

# CSCI 201 Final Project Detailed Design Document

## Team Members:

Samuel Breck ([breck@usc.edu](mailto:breck@usc.edu))  
Vadim Korolik ([korolik@usc.edu](mailto:korolik@usc.edu))  
Clifford Lee ([leecliff@usc.edu](mailto:leecliff@usc.edu))  
Prachi Nawathe ([pnawathe@usc.edu](mailto:pnawathe@usc.edu))  
Gautam Paranjape ([gparanja@usc.edu](mailto:gparanja@usc.edu))

---

## Hardware Requirements

### Client:

Optimal screen resolution: 1280px \* 720px or larger

### Server:

CPU: Intel i5 series or better

RAM: 2GB or more

HDD: 1GB or more

## Software Requirements

### Client:

Google Chrome 53.x+

### Server:

Linux: Ubuntu 16.04 LTS

Java: 1.8+

Tomcat: 8.5.x

MySQL Server: 5.7.16+

## Database Schema

(Key: PK = Primary Key, FK = Foreign Key, UC = Unique Constraint, AI = Auto Increment)

### Tables:

- Users
  -

id	BIGINT; PK; AI
username	VARCHAR(50); UC
password	VARCHAR(255)
email	VARCHAR(127)

timestamp	TIMESTAMP
-----------	-----------

- Friendships

- 

student_one	BIGINT; FK(Users.id)
student_two	BIGINT; FK(Users.id)
approved	ENUM(pending, accepted, declined)

- Note: the relationship is bidirectional and only one is stored (one->two OR two->one), queries must check both directions and insert only one.

- Events

- 

id	BIGINT; PK; AI
name	VARCHAR(255)
time	DATETIME
location	VARCHAR(255)
color	ENUM(red, orange, yellow, green, blue, indigo, violet, cyan, magenta)
notification	BIT

- EventRelationships

- 

user_id	BIGINT; FK(Users.id)
event_id	BIGINT; FK(Events.id)
relationship_type	ENUM(owned, shared)

- USCClasses

- 

class_id	BIGINT; PK; AI
section_id	UNSIGNED MEDIUMINT; UC(USCClasses.section_id, Discussions.section_id,

	Labs.section_id)
dept	VARCHAR(10)
class_num	UNSIGNED SMALLINT
semester	MEDIUMINT //format: 20163
name	VARCHAR(100)
professor	VARCHAR(50)
location	VARCHAR(255)
start_time	TIME
end_time	TIME
monday	BIT
tuesday	BIT
wednesday	BIT
thursday	BIT
friday	BIT

- Discussions



discussion_id	BIGINT; PK; AI
section_id	UNSIGNED MEDIUMINT; UC(USCClasses.section_id, Discussions.section_id, Labs.section_id)
class_id	BIGINT; FK(USCClasses.id)
start_time	TIME
end_time	TIME
monday	BIT
tuesday	BIT

wednesday	BIT
thursday	BIT
friday	BIT

- Labs

- 

lab_id	BIGINT; PK; AI
section_id	UNSIGNED MEDIUMINT; UC(USCClasses.section_id, Discussions.section_id, Labs.section_id)
class_id	BIGINT; FK(USCClasses.id)
start_time	TIME
end_time	TIME
monday	BIT
tuesday	BIT
wednesday	BIT
thursday	BIT
friday	BIT

- Exams

- 

exam_id	BIGINT; PK; AI
class_id	BIGINT; FK(USCClasses.id)
start	DATETIME
end	DATETIME

- EnrolledClasses

-

user_id	BIGINT; FK(User.id)
class_id	BIGINT; FK(USCClasses.id)

- Notifications

- 

notification_id	BIGINT; PK; AI
user_id	BIGINT; FK(User.id)
event_id	BIGINT; FK(Events.id)
notification_type	ENUM(email, online)
completed	BIT

- Study Recommendations

- 

target_user_id	BIGINT; FK(Users.id)
recommended_user_id	BIGINT; FK(Users.id)
class_id	BIGINT; FK(USCClasses.id)
event_id	BIGINT; FK(Events.id)
notification_id	BIGINT; FK(Notifications.id)

## Class Diagrams/Inheritance Hierarchy

### BACKEND

#### PDFParser Class (Apache PDFBox)

- + static read(File fileToRead) //from a ICS file
- + static read(PDFFile pdf) //from a pdf file

#### Calendar Class implements Serializable

- Vector<Event>
- + Calendar()
- + Calendar(Calendar c)
- + read() //from the database
- + update()
- + delete()
- + share(User u)
- + createEvent()
- + createEvent(Event e)
- + serialize() //convert to JSON object

#### Event Class implements Serializable

- String name
- Timestamp start
- Timestamp end
- Vector<User> collaborators
- Vector<Block> blocks
- + Event()
- + Event(Event e)
- + Event(String name, Timestamp start, Timestamp end, Vector<User> collaborators)
- + read()
- + update()
- + share(User u)
- + serialize() //convert to JSON object

#### Course Class extends Event

- String courseTitle
- + Course()
- + Course(Course c)
- + Course(String courseID)
- + serialize() //convert to JSON object

### **Block Class implements Serializable**

- Datetime (Representation of Date & Time) start
- Datetime end
- + serialize() //convert to JSON object

### **Real Time Event Class implements Serializable**

- String name
- String notification\_text
- int type
- + serialize() //convert to JSON object

### **User Class implements Serializable**

- String name
- String email
- Vector<User> friends
- + serialize() //convert to JSON object

## **FRONTEND**

### **Main View Class**

- Calendar View
- RightSideBar View
- CreateEvent Button
  - + onClick()
    - Create a default event on this day of the calendar

### **Calendar View Class**

Using UI-Calendar to anchor to an html div, which will create the calendar component in our front end.

- FullCalendar Calendar div
  - + dayClick()
    - Create a calendar event on the location clicked
    - Show the EventCustomize View for this event
    - Save the calendar with updated events
  - + eventClick()
    - Show the EventCustomize View for this event
    - Save the calendar with updated events
  - + eventDragStop()
    - Save the calendar with updated (moved) events
  - + eventResizeStop()
    - Save the calendar with updated (resized) events

### Event Customize View Class

- form createEventForm
  - input text field title
  - select drop-down startTime
  - input date field startDate
  - select drop-down endTime
  - input date field endDate
  - input text field friends with data-list of user's friends
  - ul of spans denoting added collaborators
  - input text field title
- + onSubmit()
  - Validate form. Create JSON object from input and send to server as AJAX POST request.

### RightSideBar View Class

- signIn Button
  - + onClick()
    - Show SignIn Modal over Calendar View
- upload Button
  - + onClick()
    - Show Upload Modal over Calendar View
- friends Button
  - + onClick()
    - Show Friends Modal over Calendar View
- list of Notification Views

### SignIn Modal Class

- Login Button
  - + onClick()
    - Switch to login view
      - html text input for username
      - html text input for password
      - login Button
        - + onClick()
          - Validate form. Create JSON object from input and send to server. Update frontend if valid login. The frontend receives the list of friends for the user.
- continueAsGuest Button
  - + onClick()
    - Close Modal



- Register Button
  - + onClick()
    - Switch to register view
      - html text input for username
      - html text input for password
      - login Button
        - + onClick()
          - Validate form. Create JSON object from input and send to server. Update frontend if valid login.
    - continueAsGuest Button
      - + onClick()
        - Close Modal

### **Friends Modal Class**

- searchColumn div
  - + html text input for friend search
    - onSubmit()
      - AJAX POST to the friendships table
- friendRecommendation div
  - + html text input for sectionID search
    - textInputted()
      - AJAX GET USCClass object
  - + html text to display results from result of textInputted()
- friendList div
  - + populateWithFriends()
    - populate with local instance of logged in User.friends

### **Upload Modal Class**

- html file input for PDF syllabus submission
  - + onSubmit()
    - AJAX POST request file to the server
    - Frontend switched back to calendar view - upload modal hidden
- html file input for ICS calendar file submission
  - + onSubmit()
    - AJAX POST request file to the server
    - Frontend switched back to calendar view - upload modal hidden

### **Enroll Modal Class**

- ul wrapper for input text fields

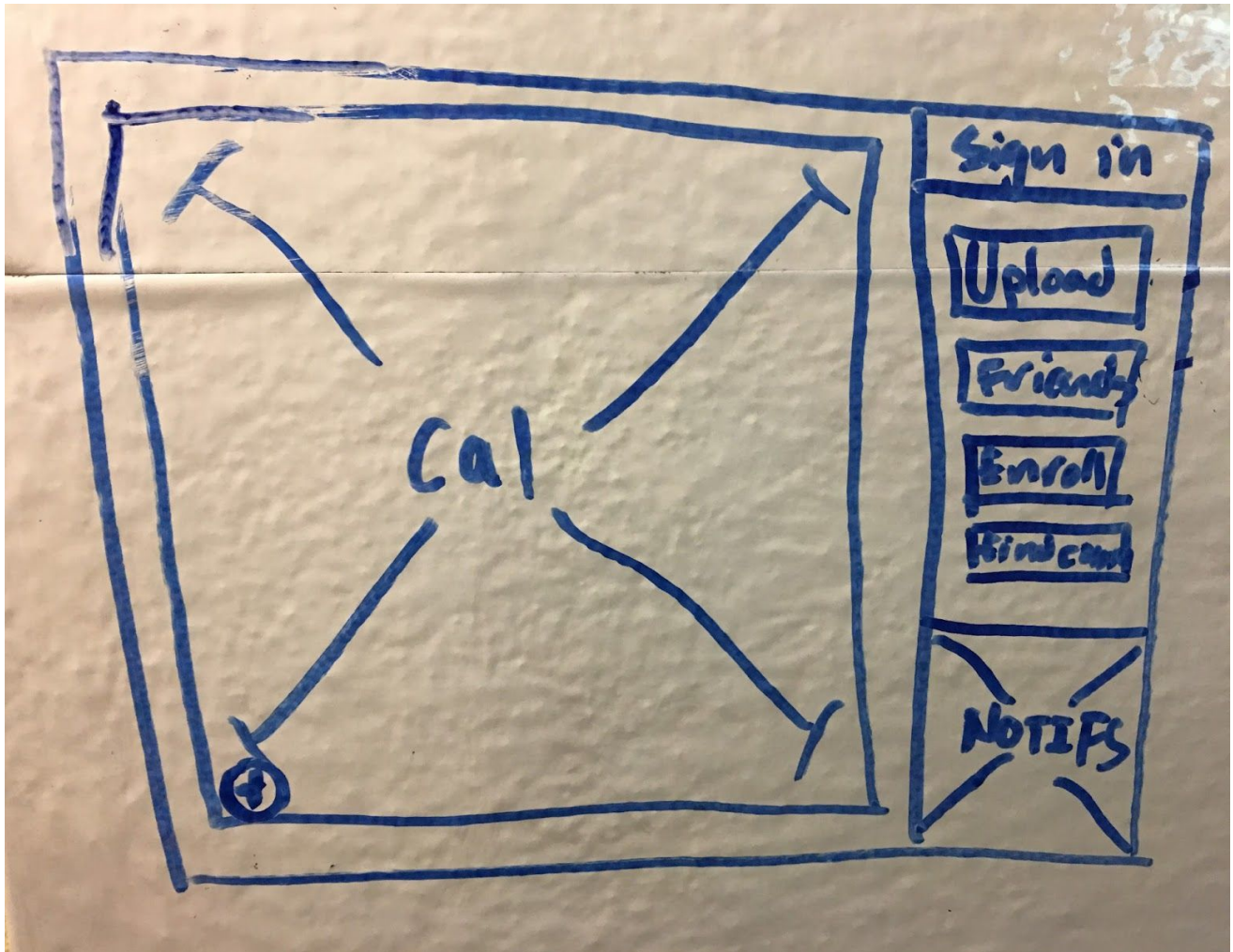
- input text fields
  - + onFocus()
    - Request to Trojan-course-API to verify section ID
    - Another input text field added as child element to ul
- + onSubmit()
  - Converts each section ID into a course JSON object via Trojan-course-API, sends JSON objects to our servers to be saved

### **Notifications View Class**

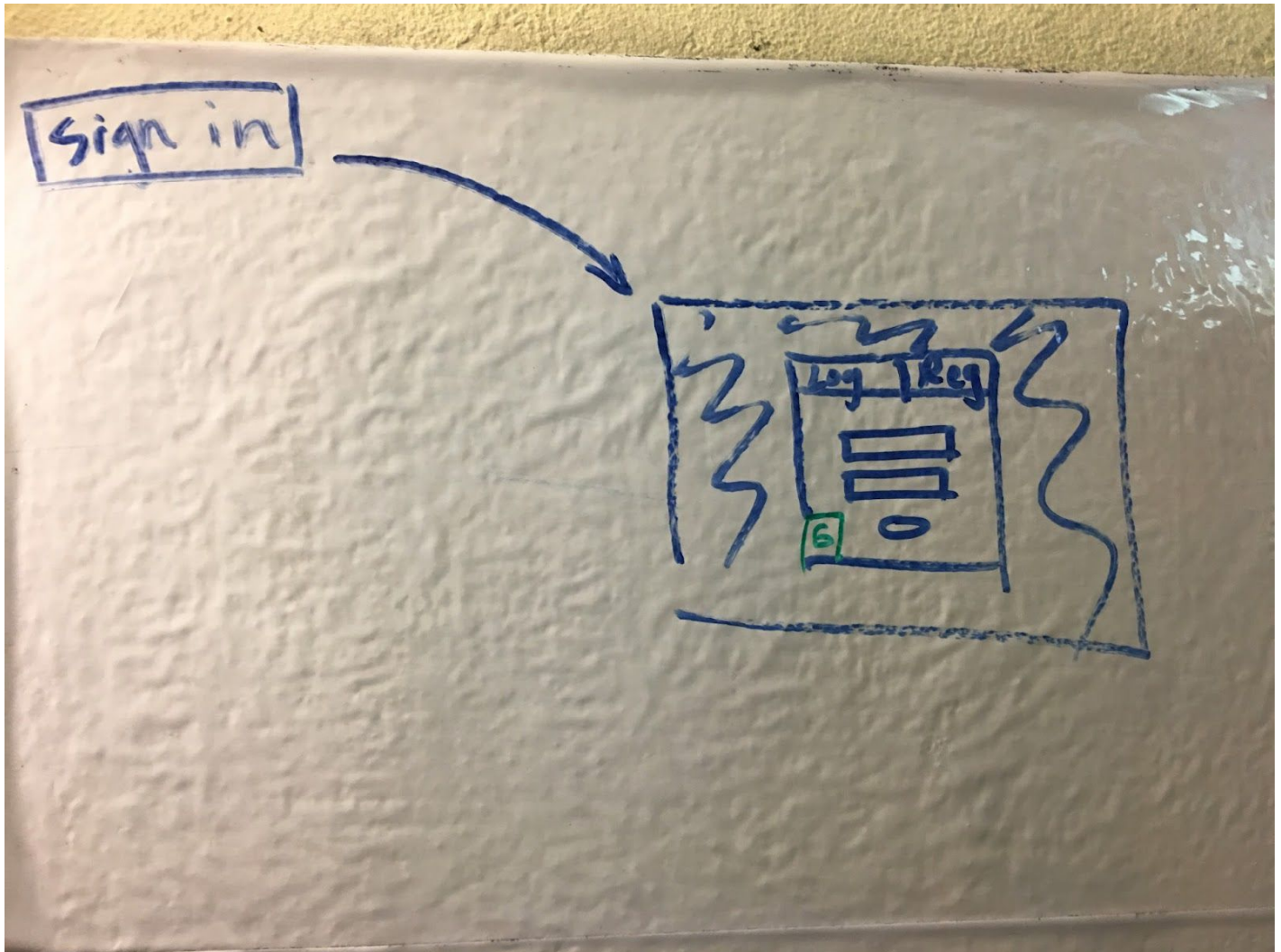
- html div with text information about the notification
- Different background color based on notification type
- clear button
  - + onClick()
    - Remove this notification view from the notifications panel

## FRONTEND GUI MOCKUPS

### Main View

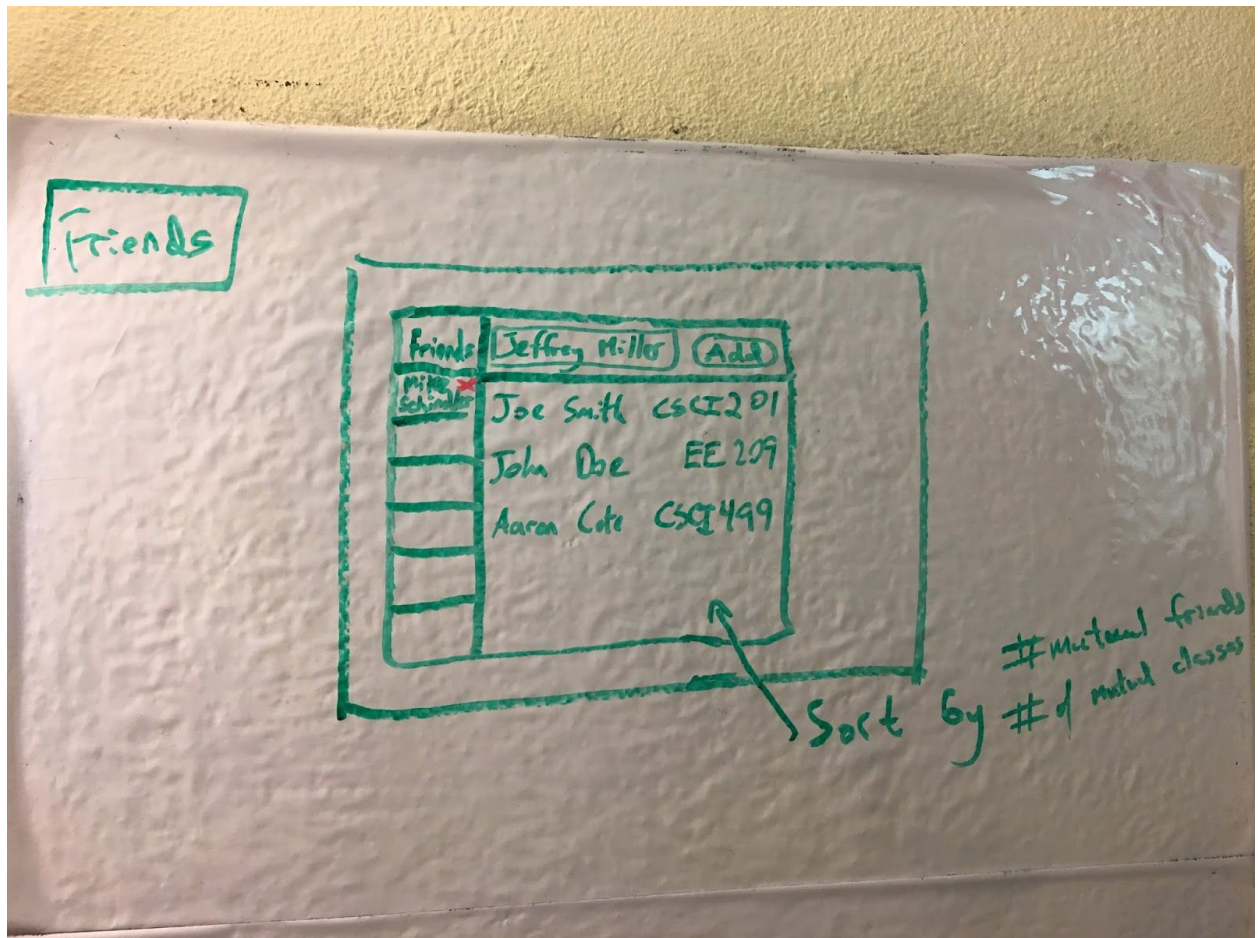


## Sign In Modal

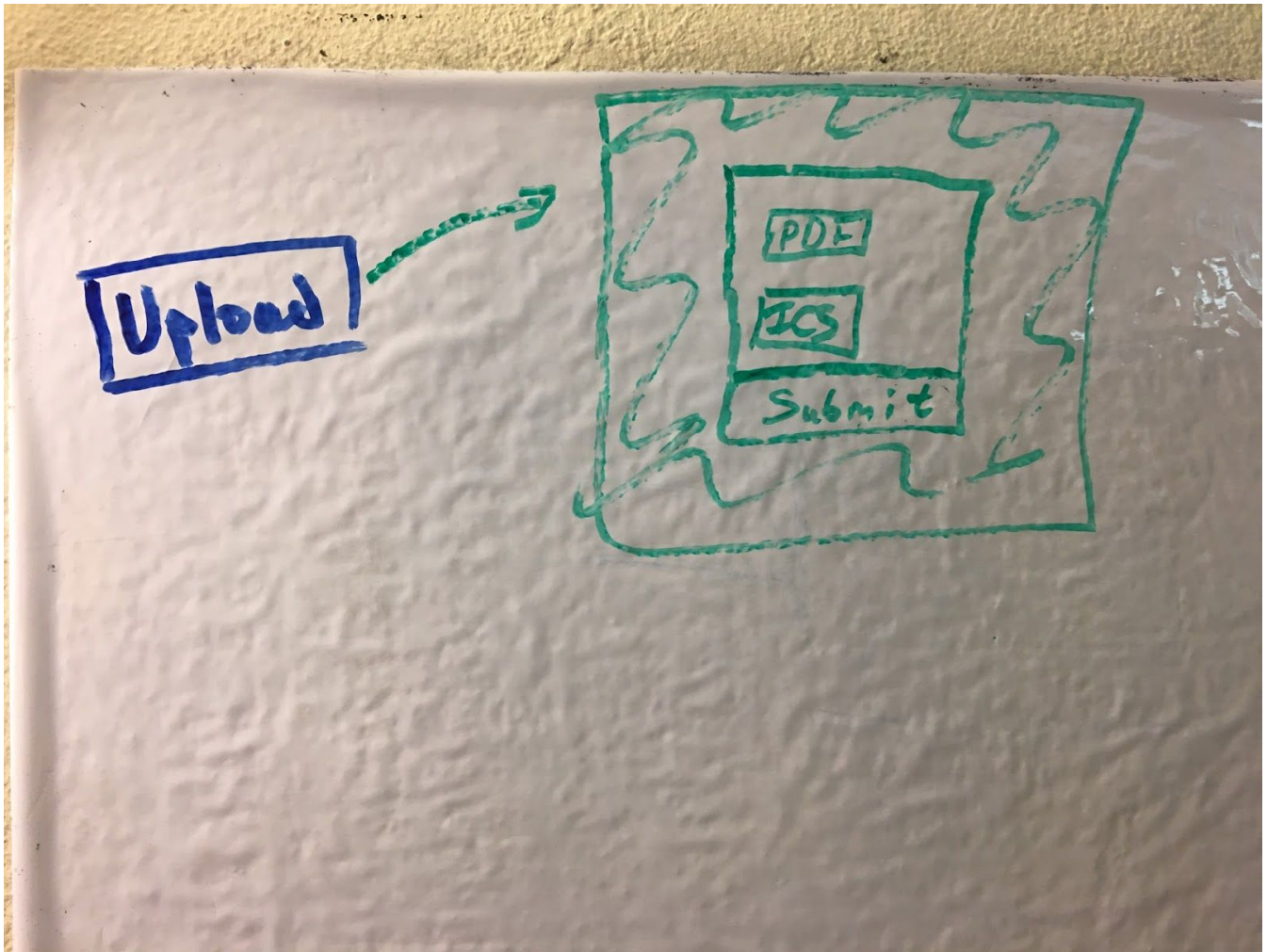




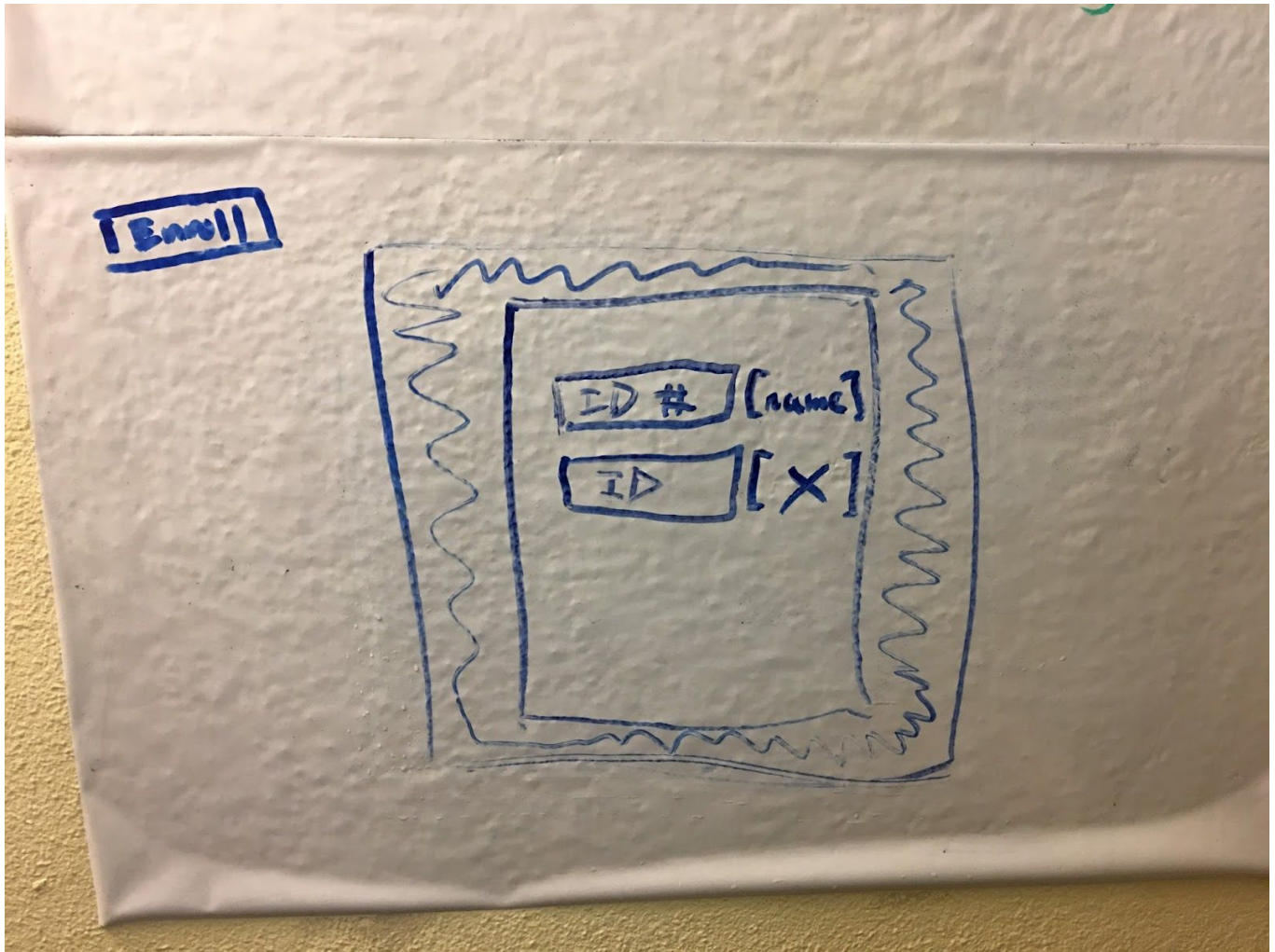
## Friends Modal



## Uploads Modal



## Class Enroll Modal





## Event Creation Modal

