

# ***iStudy***

Design Document



Team 1

Weijie Meng

Mengxue Luo

Qian Zhang

Yang Shi

Guocheng Wei

Xiaohua Shi

# TABLE OF CONTENTS

<b><u>PURPOSE</u></b>	<b>3</b>
<a href="#"><u>Functional Requirements</u></a>	
<a href="#"><u>Non Functional Requirements</u></a>	<b>5</b>
<b><u>DESIGN OUTLINE</u></b>	<b>6</b>
<a href="#"><u>High Level Overview</u></a>	<b>6</b>
<a href="#"><u>Flow of Events</u></a>	<b>7</b>
<b><u>DESIGN ISSUES</u></b>	<b>9</b>
<a href="#"><u>Functional Issues</u></a>	<b>9</b>
<a href="#"><u>Non Functional Issues</u></a>	<b>12</b>
<b><u>DESIGN DETAIL</u></b>	<b>15</b>
<a href="#"><u>Data Classes Diagram</u></a>	<b>15</b>
<a href="#"><u>Data Classes Description</u></a>	<b>16</b>
<a href="#"><u>Sequence Diagram</u></a>	<b>19</b>
<a href="#"><u>UI Mockups</u></a>	<b>21</b>

## PURPOSE

Our project is a great application that gives the students an easy way to manage their daily academic activities. This application is composed of three major

features, giving class registration suggestion according to the professor rating and class complexity, assisting students to look for classmates in the same course as study companion who might be easy to get along with based on their interests and language they are speaking and helping students to make an appropriate meeting time with group members when taking a course that requires group work. The main purpose of our application is that “one tool, one platform and one user account but manage everything in students’ academic lives”. We aim to make students’ academic life easier, more efficient and more satisfied.

## **Functional Requirement**

### **Profile:**

1. As a user, I would like to change my profile picture.
2. As a user, I would like to edit my hobbies, education info, birthday in my profile
3. As a user, I would like to set my personal information to be public or private

### **Class registration suggestion feature:**

1. As a user, I would like to search a class by typing the class name.
2. As a user, I would like to select a class taught by a specific professor.
3. As a user, I would like to know abstract information about the class.

4. As a user, I would like to see the rating of the professor.
5. As a user, I would like to see the reviews of the class by other users.

**Class Forum feature:**

1. As a user, I would like to join a class as a classmate.
2. As a user, I would like to see number of classmates in the class before I join the class.
3. As a user, I would like to see the list of classmates in the class I joined.
4. As a classmate, I would like to leave a class I joined.
5. As an administrator, I would like to create three fixed threads in each class forum as “welcome new students”, “random chat” and “academic resources sharing” where classmates are asked to discuss about a specific topic.
6. As a classmate, I would like to send messages to “random chat” and “welcome new students” threads.
7. As a classmate, I would like to receive messages from different threads.
8. As a classmate, I would like to share course resources in form of files in “academic resources sharing” thread.
9. As a classmate, I would like to manage the resources I have shared in the class by either uploading new files or deleting the files I have
10. As a classmate, I would like to have access to resources shared by my classmates.

**Rate professor feature:**

1. As a classmate, I would like to post a review for the class I have joined.
2. As a classmate, I would like to rate the class with “good”, “average” and “poor” tag while I’m writing the review.
3. As a classmate, I would like to delete my post.
4. As a classmate, I would like to ‘like’ my classmates’ reviews.
5. As a classmate, I would like to dislike’ my classmates’ reviews.
6. As an administrator, I would like to make sure only users who have joined the class are able to post reviews and modify his or her posts for the class.

**Find classmates feature:**

1. As a user, I would like to find friends from a class I joined based on my preference such as country, gender and etc.
2. As a user, I would like to have a friend list.
3. As a user, I would like to add a person to my friend list by searching his or her account email.
4. As a user, I would like to chat with my friends in the friend list privately.
5. As a user, I would like to form a group with my classmates by sending invitations to my classmates .
6. As a group member, I would like to send messages to my group and discuss course material problems with my group members.
7. As a user, I would like to delete a friend.
8. As a user, I would like to leave a group.

**Set up meeting time feature:**

1. As a user, I would like to form a group by sending invitations to my group members.
2. As a group member, I would like to create a time availability vote for my team members as an event creator.
3. As an event creator, I would like to set up the due date of the vote.
4. As an event creator, I would like to set up the due date reminder time for the vote.
5. As an event creator, I would like to cancel the event I created.
6. As a group member, I would like to receive an email notification if I get an invitation to the time availability vote.
7. As a group member, I would like to vote for the time slot for an event after I got the invitation from the event creator.
8. As a group member, I would like to change my voted time any time before the due date of the vote.
9. As a group member, I would like to get an email reminder on the due date reminder time.
10. As a group member, I would like to receive a notification about the final meeting time after the vote is done.
11. As a group member, I would like to receive a reminder one day before the meeting.
12. As a group member, I would like to be able to retrieve events I have voted.

## **Non-Functional Requirement**

### **Performance:**

We want to make sure that the users will have pleasurable experience with our application. Thus we need to assure that the responding time for each request (searching, web page loading, data filtering and etc.) is less than 500ms. Our final goal is to provide an efficient and user-friendly application

### **Security:**

Security is a significant aspect for all the online applications. As a user, I would like to prevent strangers to get access to my personal information especially phone number, address and etc. We plan to set restriction on users' passwords when users try to reset or set their passwords such as we might require a user's password is consisted of certain number of characters and digital numbers, both lowercase and uppercase alphabetic and better with some symbols. We also want to allow the user to change the password and information anytime, to retrieve their passwords when they forget it.

### **Development:**

As a developer, I would like to ensure that our framework is reusable so that we can add more new features while maintain the currently implemented features well in the future development.

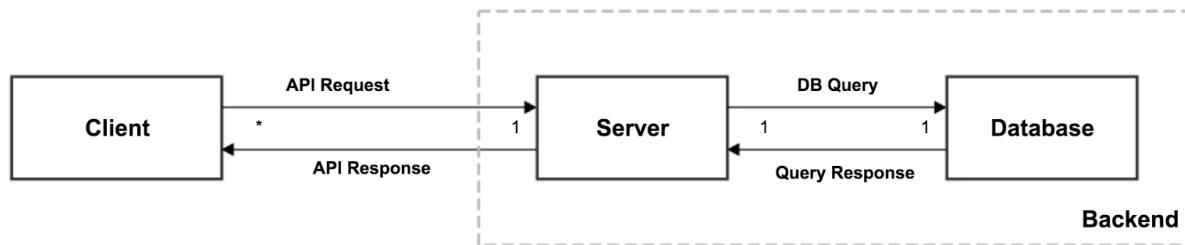
### **Appearance:**

Since our application's target users are Purdue students, we plan to add "Purdue feature" icons in our webpage to give Purdue students a more straightforward impression. We aim to provide a clean, creative and inspiring user interface. The UI design will be the most important for our web application since usually you will decide whether to use an application by its first impression.

# DESIGN OUTLINE

## High Level Overview

Our project will use the client-server model. The server will respond to the client request and the client will parse and render the response data from server using React. The figure below demonstrates a high-level overview of the system.

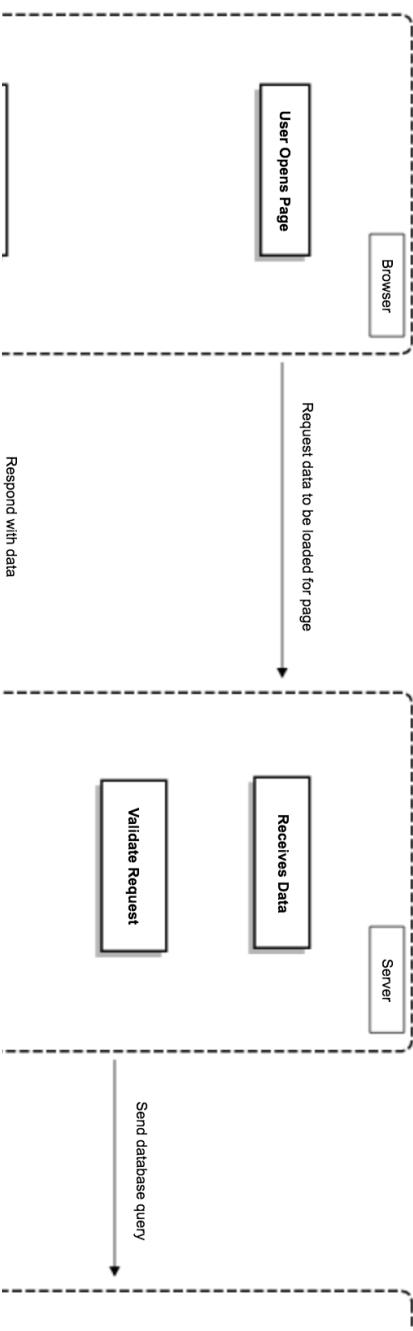


1. Client
  - a. Client sends AJAX requests to the server API
  - b. Client receives an AJAX response from the server
  - c. Response data is interpreted and rendered to the user using React
2. Server
  - a. Receives and handles all client requests
  - b. Validates requests before processing
  - c. Queries the database and sends appropriate response to client
3. Database
  - a. Database stores all the system information, such as users, classes, professors, events, etc...

## Flow of Events

The diagram below shows a typical flow of events. First, a user opens the web application on browser. When a client requests data from server in order to load the page, the client will make AJAX request to the server. On the server side, it first need to check the validity of the request from the user to make sure that requesting user has required permissions for such a request. In our app, we have split the users as normal users and admin users. So a normal user cannot load information that is only available to admin users. If the request is allowed, the server will query the database and the database will send a response object to the server. If the request is not allowed, no query will not be made and an error response will be sent to the client. After this, the client is able to parse and render the data onto the page.

(The image below will become clear when you zoom in and you could also use this [link](#) to visit it.)



# **DESIGN ISSUE**

## **Functional Issue**

### **1. Which page should be used as a welcome page?**

Option 1: Use main page as welcome page

Option 2: Design a new introduction page as welcome page

Decision: Option 1.

Compared with an introduction page, the main page more tends to give user a clear idea of the features and functionalities of the app, since it includes an abstract of the whole app as well as comprehensive information about three major functionalities. User will be prompted to set up an account once he / she knows what the app does and what benefits him / her.

### **2. Should we set up a quick login on the main page?**

Option 1: Yes

Option 2: No

Decision: Option 1.

A quick login link on the main page is designed for user to do a quick login without being forwarded to another page. User will stay on the same page and continue exploring the full features of the website after login from the main page.

### **3. When should user be asked for detailed personal information?**

Option 1: While setting up a new account

Option 2: While using one of three major features

Decision: Option 2

The detailed personal information required for each feature varies. In this case, we decide to ask user for detailed personal information while the user is exploring one of three major features. We only require basic personal information for setting up an account, including username, first name, last name, email address, password,

birthday and gender. The information required for three features are different. Group meeting focuses on user's time and transportation availability. Class registration requires information of user's school year, major, class, schedule, and academic preferences. Academic preference include but not limited to preference for professor, preference for class size, and preference for exam format. Looking for classmates focuses on user's schedule, personal information and expectation for classmate.

#### **4. How many pages should be used for filling in detailed personal information?**

Option 1: One page

Option 2: Two to three pages based on personal information categories

Decision: Option 2

We decide to use two to three pages for user to filling in detailed personal information. The format is similar to that of online job application. Each page will contain one or two major categories. For example, the first page will ask for basic information, including first name, last name, school, graduation date, email and GPA. These information may have been asked while setting up account. Our system will store user historical information and ask to fill in left questions. The second and third pages will vary based on different features. Using looking for classmates as an example. The second page will require information user personal information and personal preference, including identity information, and self-description. The third page will focus on user's expectation for classmates, including classmates' identities and description.

#### **5. In class registration, what type of feedback is prompted for a particular class or professor?**

Option 1: Class / professor rating

Option 2: Class / professor rating and comment

Decision: Option 2

User is prompted to rate class / professor and leave comment. The comment is used for user to provide any information about the class or professor that is not covered in the rating. Or user is free to leave any comments or concerns that he / she thinks that would be helpful for students who are going to take this class in the future.

The rating will ask user for satisfaction about the class or professor based on course organization, professor performance, homework and exams. User is able to leave as many comments as he / she wants, but the user must be responsible for the comment.

## **6. What type of switch among feature pages are we supposed to build?**

Option 1: Go back to the main page and forward to another feature page from main page

Option 2: Switch to another feature page directly from current feature page

Decision: Option 1

In order to unify the page layout and outlook, we would prompt user to go back to the main page and switch to another feature page. In order to do that, we will set up a return button returning to the main page, and user is able to switch to another feature page by simply clicking feature entry. This design will increase the independence of each feature page and increase the reusability at the same time. We may add more feature pages in the future, and we would like to minimize the influence of new added page to existing pages.

## **Non-functional Issue**

### **1. How to encrypt our data?**

Option 1: Do not do any encryptions

Option 2: Use hashing functions to hash the data

Option 3: Using encryption functions

Decision: Option 3

We decided to use some common encryption functions for example, identity-based encryption to ensure the safety of our data such as the user's account password.

## **2. Which web services will we choose to host our backend services?**

Option 1: Amazon Web Services

Option 2: Other kinds of services like VPS cloud

Option 3: Microsoft Azure

Option 4: Bluehost

Decision: Option 1

We chose Amazon Web Services for hosting our project's backend components because compared with other Web services, Amazon Web Service provides 365 days of free trial, which greatly decreases the cost for hosting the website. Also, Amazon Web Service is the quite popularly used as well as a quite well-known cloud system, it would make it easier to find help from the community.

## **3. Whether we need code review system?**

Option 1: Use code review system, ask colleague review codes before push to repo.

Option2: Do not use, and let members commit, and push without restriction.

Decision: Option 2

We decide not to use any code review system, because the code review system will slow down our develop life cycle. Since our project is not a very big one, which has only 6 people doing programming, it will be better for us to review the code on the Git repository and then make some tests together.

## **4. What platform will we select?**

Option 1: Windows/OS X platform specific application

Option2: IOS/Android/Windows Phone application

Option3: Web-based application

Decision: Option 3

We choose option 3 for the following three reasons:

1. Our user will not be limited by the platforms that they are using.
2. There are a lot of well-designed frameworks and resources, especially for frontend that we could use.
3. No prerequisite works are required for web-based application.
4. Web-based applications have been very popular recently.

## **5. Which frameworks are appropriate for implementing our backend services?**

Option 1: PHP native frameworks, like laravel, Zend.

Option 2: Python native frameworks, like Django, Flask.

Option 3: Ruby native frameworks, like Rails.

Option 4: Node.js

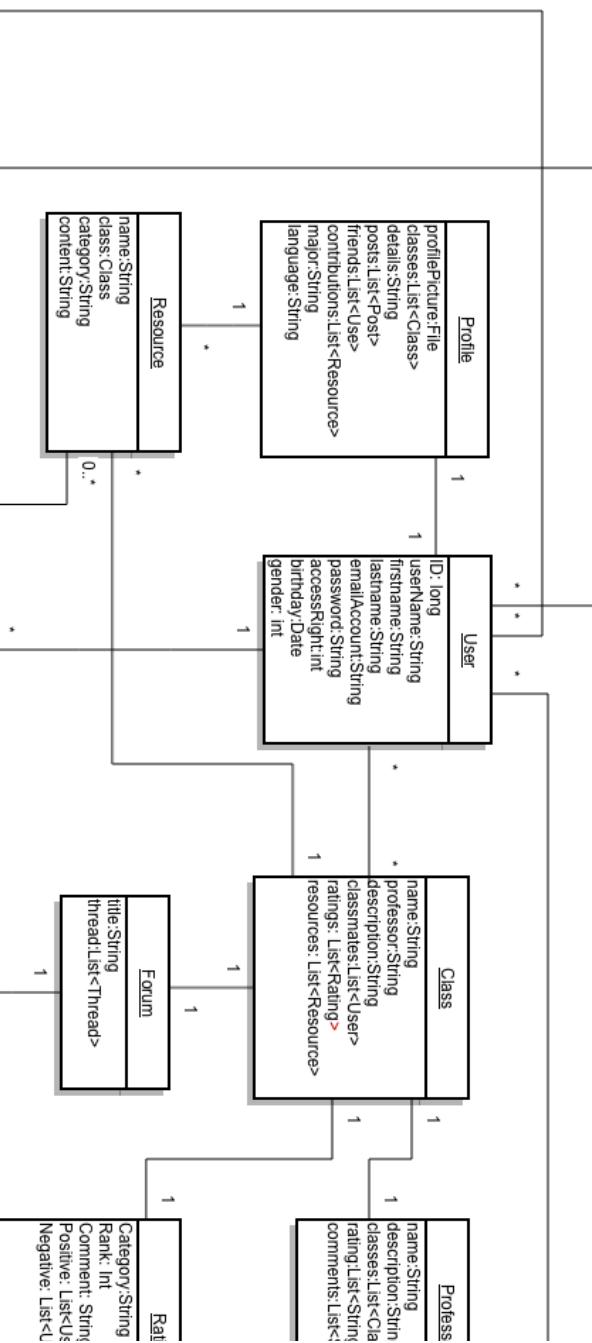
Decision: Option 4

At first, we decided to use Rails as our web framework. Because compared with PHP and Python, Rails tends to be a more traditional as well as popular web-develop framework. While Node.js is relatively young though popular as well, we decided to go with the traditional way of doing web development. But later, we found that developing our web app backend and frontend in a same code base could definitely make our development cycle more compatible and efficient, since we use javascript on our front end. Building an app using javascript on both backend and frontend is quite fascinating to our team developers. Thus, we changed our mind to return back to Node.

# **DESIGN DETAIL**

## **Data Classes Diagram**

(The image below will become clear when you zoom in and you could also use this [link](#) to visit it.)



## Data Classes Description

Above is a mockup of our database schema. Although things might change later when the app is under development, for example, we may come up with some new ideas or new features that we want to add, we believe this design we made will satisfy our current functional features.

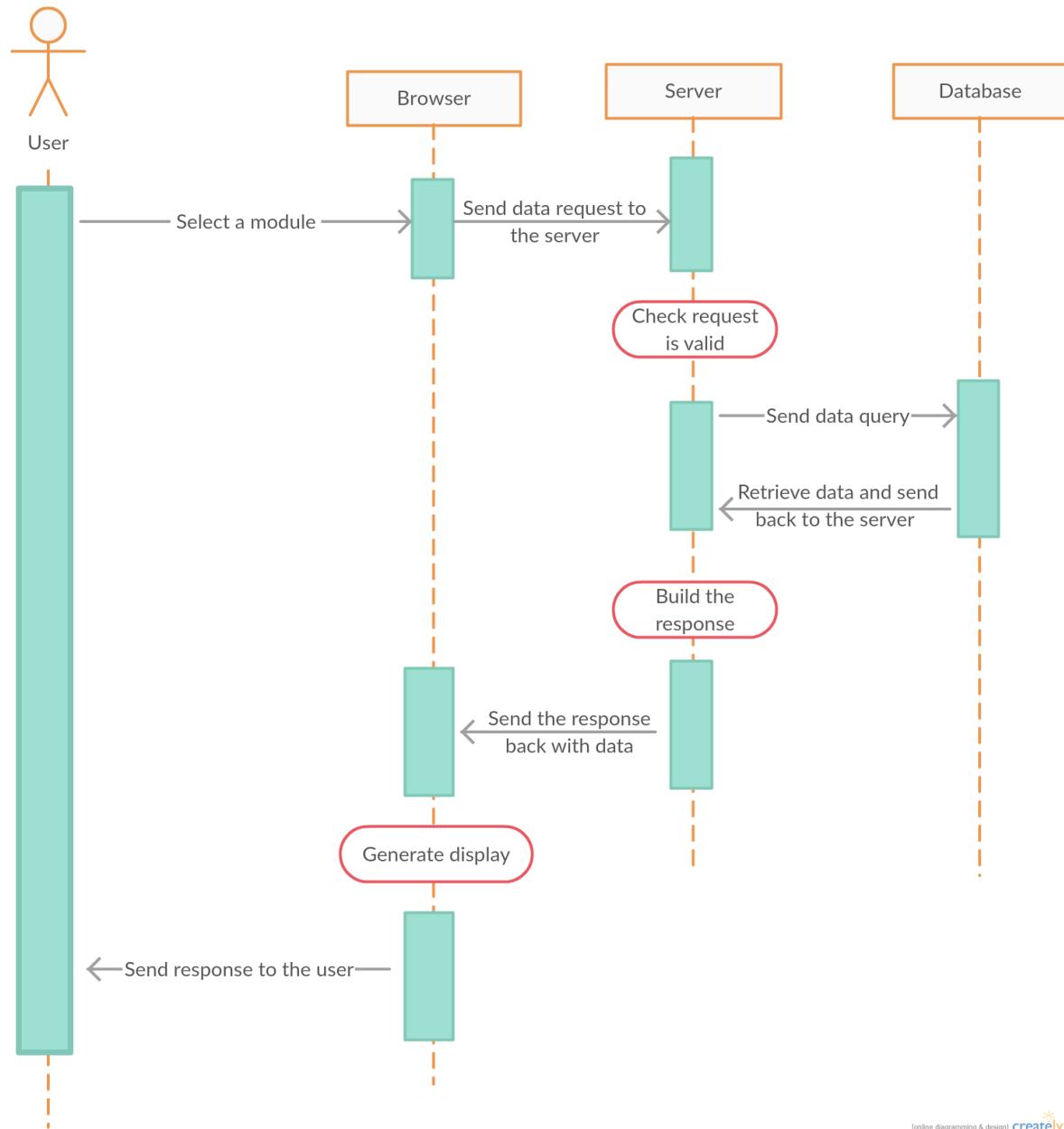
1. User
  - a. All users are included in this class includes admins and students
  - b. Has an accessRight attribute to mark the type of user
  - c. Contains basic information as username, email address, gender and birthday
2. Profile
  - a. Stores customized information about the user such as profile picture
  - b. Stores language user able to speak and major the user has to help other users to find classmates they might be interested in based on the common things they have
  - c. Stores courses the user has taken or currently taking for future checking whether user has right to write reviews for a course, since only a student who has taken the course could write reliable reviews for the course
  - d. Stores a friends list a user has since classmates might become friends after taking a course together
  - e. Stores resources a user has shared in different courses forum
3. Resource
  - a. Contains the course the resource is related to
  - b. Includes the content of the resource
4. Class
  - a. Stores relevant information about a course such as course name, course description, the professor who is teaching this course, and ratings related to this course
  - b. Contains students who have enrolled in this course as classmates
5. Professor

- a. Includes basic information about the professor such as name and description about the professor as his title and his teaching field
  - b. Associates a professor with courses he teaches
  - c. Gives access to external resources about the ratings related to the professor
6. Rating
- a. Defines different rating levels with different names such as a good, average and poor as category, rank
  - b. Includes detailed comment about this rating
  - c. Associates the rating with user who has made it as well as list of users who like this rating and who dislike this rating
7. Forum
- a. A forum covers three threads for users to discuss about different topics in each course. The three threads are “Welcome New Students”, “Random Chat”, “Academic Resources Sharing”
8. Thread
- a. A thread is an independent channel that students can have a common topic to discuss here. For example, in “Academic Resources Sharing” thread, students could only share resources here rather than do other things
  - b. Associate list of posts made by users
9. Post
- a. A post could either be a comment a user makes or resources a user shares in “Academic Resources Sharing” thread.
10. *Group*
- a. A group consists of a group of users who need to do group work and schedule group meetings
  - b. Includes information relevant to the group work such as the group chat messages as well as group scheduled events
11. Event

- a. Event includes information associated with a group event such as the time, location and content for a group meeting

# Sequence Diagram

iStudy Sequence Diagram



[online diagramming & design] [creately.com](#)

This diagram shows a sequence of events among the user, browser, server and database while using iStudy.

Request is first made from the user by selecting any component on the web, and then the request was sent to the server. The request sent to the server will be checked is valid first and then valid request will be processed by the server which retrieves the data needed from the database. Invalid request will get an error message without data from the database. After data query, the server will generate a response to the browser which then will use it to generate display for the user.

## UI Mockups

Main page:

1

This UI mockup shows a main page for 'iStudy'. In the top left corner, there is a small image of a cartoon character holding a hammer. To the right of the character, the text 'Welcome to iStudy' is displayed in a large, bold font. Below this, there are two links: 'login' and 'sign up'. A yellow box with the text 'link this component via drag-and-drop to a screen' is overlaid on the character's image. The central part of the page features a 3D-style illustration of several orange humanoid figures sitting around a white table, engaged in a meeting. At the bottom of the page, there is a grey footer bar containing three buttons: 'meeting', 'class registration suggestion', and 'find classmates'.

2

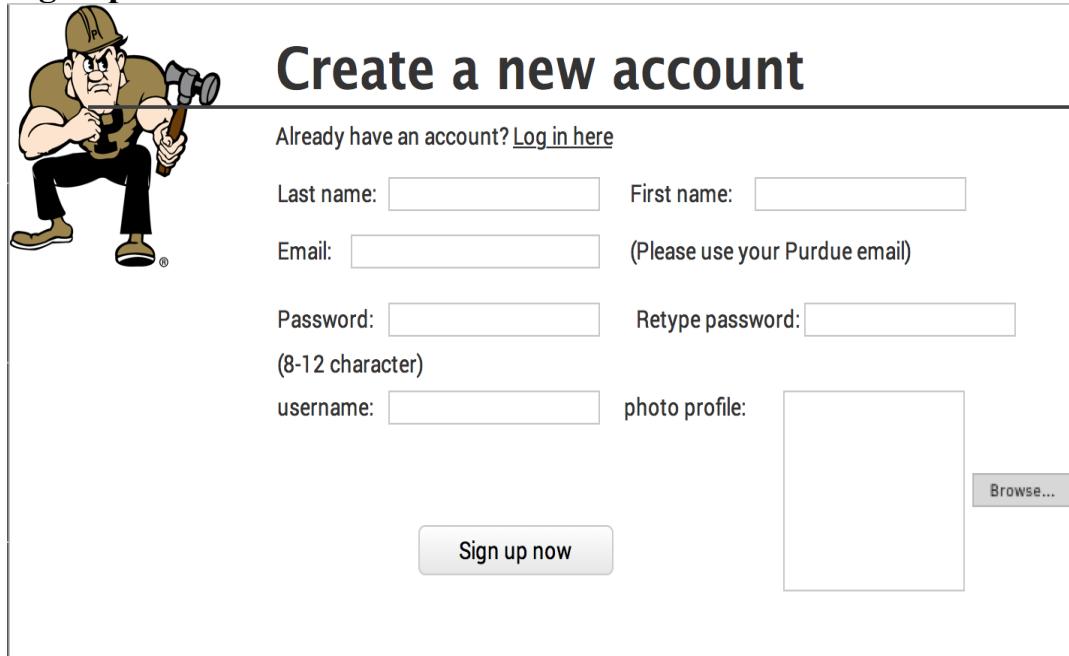
This UI mockup shows a main page for 'iStudy'. It features the same layout as UI Mockup 1, with the cartoon character, 'Welcome to iStudy' text, and 'login'/'sign up' links. However, the central image has changed. Instead of the meeting scene, it now shows a clipboard with a checklist and a pencil writing on it, with the text 'Registration Information' written in a large, stylized font next to it. Below this central image are three small circular dots. The footer bar at the bottom contains the same three buttons as UI Mockup 1: 'meeting', 'class registration suggestion', and 'find classmates'.

3



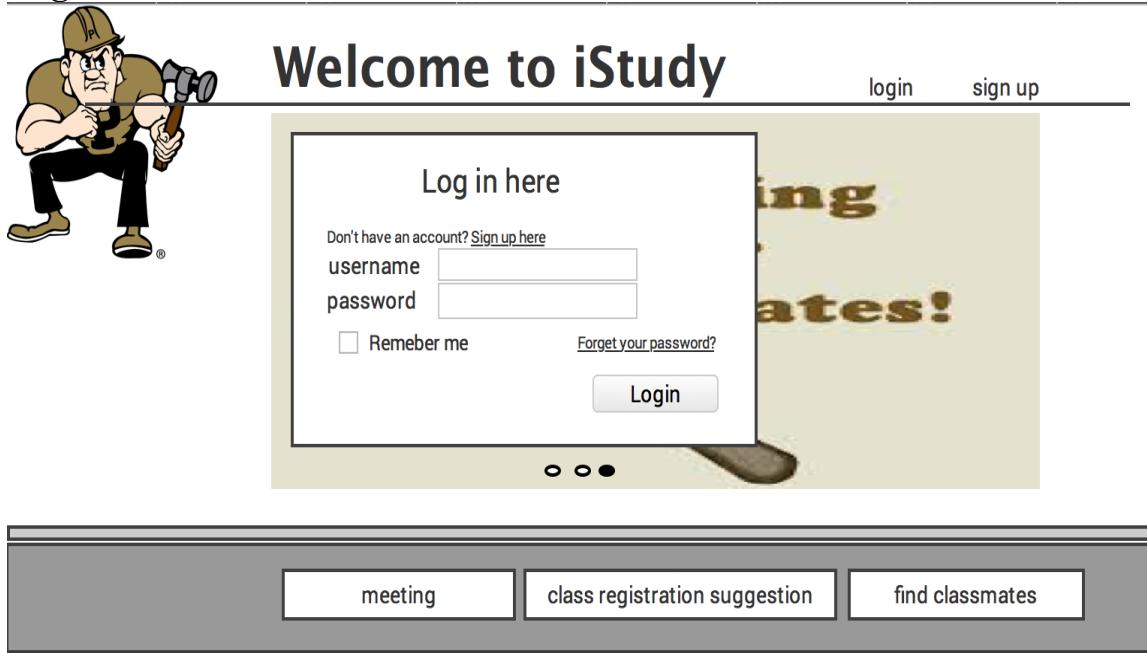
The image shows the homepage of iStudy. At the top left is a cartoon character of a construction worker in a hard hat and vest, holding a hammer. To the right of the character is the text "Welcome to iStudy" in a large, bold, black font. Below this, there are two links: "login" and "sign up". The central part of the page features a large magnifying glass over the words "Looking for Classmates!". Below the magnifying glass are three small dots. At the bottom of the page is a grey horizontal bar containing three buttons: "meeting", "class registration suggestion", and "find classmates".

Sign up:



The image shows the "Create a new account" page. On the left is the same cartoon construction worker character. The main title is "Create a new account". Below the title is a link "Already have an account? [Log in here](#)". There are several input fields: "Last name:  First name: ", "Email:  (Please use your Purdue email)", "Password:  Retype password:  (8-12 character)", "username:  photo profile: ", and a "Browse..." button next to the profile input field. At the bottom is a "Sign up now" button.

## Login:



The image shows the login page of the iStudy website. At the top left is a cartoon character of a construction worker holding a hammer. The main title "Welcome to iStudy" is centered above a login form. The login form has fields for "username" and "password", a "Remember me" checkbox, and a "Login" button. Below the form is a "Forgot your password?" link. At the top right are "login" and "sign up" links. The background features a large, faint watermark of the word "ates!".

Log in here

Don't have an account? [Sign up here](#)

username

password

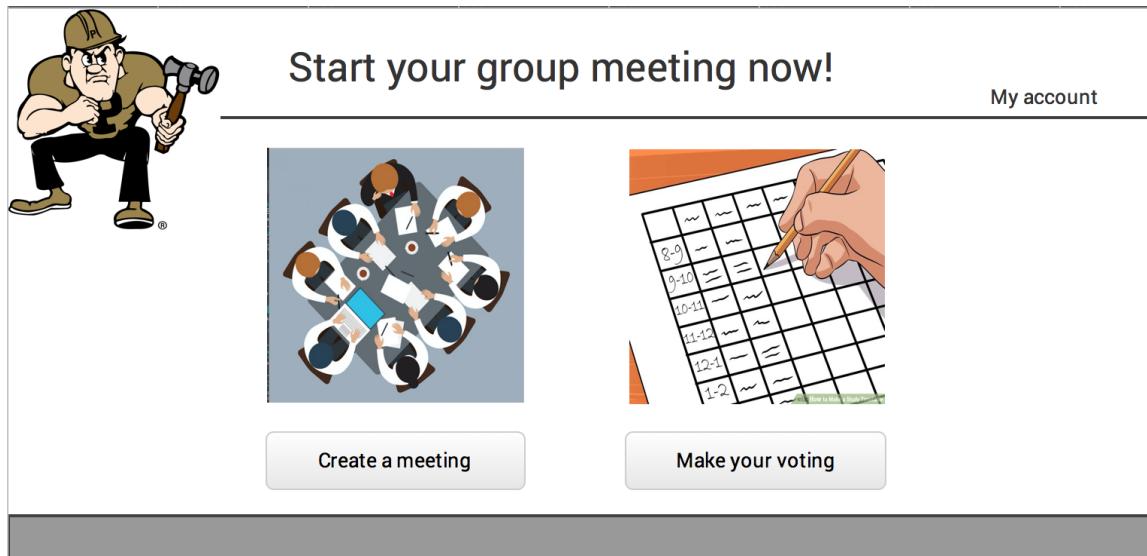
Remember me [Forgot your password?](#)

[Login](#)

ing  
ates!

meeting class registration suggestion find classmates

## Set up meeting:



The image shows the meeting setup page. On the left is the same cartoon character. The main heading is "Start your group meeting now!". To the right is a "My account" link. Below the heading are two images: one showing a group of people in a meeting and another showing a hand marking a grid. At the bottom are two buttons: "Create a meeting" and "Make your voting".

Start your group meeting now!

My account

Create a meeting

Make your voting

## Registration Suggestion:



## Need suggestion for registering?

My account



[know more about courses](#)

[know more about professor](#)

### Find your classmates:



## Find your Classmates

Class Title:  Class Number:

Preferred Language:

Preferred gender:  Male  Female

Preferred class standing:  Freshman  Sophomore  
 Junior  Senior

[Search your pair](#)