and meetups, created by Class.ly designers, and stored in the non-relational MongoDB document datastore. This limited backend, in conjunction with page routes (enabled by ExpressJS), will provide enough functionality to perform usability testing with a small subset of potential student users. By both guiding users to perform certain actions (e.g. schedule a new meetup), and watching how they normally interact with the application on multiple devices, Class.ly designers can iterate on the interface design, as well as the available functionality in order to improve the intuitiveness of the system and value provided to users and other key stakeholders. Additionally, in the future implementations of Class.ly, developers plan to perform simple logging in the application using the Elastic-Logstash-Kibana (ELK) stack. This open-source tools automatically creates a dashboard for developers to see how users are using their app, the faults they create, load times of specific resources, and other key metrics that can help improve app performance and value-added. A link to the Ionic Framework which facilitates creation of the high fidelity prototypes can be found here: http://ionicframework.com/.

Development of the high fidelity prototypes with Ionic is well under way. The backend systems, application framework, as well as some of the user-facing pages have been outlined and are being seeded with static data. Creating a feedback loop with potential users is not yet possible as the meetup scheduler is not fully (initially) implemented; however, as development continues and the application takes shape, initial user testing can begin.

VIII. Development Timeline (No Major Updates)

COMPLETE **ONGOING** **REMOVED**

Date	Class Requirements	Class.ly Requirements
9/10	Initial Project Description Documents	Initial Project Description Documents
9/14		Wireframe Prototypes (Draw.io Wireframes)
9/24	Requirements Documents	Requirement Documents App Development Begins (Ionic Framework)
10/1	Data Analysis Results, Prototyping Plan	Data Analysis Results, Prototyping Plan
10/6		Mock Data Complete (In Non-Relational Database) Back-end Prototype Complete (Database, API)
10/8		Front-end Prototype Complete (Ionic Framework, Bootstrap Style)
10/13	Prototype Demonstration	Prototype Demonstration
10/22	User Test Plan	User Test Plan
11/30	Project and Supporting Documentation	Project and Supporting Documentation
12/10	Partner Evaluation	Partner Evaluation

The documentation for CS3892 Human-Computer Interaction, as well as the source code for Class.ly is publicly available online in a Git repository. The repository can be found here: https://github.com/masiamj/CS3892-HCI-Class.ly.