Scribe Scholars

PREPARED BY Team 7

Riley Robertson, Jake Koontz, Walter Jacquette, Kyle Brown, Jeremy Putnam, and Eric Thiem

1. Purpose

Technology is rapidly pushing itself to the forefront of our classrooms by providing cutting-edge advancements to learning, allowing the students to not only learn but to have fun while doing so. Here is the problem though: there is no-one stop shop for teachers and students to connect with each other. There needs to be a platform for all the students' and teachers' needs that is also user-friendly, easily accessible, and open to a wide variety of students of all ages. While there are certain applications that have some of these desired features, Blackboard, Piazza, Canvas, etc., none of them combine them into an all-in-one classroom application. This application will feature in-class activities, out of class assignments, discussion boards, messaging systems between parents and teachers, and a detailed dashboard for teachers to gauge student performance.

Functional Requirements

1. Log In / Sign Up
    a. All users should be able to create an account.
    b. When creating an account they should be able to pick a role (student, parent, teacher, administrator).
    c. On successful login, they should be greeted with a home page.
2. Creating / Joining A Classroom
    a. Teachers
        i. They should be able to create a classroom and have an auto-generated code for students to use to join the class.
    b. Students
        i. Using the code given by teachers they should be able to join the class.
3. Landing Page
    a. The page will have a calendar in the center of the page with upcoming events.
    b. Underneath the calendar will be a notifications section with important messages.
    c. The sidebar will have all the classes the student or teacher is in.
4. Announcements
    a. On the class page, there will be a section with announcements from the teachers for students or parents to see.
5. In-Class Lessons / Homeworks / Quizzes / Tests
    a. Teachers

        i. There will be a section where teachers will be able to create their lessons with videos, readings, and questions.
        ii. While students are completing the lesson the teacher will have a live feed of their progress for in-class lessons.
    b. Students
        i. On the course webpage, there will be sections the teacher has created that the students will be able to complete in class or at home, depending on what the teacher desires.
6. Classroom StatsBoard
    a. The teacher will have a specialized landing page with classroom statistics that show classroom performance.
7. Student StatsBoard
    a. The teacher will also have access to a student personalized dashboard with specific statistics on individual students.
8. Lesson Creation
    a. The teacher will have full control over the assignment making, with the ability to add questions, readings, links, and videos.
9. Administration StatsBoard
    a. The school administrators will have their own dashboard with not only classroom performance and student performance, but all the school's classes' performance.
10. Classroom Discussion Board
    a. The discussion board will be combined, giving teachers, parents, and students the ability to ask any questions they have regarding assignments, deadlines, or just classroom problems.
11. Parent-Teacher Messaging
    a. This feature will be only for teachers and parents to message about problems in the classroom, or parents to ask questions individually to the teachers.

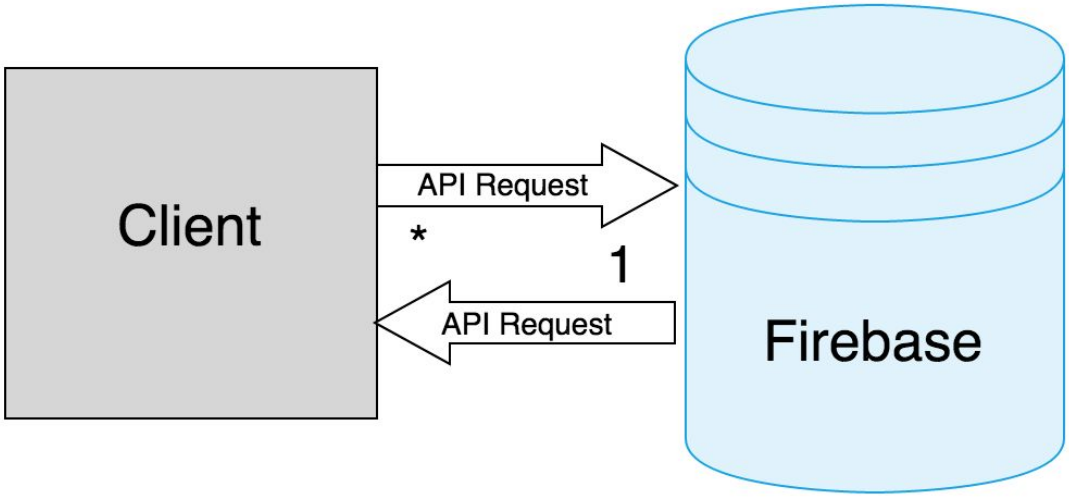
2. Design Outline

1. Client
    a. Sends requests through Firebase API
    b. Receives response from Firebase servers

2. Firebase
    a. Server
        i. Receives requests from clients
        ii. Sends response to clients
        iii. Receives requests from database
        iv. Sends response to database
    b. Database
        i. Receives and stores data from Server
        ii. Retrieves and sends data stored

High-Level Overview

Our website uses a server-client model as shown in the diagram below. The web pages are clients which send requests and receive responses from Firebase, which acts as our Server and Database.



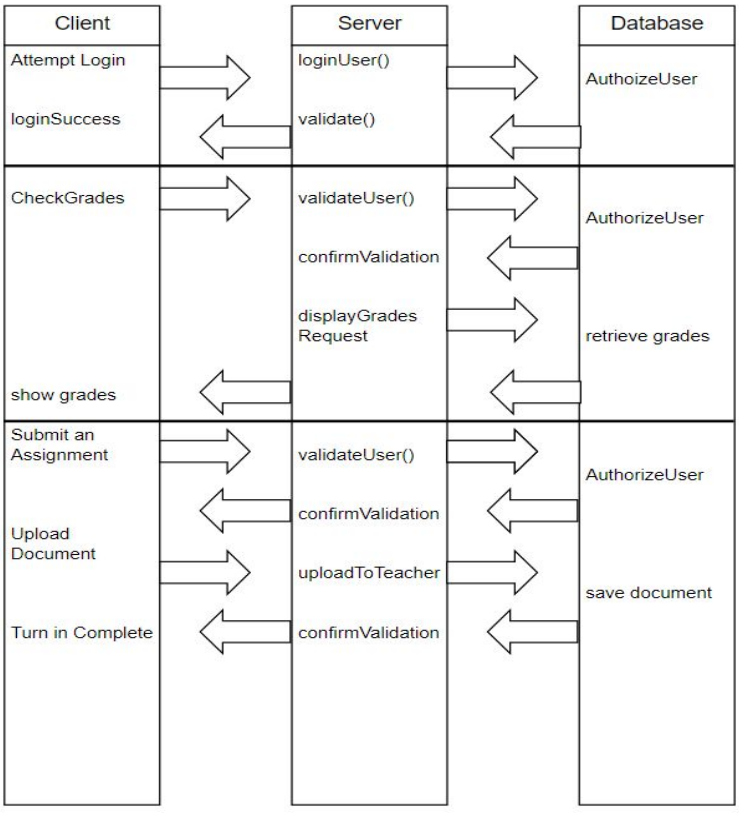
<u>Interactions between System Components:</u>

1. Client opens new page
    a. Client requests page information from Firebase
    b. Client loads page from response

2. Client attempts to login/sign up
    a. Client requests information on user based on username and password
        i. Firebase checks if user exists and password is correct
        ii. Returns appropriate response to Client
    b. Client loads landing page if successful login

3. Client attempts to create new classroom
    a. Client sends user information to Firebase
        i. Firebase verifies user is teacher and if so creates a new classroom
        ii. Returns appropriate response
    b. Client loads landing page of new class

4. Client attempts to turn in assignment
    a. Client sends user information, assignment information, and time
        i. Firebase verifies user, checks the time, and sets assignment for that user to complete
        ii. Firebase returns verification of submission and the assignment score if teacher has this turned on for the class
    b. Client updates page with information returned from Firebase



3. Design Issues

a. Functional Issues

How to handle different types of users?

Option 1: One user class which differentiates between privileges on its own
Option 2: Individual classes for students, teachers, parents, administrator inheriting user

We ended up choosing to have an individual class for each classification. Originally we debated having one single landing page and then only certain features available to specific users. We went against this option because we believe it would be better to have specialized pages for each user. For example, the student landing page can have a calendar of upcoming events and the teacher can have a dashboard of her classroom's performance.

How to organize the navigation bar?

Option 1: Sidebar navigation
Option 2: Top of page navigation

Top bar navigation is extremely common in today's websites, such as Facebook and Blackboard, and is very useful. Sidebar navigation is becoming increasingly popular and adds a bit more modern feel to the website. Our sidebar will be crucial to the application because it will hold the link to the student's classes and a home button to return to the landing page.

How to organize student landing page?

Option 1: Class buttons in the middle of the page
Option 2: Class buttons on sidebar and useful information in the middle

Many programs similar to ours (Canvas / Blackboard) structure their main pages to have the first step be clicking on the class you want to see. We believe when a student first arrives at the site they should still have their classes on the sidebar, but the main focus should be a calendar with all important dates coming up. This allows students to be able to see important dates and prioritize how to spend their time rather than having to keep track by clicking on each class to find the information.

b. Non-Functional Issues

What backend framework should we use?

Option 1: Django (Python)
Option 2: Firebase
Option 3: NodeJS

When it comes to the backend, our team did not have much experience with any server frameworks and we were going to have to learn anything we wanted to use from scratch. Django has the great benefit of being a well-developed framework that has a very detailed structure that helps to keep the code very organized. NodeJS is more accurately an extension of Javascript that has the capability to be a server with libraries to extend it even farther. Firebase is a backend server created by Google that has capabilities in secure authentication, database, picture storage, and analytics. Given that we have so many features we would like in our application, we decided being able to have an already-built backend end will dramatically increase our productivity and allow us to release more features.

What frontend framework should we use?

Option 1: ReactJS
Option 2: AngularJS
Option 3: Pure HTML, CSS, Javascript

One key component in our application is to have an easy-to-use user interface. While it is definitely possible to make websites using pure HTML,

CSS, and Javascript, it adds a huge amount of time to make everything go on the page the way you want it to, and it also has little to no reusability. This is where Angular and React come in. Both frameworks have their benefits and popularity, but our team had more experience with React and believe it is easier to pick up in a short amount of time. Angular has its benefits but is a lot heavier with more features we don't truly need.

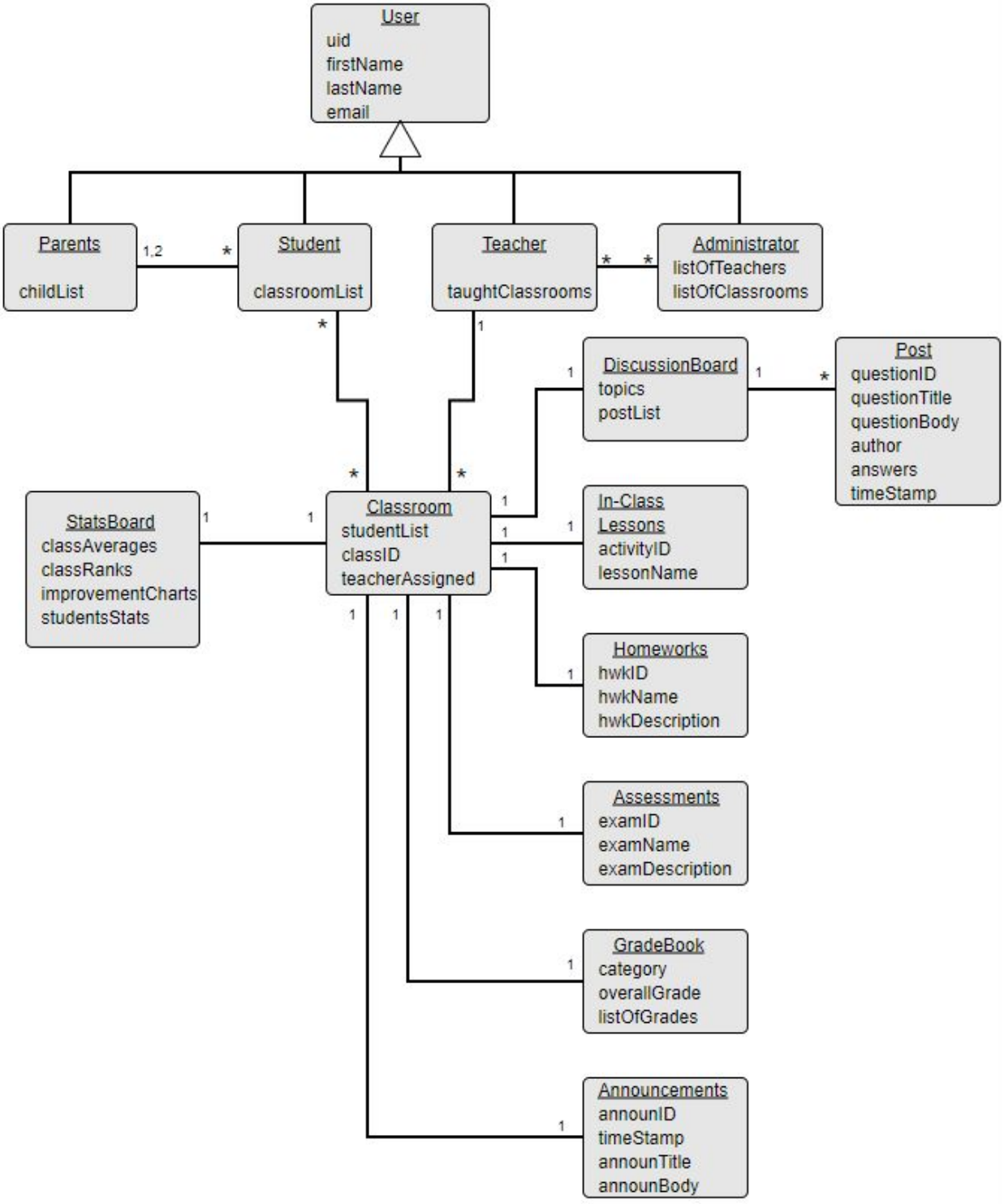
What CSS Grid System should we use?

Option 1: Foundation
Option 2: Bootstrap

When it comes to CSS frameworks it is almost a given to use one to make our lives easier and have a responsive website without having to spend hours trying to manually do it. Both systems are very similar in the fact they have grid systems and pre-built components that can be modified to fit your website. While our group has a small amount of experience with Foundation and none with Bootstrap, we believe Bootstrap will be a better choice because of its popularity, which means a larger amount of resources for us to use online.

4. Design Details

a. Class Level Design



b. Class Interactions

i. Student
   1. The students will be able to interact in the classrooms, with each of the classes that are assigned to a classroom such as completing homework, viewing announcements, posting on a discussion board, etc.

ii. Parent
   1. The parent will have a messaging system to contact the teachers.
   2. They will also be able to view student grades and class announcements.

iii. Teacher
   1. The teacher will be able to create a variable number of classrooms which will contain all its associated classes.
   2. The teacher will be able to push notifications/announcements to the students and parents.
   3. The teacher will have a messaging system to contact the parents.
   4. The teacher will be able to create in-class activities, homeworks, and assessments.
   5. The teacher will be able to create announcements and send them to the students.

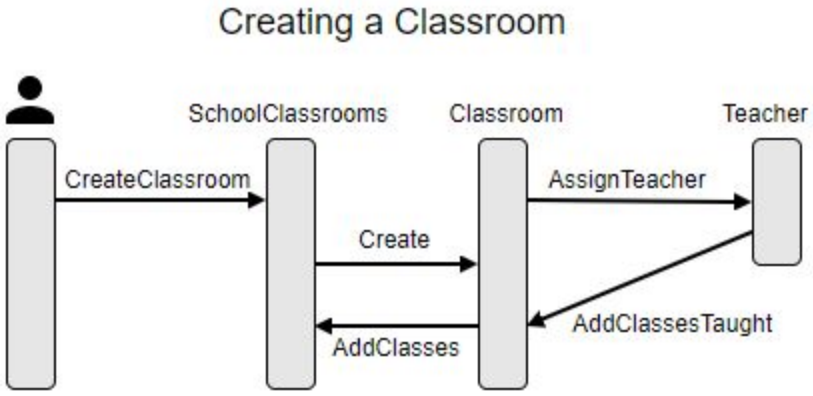
iv. Administrator
   1. The administrator will be able to view information about the teachers and will be associated with all of the teachers in the school.

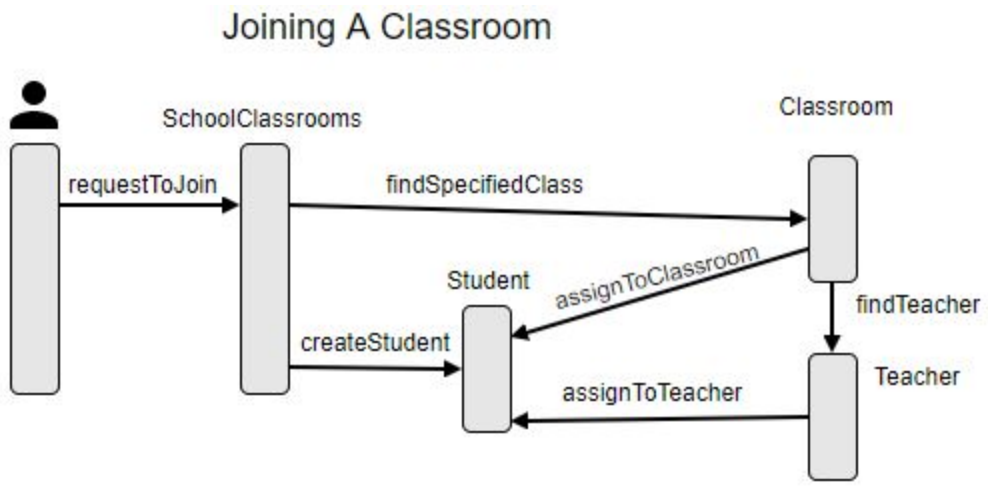
v. Classroom
   1. The classroom will be interacting with all the different associated classes.
   2. Each classroom will have in-class activities, homeworks, and assessments created by the teacher.
   3. They will contain the grades for all of the students in the class and tell the students' their own grades.
   4. They will have a discussion board for each specific class.
   5. There will be announcements that will be pushed to each student by the teacher.

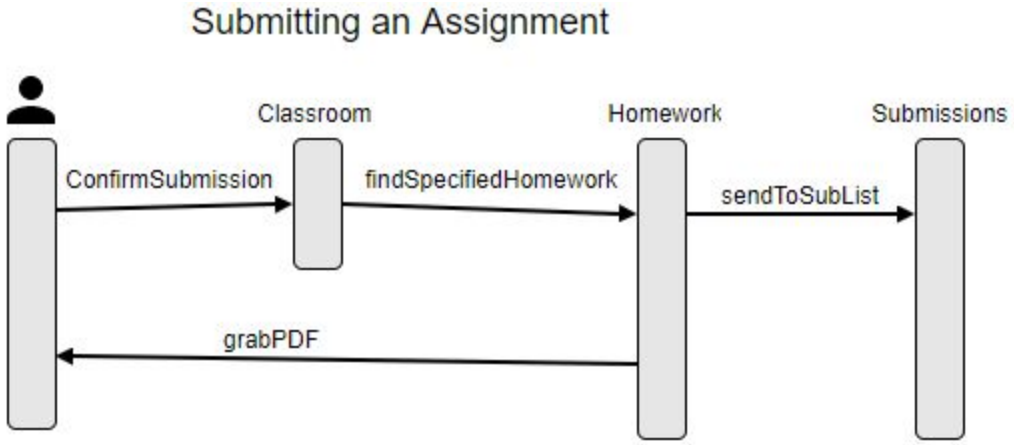
vi. Discussion Board

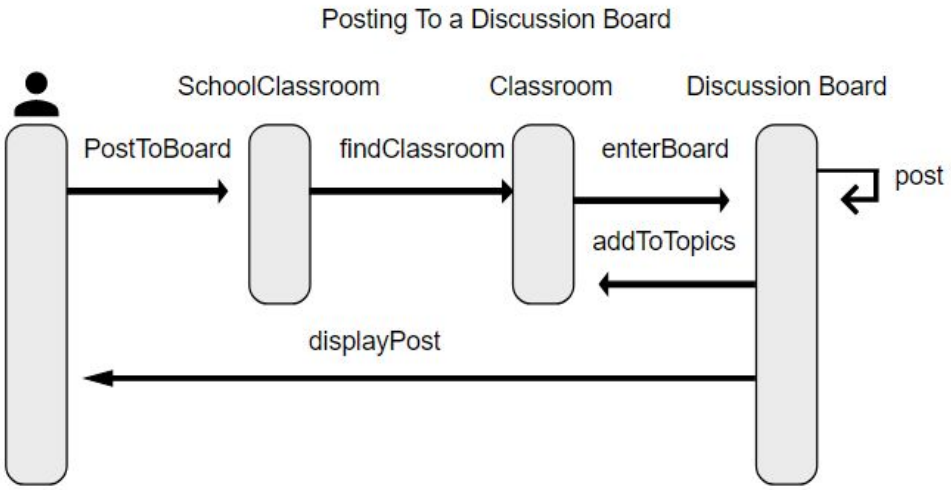
1. The board will have access to each of the posts associated with this board and display them.

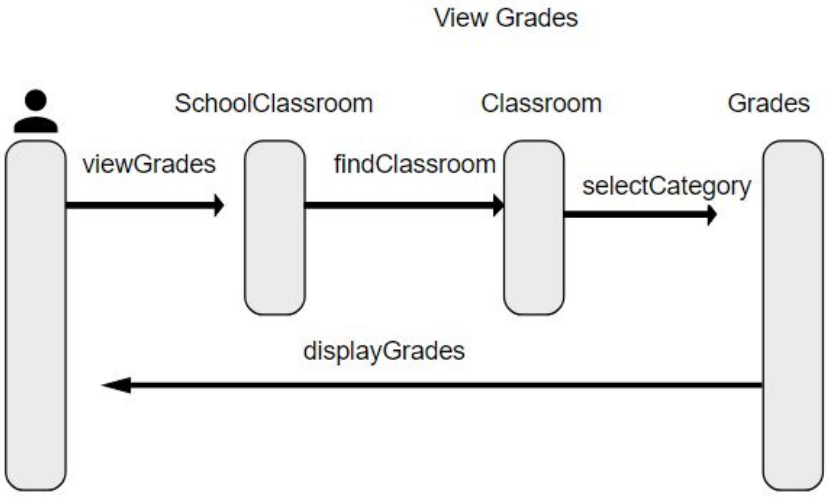
c. Sequence Diagrams





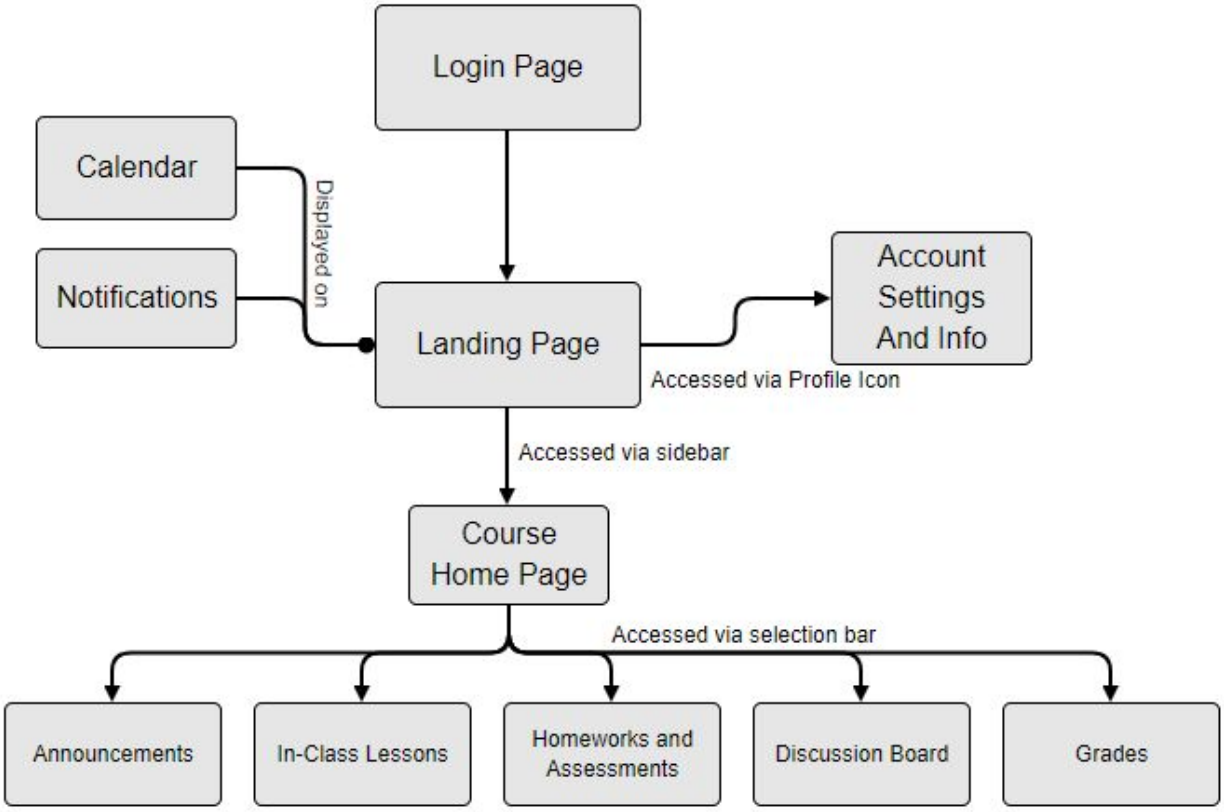






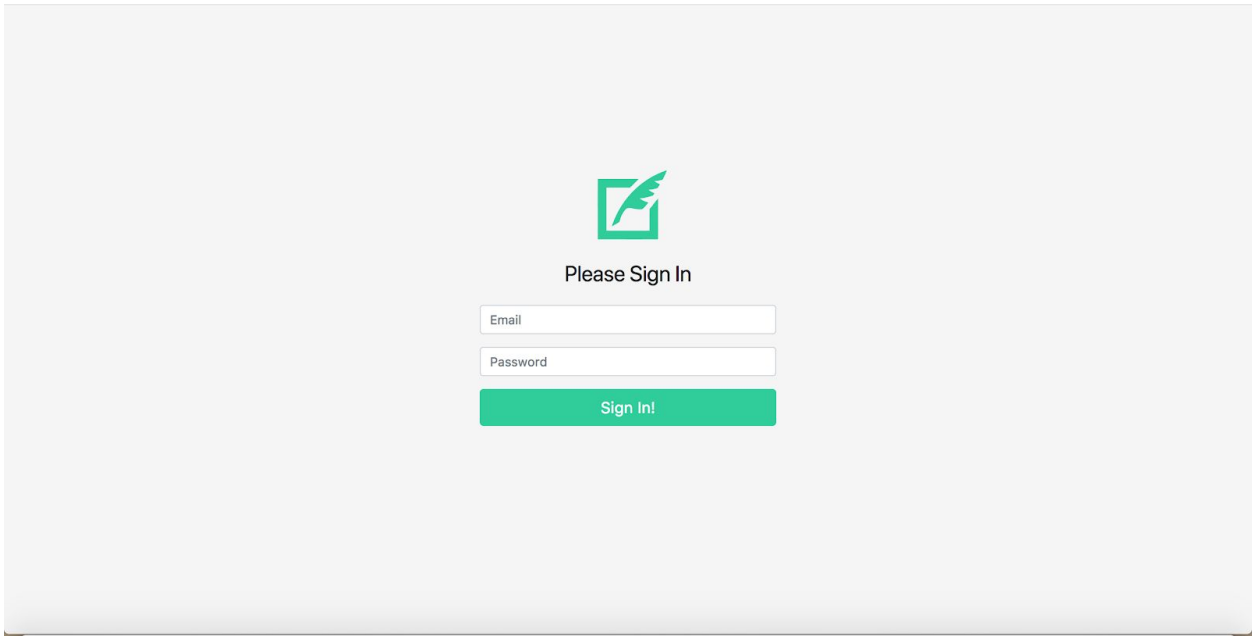
d. Activity Diagram

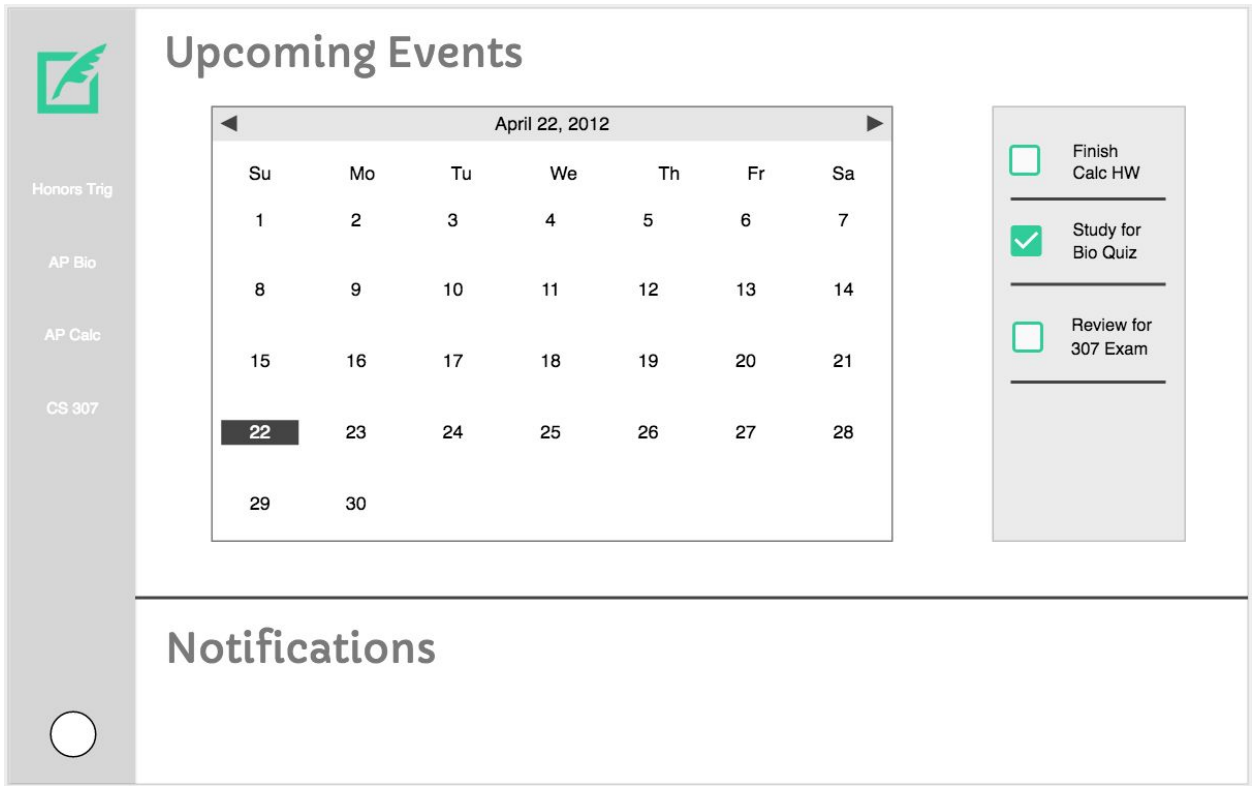
The goal of the navigation is for the user to easily be able to view/access anything they need quickly. There should never be more than a few clicks before they access the page they're looking for. The landing page should be able to display most of what a student would need to quickly view, and they should only need to go further into course pages when needing something more specific.

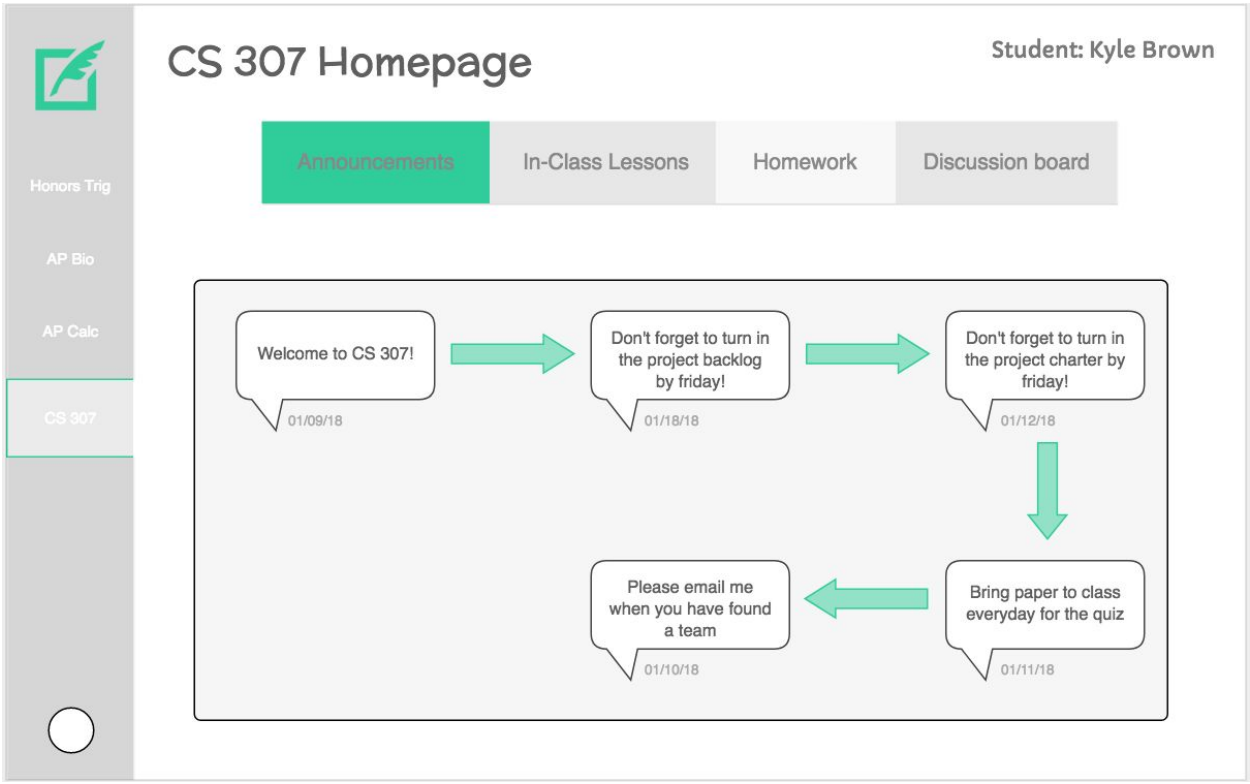


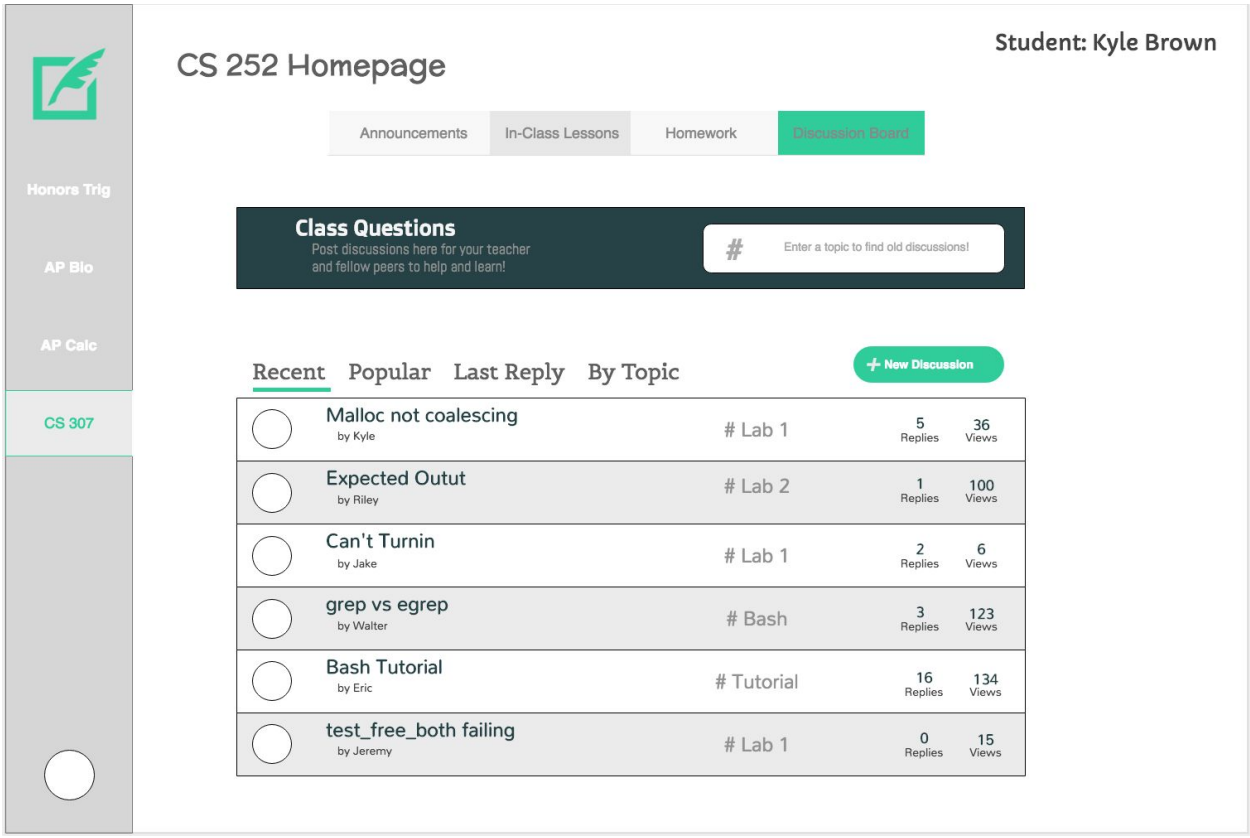
e. UI Mockups

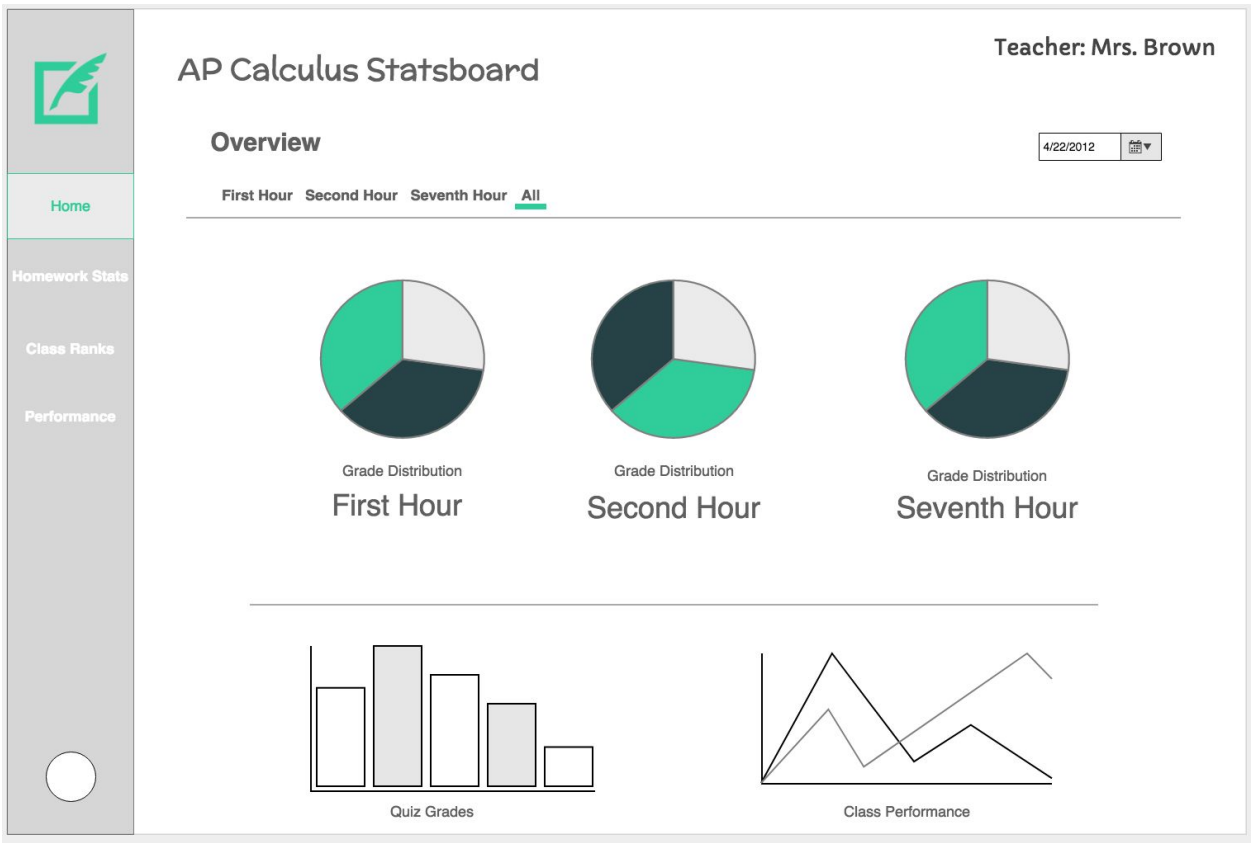
The UI throughout the website is designed to attempt to be unique yet simple and convenient. The user should be able to easily see everything available and confusion should be minimized. With the UI bring clearly labeled and organized well, there should be no trouble traveling through the website.


Please Sign In
Email
Password
Sign In!


Upcoming Events
Honors Trig
AP Bio
AP Calc
CS 307
April 22, 2012
Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
Finish Calc HW
Study for Bio Quiz
Review for 307 Exam
Notifications


CS 307 Homepage
Student: Kyle Brown
Honors Trig
AP Bio
AP Calc
CS 307
Announcements
In-Class Lessons
Homework
Discussion board
Welcome to CS 307!
01/09/18
Don't forget to turn in the project backlog by friday!
01/18/18
Don't forget to turn in the project charter by friday!
01/12/18
Bring paper to class everyday for the quiz
01/11/18
Please email me when you have found a team
01/10/18


CS 252 Homepage
Student: Kyle Brown
Honors Trig
AP Bio
AP Calc
CS 307
Announcements
In-Class Lessons
Homework
Discussion Board
Class Questions
Post discussions here for your teacher and fellow peers to help and learn!
Enter a topic to find old discussions!
Recent
Popular
Last Reply
By Topic
+ New Discussion
Malloc not coalescing
by Kyle
Lab 1
5 Replies
36 Views
Expected Outut
by Riley
Lab 2
1 Replies
100 Views
Can't Turnin
by Jake
Lab 1
2 Replies
6 Views
grep vs egrep
by Walter
Bash
3 Replies
123 Views
Bash Tutorial
by Eric
Tutorial
16 Replies
134 Views
test_free_both failing
by Jeremy
Lab 1
0 Replies
15 Views


AP Calculus Statsboard
Teacher: Mrs. Brown
Overview
4/22/2012
Home
Homework Stats
Class Ranks
Performance
First Hour
Second Hour
Seventh Hour
All
Grade Distribution
First Hour
Grade Distribution
Second Hour
Grade Distribution
Seventh Hour
Quiz Grades
Class Performance