

# PurdueParty.io

Design Document - CS407 Team 4

*Aaron Cox, Yu-Lun Gao, Michael Maganini, Raziq Ramli, Matthew Story, Bryan Wang*

# **Index**

## **1. Purpose**

- a. Introduction
- b. Project Objectives
- c. Requirements

## **2. Design Outline**

- a. Client-Server Model
- b. API Calls
- c. User Interface Experience

## **3. Design Issues**

- a. Functional Issues
- b. Non-Functional Issues

## **4. Design Details**

- a. Data Design (Class Level)
- b. Data Models/Classes Description
- c. Sequence Diagrams
- d. UI Mockups

# **Purpose**

## **Introduction**

Currently, Purdue students find themselves flipping between several different applications to meet their needs - an app for the Corec, websites for buying used goods, another for social media, and even sometimes plain word of mouth to figure out what the dining courts are serving. Our solution, PurdueParty.io, will be an all-in-one website where students can view and post all information related to clubs, events, dining halls, the CoRec, transit, used goods, class group chats, and much more. The site will even include an area where students can be a part of social media as well as find and post useful information related to campus. Unlike other projects in this space that are limited in their feature set, our app will be a one-stop-shop for all of a Purdue student's needs.

## **Project Objectives**

- Program a React application for Purdue students that provides a multitude of relevant features for students.
- Include features from popular social media platforms, including:
  - Reddit Forum - Class/Major Chats - Events - Clubs Page
- Include useful information about campus utilities and services, including:
  - Gym usage - Dining Halls usage/menus - Bus Maps
- Include a platform that allows students to easily sell and purchase items/services amongst other Boilermakers

## **Functional Requirements**

### **Profile**

1. As a user, I would like to create a new account with my own email address.
2. As a user, I would like to edit my profile.
3. As a user, I would like to delete my account.
4. As a user, I would like to have a unique username.
5. As a user, I would like to have a profile with a brief bio and image.
6. As a user, I would like to be able to change my password for my account.
7. As a user, I would like to be able to log in.
8. As a user, I would like my password to be obscured when I type it to log in.
9. As a user, I would like to be able to log out.
10. As a user, I would like to reset my password if I forget it.
11. As a user, I would like to check other user profiles.
12. As a user, I would like to choose between light and dark mode (if time allows).
13. As a user, I would like to hide my profile from users (if time allows).

14. As a user, I would like to choose specific features from the site as part of my homepage (if time allows).

### **Events Board**

15. As a user, I would like to be able to add events for my Purdue club if I have club permission.
16. As a user, I would like to be able to create public personal events.
17. As a user, I would like to be able to edit an event I created.
18. As a user, I would like to be able to delete an event I created.
19. As a user, I would like to be able to view a list of Purdue events.
20. As a user, I would like to be able to view public personal events (if time allows).
21. As a user, I would like each event to have an information page.
22. As a user, I would like to be able to filter some events based on some criterias.
23. As a user, I would like to be able to RSVP to club events (if time allows).
24. As a user, I would like to see past events be hidden from my feed (if time allows).
25. As a user, I would like to see club events I create added to that club page if I have permission.
26. As a user, I would like to see the events posted on the boiler links website to be listed on this application as well (if time allows).

### **Club Pages**

27. As a user, I would like to be able to create a new club.
28. As a user, I would like to be able to manage user permissions of the club if I am a creator.
29. As a user, I would like to be able to add a description for my club.
30. As a user, I would like to be able to add contact information for the club administrators.
31. As a user, I would like to be able to view a list of clubs available at Purdue.
32. As a user, I would like to be able to search the list of Purdue clubs.
33. As a user, I would like to view relevant information about clubs I'm interested in.
34. As a user, I would like to be able to edit a club's information if I have necessary permission.
35. As a user, I would like events that I make to be automatically added into the events board.
36. As a user, I would like to have access to edit the events that automatically added into the events board.
37. As a user, I would like to have access to delete the events that automatically added into the events board.
38. As a user, I would like to see a list of events that a club is holding on their page.
39. As a user, I would like to see the clubs created in the boiler links official website to be listed on this application as well (if time allows)

### **Living**

- 40. As a user, I would like to be able to view a labeled map with bus stops and routes that run through the Purdue campus.
- 41. As a user, I would like to be able to view a bus route's next stop.
- 42. As a user, I would like to be able to view a list of all Purdue dining halls.
- 43. As a user, I would like to be able to view daily menus for each dining hall.
- 44. As a user, I would like to be able to check how crowded each dining hall is.
- 45. As a user, I would like to be able to check how crowded different sections of the Purdue CoRec are.
- 46. As a user, I would like to be able to see which washing machines are currently in use in each dormitory.
- 47. As a user, I would like to be able to see which dryers are currently in use in each dormitory.

### **Marketplace**

- 48. As a user, I would like to be able to create sell listings on the marketplace.
- 49. As a user, I would like to be able to see a list of sell listings on the marketplace.
- 50. As a user, I would like to edit my sell listing on the marketplace.
- 51. As a user, I would like to delete my sell listing on the market place.
- 52. As a user, I would like to be able to set a sell price and item description for my sell listings.
- 53. As a user, I would like to be able to list my contact information to potential buyers.
- 54. As a user, I would like to be able to include an image of the item I am selling in my sell listings (if time allows).
- 55. As a user, I would like to be able to message other students on the marketplace about their listings (if time allows).

### **Class Forum**

- 56. As a user, I would like to be able to view class forums.
- 57. As a user, I would like to post on class forums.
- 58. As a user, I would like to edit my post.
- 59. As a user, I would like to delete my post.
- 60. As a user, I would like to comment on posts.
- 61. As a user, I would like to comment on comments.
- 62. As a user, I would like to edit my comment.
- 63. As a user, I would like to delete my comment.
- 64. As a user, I would like to filter posts by popularity or recency (if time allows).
- 65. As a user, I would like to join class forums (if time allows).
- 66. As a user, I would like to leave class forums (if time allows).

### **Other**

- 67. As a user, I would like to install this web application on my desktop and mobile devices as a progressive web app (if time allows).

## **Non-Functional Requirements**

- Our application will work across different browsers.
- Our application will have an intuitive UI for users to easily navigate the site.
- Our application will be visually appealing.
- Our application will have fast response and load times.
- Our application will be reasonably secure for storing user information.
- Our application will be developed in a way that code modification and testability is practical.

### **Performance**

In order to give users a pleasant experience using PurdueParty.io, most of the web pages should be fully loaded in, at most, 3 seconds. Pages that end up embedded, such as the labeled map with bus stops, may take up to 5 seconds.

### **Scalability**

We intend to use Firebase as both our hosting platform and database service. As a result, we have a variety of options in terms of scalability. During initial development, we will have access to a Gigabyte of storage, 100 simultaneous connections, and 360 MB of data transfer a day. This should be more than enough to meet our needs when testing with a small subset of Purdue's campus population. When it is time to deploy production code and expand our user base, Firebase provides scalable and affordable payment options that will be proportionate to our usage. Given that Firebase is a Google product intended for large-scale applications, it can be safely assumed their servers will be able to meet the demand for our product.

### **Security**

PurdueParty.io primarily handles non-sensitive information. However, due to our planned implementation of account registration, privilege-level editing, and more, we still need to securely protect both user information and our database. To that end, we intend to use Firebase, our chosen backend infrastructure, to enforce strict security across the site. Notably, it uses a SSL certificate for our domain to ensure that data is transmitted securely. Additionally, we intend to utilize Firebase's database rules to allow users to only access and modify data which is pertinent to them, such as club information if they created that specific club. We also hope to provide users the ability to hide their profile information from other users if they have privacy concerns.

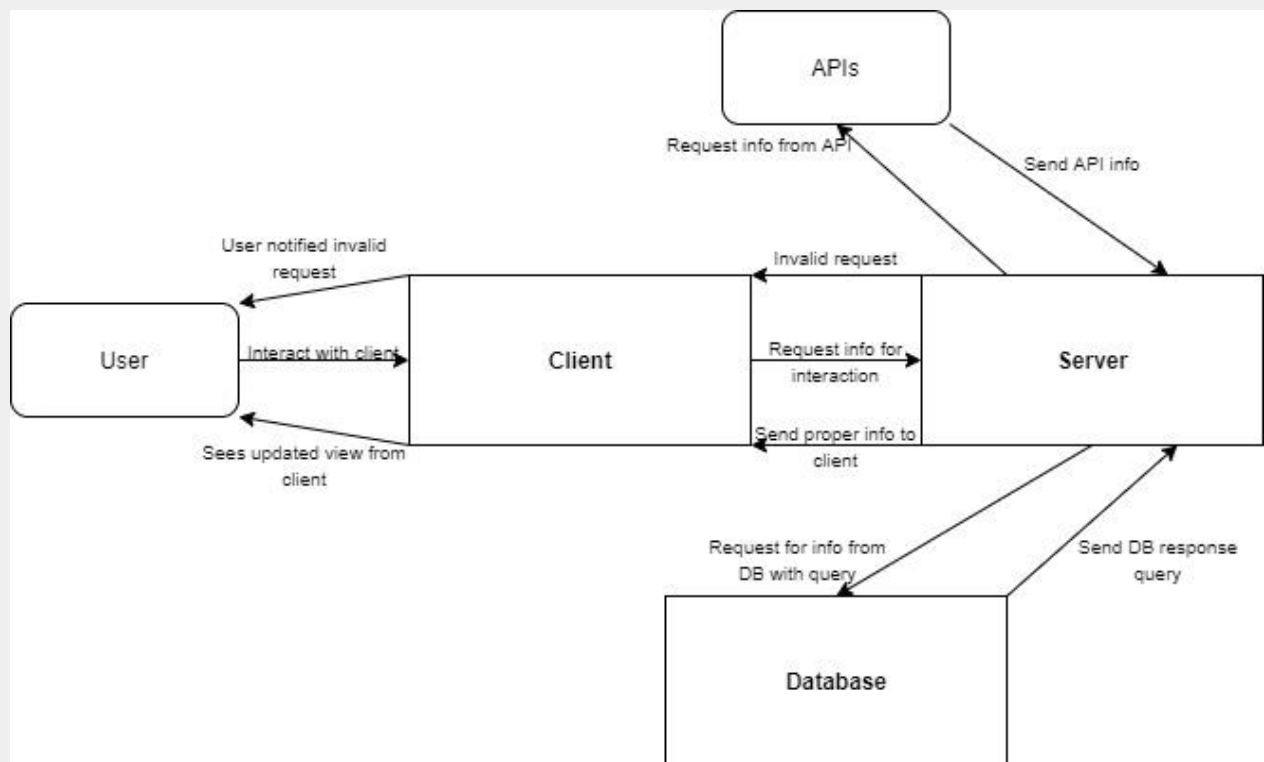
### **Usability**

We want to design the application in a way such that it will be easy for the user to be aware of the categories of features that we are providing and where to locate them. To ensure that users will be able to parse and read the information presented on our application well, we will choose suitable fonts and font sizes and a color scheme that provides good contrast for our texts according to WCAG guidelines.

## **Design Outline**

The Purdue Party web app will follow the client-server model. The user will interact with the client which will interact with the server, and the server will interact with the database and APIs.

### Client-Server Model UML Diagram:



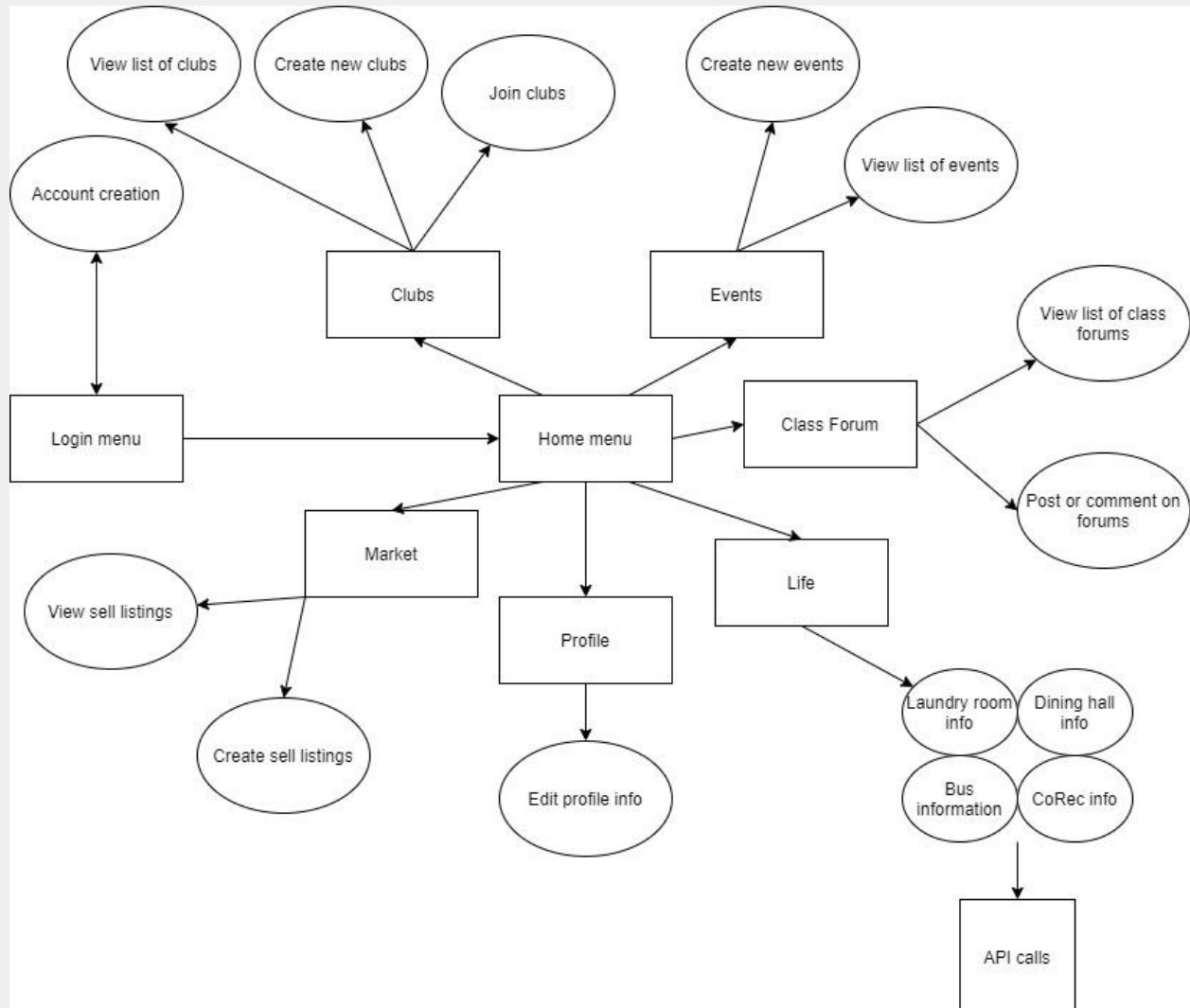
When the user interacts with the app, the client sends the appropriate request to the server. Based on the request, the server will deny the request or it will store/retrieve information from the database and/or APIs through database queries or API calls. The server collects this information then will send a response back to the client. The user interacts with the client and will see the response via some change in the UI. For the front-end, we will be using React.js to create the client side of the webapp. For the back-end server and database, we will be using Redux and the Firebase database.

### API Calls:

We will be making API calls to collect information from different Purdue services for information that will not be user inputted and stored in the database. For Purdue dining halls, we will be collecting data for daily breakfast, lunch, and dinner menus. For Purdue bus stops and routes, we will be collecting CityBus data. For the Purdue CoRec, we will be collecting data on

the level of crowdedness of each section of the rec center. Finally, we will be collecting data on laundry room availability for each student dormitory. This will include which washers and dryers are open for students to use. The database will store user inputted information such as profile information, forum posts and comments, and club information and events.

## UI Experience:



After a user either creates an account or logs in with an existing account, the user is taken to the home menu. The menu has a hotbar that navigates them to each of the six sections of Purdue Party. First, a user can view a list of clubs. They can create a club and other users can join said club. This info is stored in the database. Second, a user can navigate to a page that shows a list of events pulled from the database. Users can also create new events from this page which will be stored in the database. Third, a user can navigate to a class forum page where they can view a list of different class forum pages. Users can also post and comment on the different forums. Forums, comments, and posts are stored in the database. Fourth, users



can navigate to a market where they can view a list of sell listings and where they can create their own sell listings. Fifth, users can navigate to a profile page where they can add an image and a bio. Both profile info and market sell listings are stored in the database. Finally, users can navigate to a life section. They can view real-time information on availability of the CoRec and washers and dryers in dorms, bus stops and routes, and dining hall menus. API calls are made in order to provide the necessary data in the life section.

# Design Issues

## Functional Issues

### **1. Will all users be able to modify/add to all club pages?**

- a. All users will be able to modify or add to any club page
- b. Only club owners and administrators can modify their own club pages**
- c. Club pages will not be modified after their creation

Although club information is sporadic and having all users possess modification ability could help with keeping the pages updated, we decided to only let the creator of the club page and its designated administrators modify the page. We chose this option to prevent targeted attacks towards clubs, and to allow its board more explicit control of the content that gets added. Additionally, if a user wants to help with the management of a club page, all they have to do is request admin access from its owner.

### **2. How will the events page be ordered?**

- a. Chronologically, with the soonest events shown first**
- b. By popularity, with the most popular events shown first
- c. By most recently added

We have decided to order the events in chronological order. By doing so, users will be less likely to miss events that they have just discovered. This also allows for users to plan ahead by scrolling down the list in order to view events that are happening further in the future. While some users may prefer the most popular events being presented first, they will be able to discover popular events in their areas of interest through the forum feature. On the other hand, showing events in the order they are added may lead to confusion as a user navigates the page.

### **3. How should users navigate to the forum pages they are interested in?**

- a. By searching for the forum name in the general forum list
- b. By only viewing forums they explicitly favorite
- c. By viewing their favorited forums first, before the others in the list**

Trying to search or scroll through the forum list to find a user's relevant pages can be a cumbersome task, generally diminishing that user's experience. As a result, users will have the ability to "favorite" their desired forums in order to navigate to them

quickly and easily after discovery. Furthermore, presenting users with only the forums they have favorited could cause frustrations when trying to discover new forums.

#### **4. Who will be able to post items to the Marketplace?**

- a. Logged-in users**
- b. Anybody with access to the site

Given that our user base is exclusively Purdue students, only logged-in users will be allowed to post items to our marketplace. This will help prevent unwanted items that are not relevant to Purdue students from crowding the site. Additionally, enforcing this action will discourage individuals outside of our target audience to post things because they will have to go to the trouble of making an account.

#### **5. What information should be included in a user's Profile?**

- a. Username + Bio**
- b. Username + Bio + Address
- c. Username + Bio + Address + Social Media Links

Providing unnecessary information can be concerning to a prospective user from a security standpoint, and since we want to make our app welcoming to all Purdue students, only a username and bio will be needed for an account. On the other hand, if a user wishes to provide such personal information, they will have the opportunity to provide it in their bio. This implementation also prevents development bloat as well.

#### **6. How should users navigate the “living” page?**

- a. No navigation present; all info presented on one screen
- b. Through a top navigation bar that redirects the user
- c. Via a sidebar that loads the relevant information within the parent page**

We decided to implement the living page with a sidebar that loads the desired information within the parent page. This will make our UI feel snappy and crisp to the user, which is extremely important for our application. A top navigation bar that redirects the user would take away from this performance. Moreover, the sidebar implementation will reduce clutter and allow for a better experience than shoving all data - laundry, city-bus, corec, and dining courts - onto one screen.

## Non-Functional Issues

### 1. What platform best serves Purdue students (our user base)?

- a. Android
- b. iOS
- c. **Web**

We decided to make our project a web-based application due to the nature of our user base. Purdue is home to tens of thousands of diverse students who each have their own preference regarding mobile platform. In order to keep our user base from splintering along those platform lines, we chose to develop our app for the web to ensure that it is available to each and every student. Additionally, given that many Purdue stats come from web-pages of their own, a web app allows us to more easily integrate that constantly-updating information.

### 2. What type of architecture will our app use?

- a. **Client Server**
- b. Client Serverless
- c. Unified Architecture

We decided to use the client server architecture for this project. While not a perfect solution, the model fits our chosen technologies the best out of all the options. Additionally, it allows us to store our data in a central location, something that is integral when it comes to real-time club or event updates. Finally, it allows us to easily divide work into front-end focused activities and back-end focused activities, ensuring smoother development.

### 3. Which framework should be used for the front-end?

- a. Angular
- b. **React.js**
- c. Vue.js

In order to facilitate smooth development, we chose React as our front-end framework. Several of our members already have experience with React, allowing us to focus more of our resources on usability and performance. Additionally, this prevents us from having to learn new frameworks - Angular has a reputation of a steep learning curve and Vue has a smaller community to learn from. Finally, React has easy integration with our back-end to allow for seamless development.

#### **4. Which back-end framework should be used?**

- a. MongoDB + Node.js
- b. MySQL + Node.js
- c. Firebase + Redux**

Firebase + Redux is our chosen back-end framework for a multitude of reasons. Similar to the front-end, we have a few members with experience on the platform, allowing us to better allocate resources to the user experience. Additionally, Firebase facilitates easy user authentication, database storage, and security, all of which are critical to our application. Finally, the Redux approach allows for a tighter-knit React project, facilitating smoother communication and development between the front-end developers and the back-end developers.

#### **5. What model should we rely on for collecting campus-related data (corec traffic, dining usage, and laundry availability)?**

- a. User Input Data
- b. Various API's
- c. User Input Data + API's**

We have decided to use both API integration as well as rely on user-input data in order to provide accurate information about Purdue facilities. While API's alone serve as the best option for ease-of-development and up to date information, not all of Purdue's facilities provide such an API. In those circumstances, and when the API themselves become unresponsive, we know that we will still have a method for collecting some data points about Purdue facilities. Additionally, given that our app is targeted towards Purdue students who use some or all of these facilities on a daily basis, it can be safely assumed that we will have a reliable data stream.

#### **6. How should we collect city-bus data?**

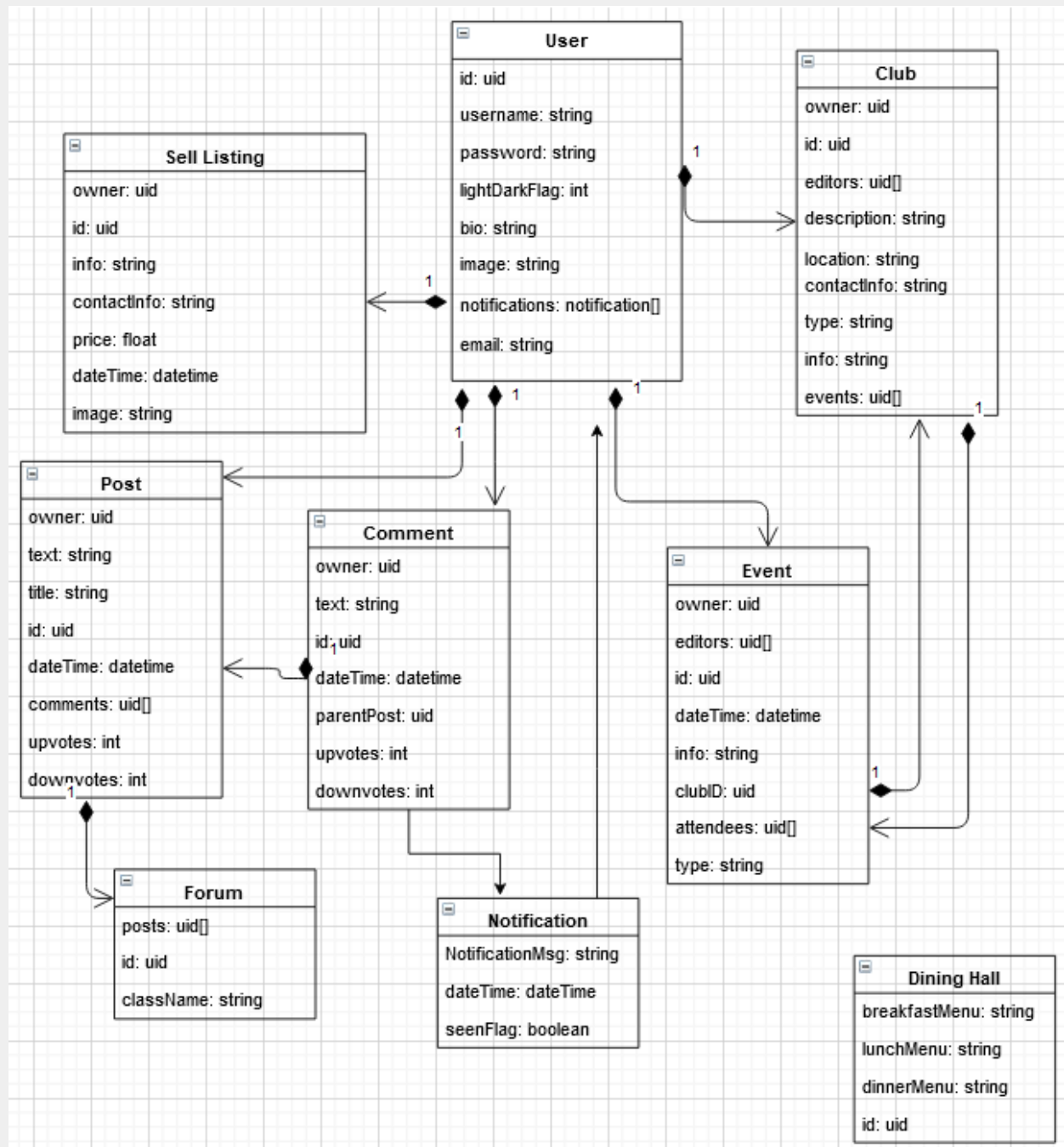
- a. User Input Data
- b. City Bus API**
- c. User Input Data + API's

We have decided to use only the city-bus API for bus routes, timing, and capacity. In this circumstance, user data isn't as reliable because the buses aren't technically a part of Purdue's own infrastructure and, as a result, are not used as predictably as other campus facilities. Furthermore, the city bus API provides all of the aforementioned features, allowing for easy integration and development into our application.

# Design Details

## Class Level Design:

### a) Class Level Design Diagrams



## **b) Class and Interaction Descriptions**

### **1) User**

- Every PurdueParty.IO user will have a set of values to distinguish themselves from other users.
- To identify the user, every user is given a unique uid, or user id.
- Each user also has a username and password, which are used to authenticate the user.
- The user has information such as bio, image, email, and the lightDarkFlag to customize their profile and provide information about themselves.
- A user has a set of notifications from any posts or comments they have made.
- A user can make posts or comments and interact with the forum or create events for a club as an owner. They can also create a sell listing.

### **2) Sell Listing**

- Each sell listing has an owner uid, which corresponds to the user that created it.
- A sell listing has a unique uid to identify itself from the other sell listings.
- The sell listing contains info, contact info, a price, and an image to display various information about what is being sold.
- A sell listing contains a dateTime to display when the sell listing was created by a user.

### **3) Club**

- Each club has an owner uid, which corresponds to the user that created it or the user that has been given the permissions of owner.
- A club has a unique uid to identify itself from the other clubs.
- Each club has fields to provide general information and construct the page displaying said information, such as description, location, contactInfo, type, and info.
- The uids of each user that can edit this club's fields is stored to help handle permissions.
- Each club has a list of events that are created by a user and correspond to the club.

### **4) Comment**

- Each comment has an owner uid, which corresponds to the user that created it.
- A comment has a unique uid to identify itself from the other comments.
- A comment contains a dateTime to display when the comment was created by a user.

- Each comment contains a field of text that contains the body of the comment.
- A parent post id is provided to ensure that comments are placed under their proper corresponding posts.
- Each comment has an integer that corresponds to upvotes and one for downvotes to determine the popularity of the comment.

#### **5) Post**

- Each post has an owner uid, which corresponds to the user that created it.
- A post has a unique uid to identify itself from the other posts.
- A post contains a dateTime to display when the post was created by a user.
- Each post contains a field of text that contains the body of the post.
- Each post has an integer that corresponds to upvotes and one for downvotes to determine the popularity of the post.
- A post has a title to provide a brief description of the post.
- An array of comments are stored under each post for easy access to the corresponding comments of each post.

#### **6) Event**

- Each event has an owner uid, which corresponds to the user that created it or the user that has been given the permissions of owner.
- An event has a unique uid to identify itself from the other events.
- The id of the club that the event corresponds to is stored in the event.
- Each event has fields to provide general information and construct the page displaying said information, such as type and info.
- The uids of each user that can edit this event's fields is stored to help handle permissions.
- Each event has a list of attendees that are going to said event.
- An event contains a dateTime to display when the event is to occur.

#### **7) Notification**

- Each notification contains a NotificationMsg that displays the message associated with comments made to a user's post.
- A notification contains a dateTime to display when a comment was made to a user's post.
- A boolean seenFlg field is created to determine if the notification has been seen by a user or not.

#### **8) Dining Hall**

- Each dining hall contains strings for the breakfast, lunch, and dinner menus.
- A dining hall has a unique uid to identify itself from the other dining halls.



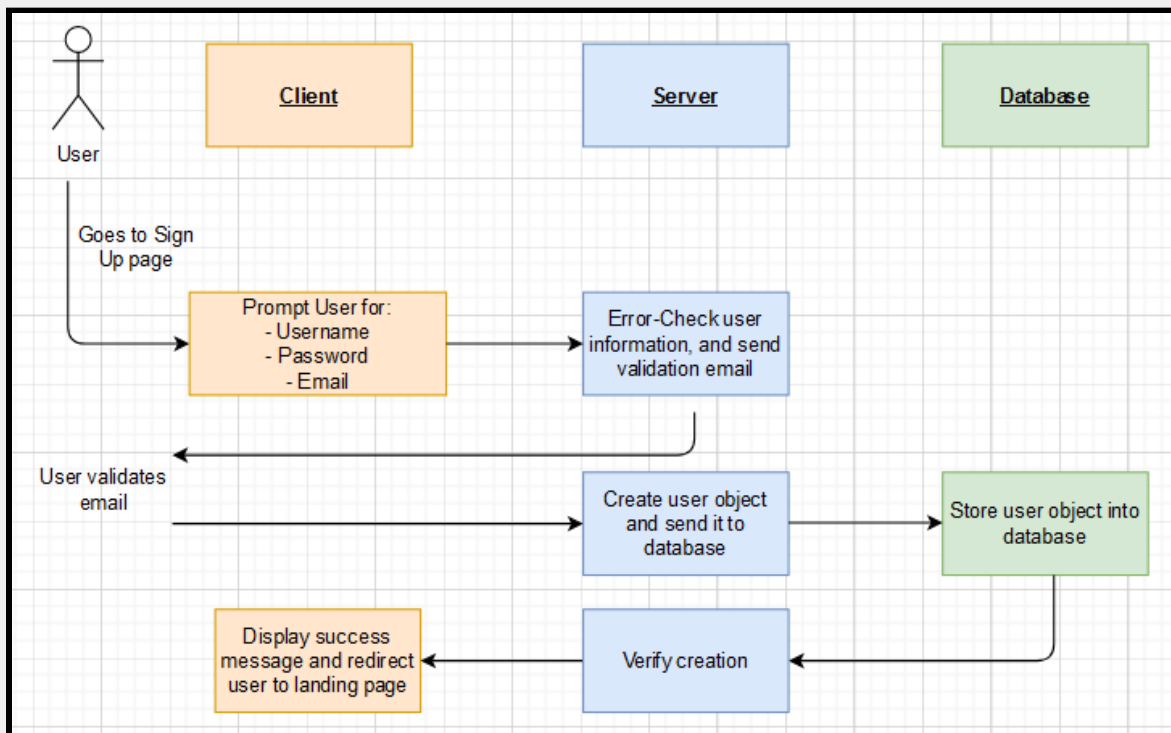
## 9) Forum

- A forum has a unique uid to identify itself from the other forums.
- Each forum has a className string to distinguish which class the forum is associated with.
- A forum contains an array of posts that are associated with that specific class.

## c) Sequence of Events Diagrams

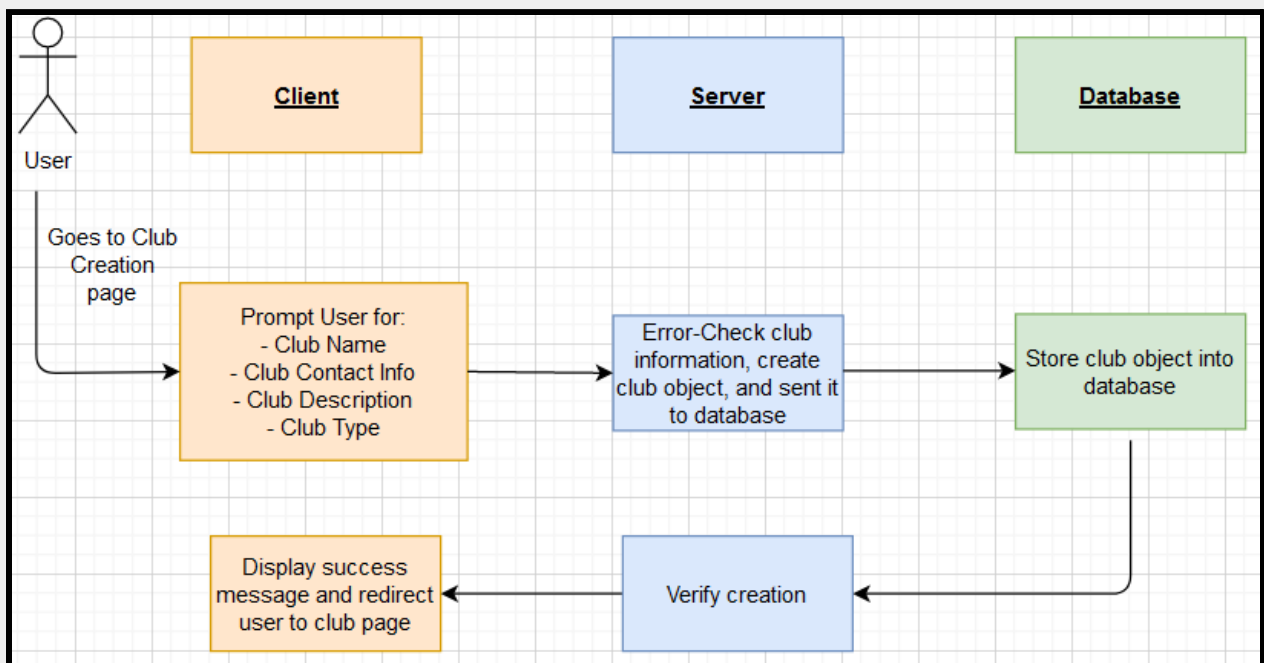
### 1. Create user account

A user goes to the “Sign-Up” page on the landing screen. They are then prompted to enter an email address, username, and password. The inputted information is checked for validity and a user object is created. An email is sent for verification purposes to the email. Once the email has been verified, the object is then stored into the database. The server verifies if the user object was successfully created and in the database. Then, the user is redirected into the website homepage.



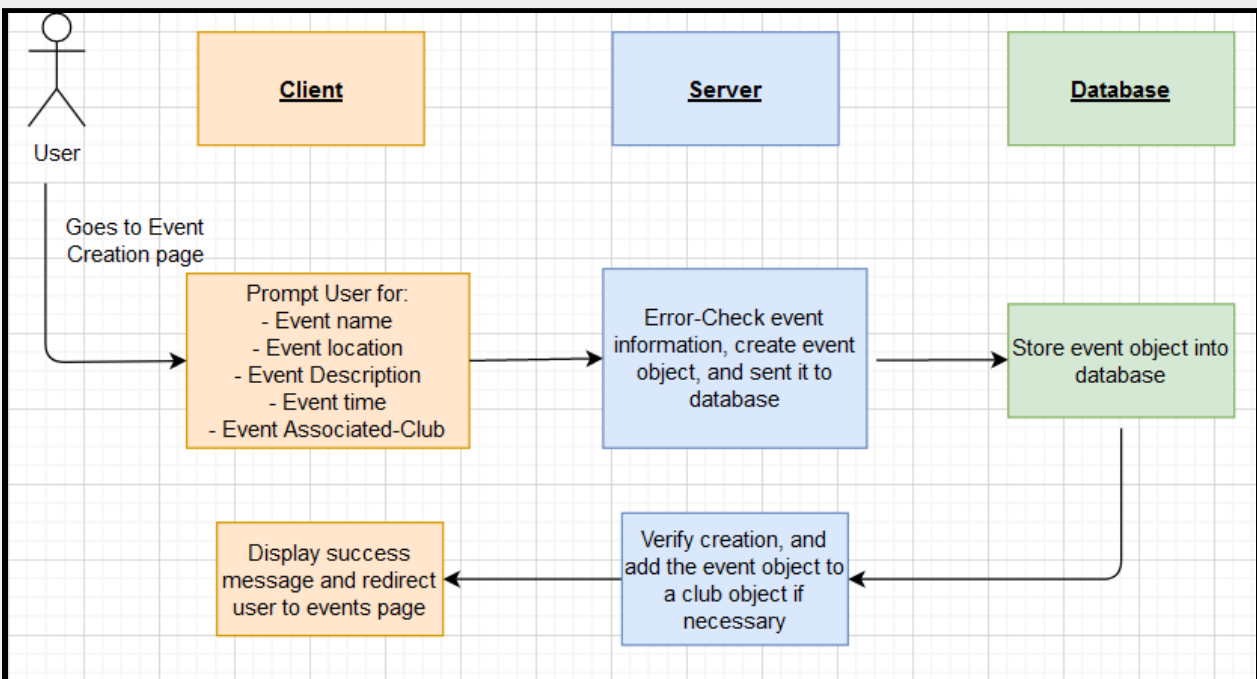
## 2. Create club page

A user goes to the “Create Club” page on the clubs screen. They are then prompted to enter a club name, club contact info, club description, and a club type/category. The inputted information is then checked for validity and a club object is created. The object is then stored into the database. The server then verifies if the club object has been successfully entered into the database and the user is redirected into a club page.



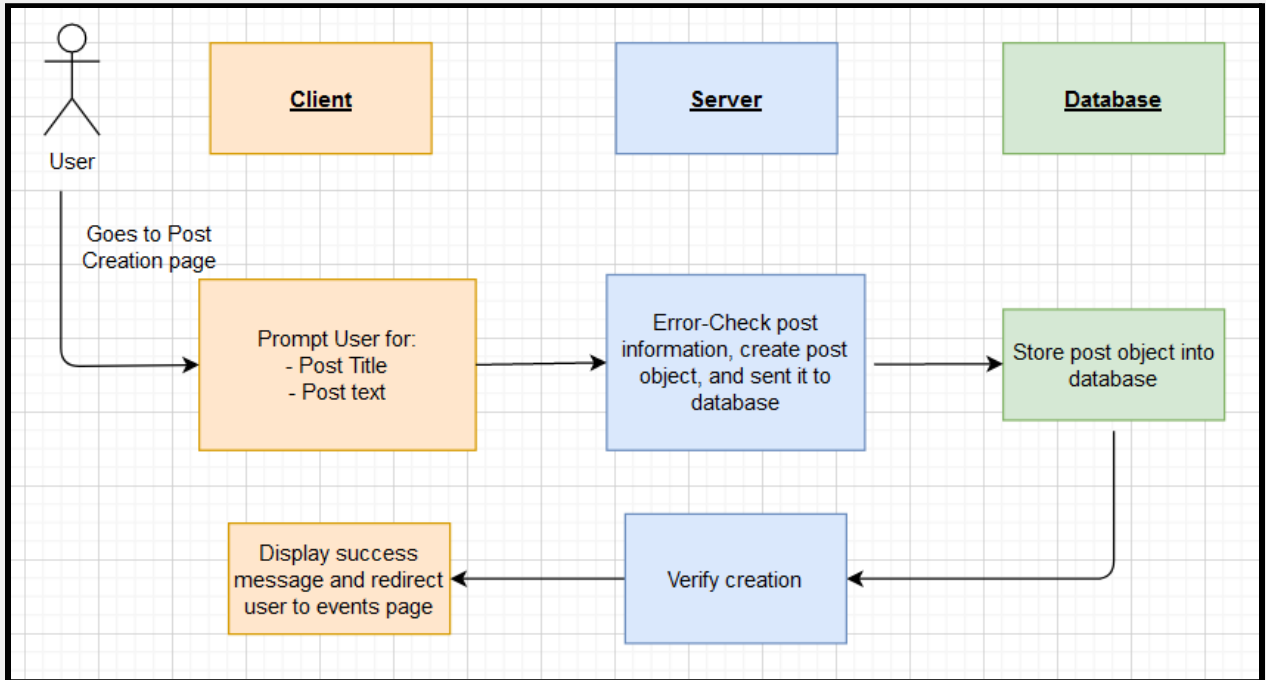
### 3. Create event

A user goes to the “Create Event” page on the events screen. They are then prompted to enter an event name, event location, event description, event time, and an associated club if applicable. The inputted information is then checked for validity and an event object is created. The object is then stored into the database. The server then verifies if the event object has been successfully entered into the database then adds the event object to the corresponding club object if necessary. Lastly, the user is redirected into an events page.



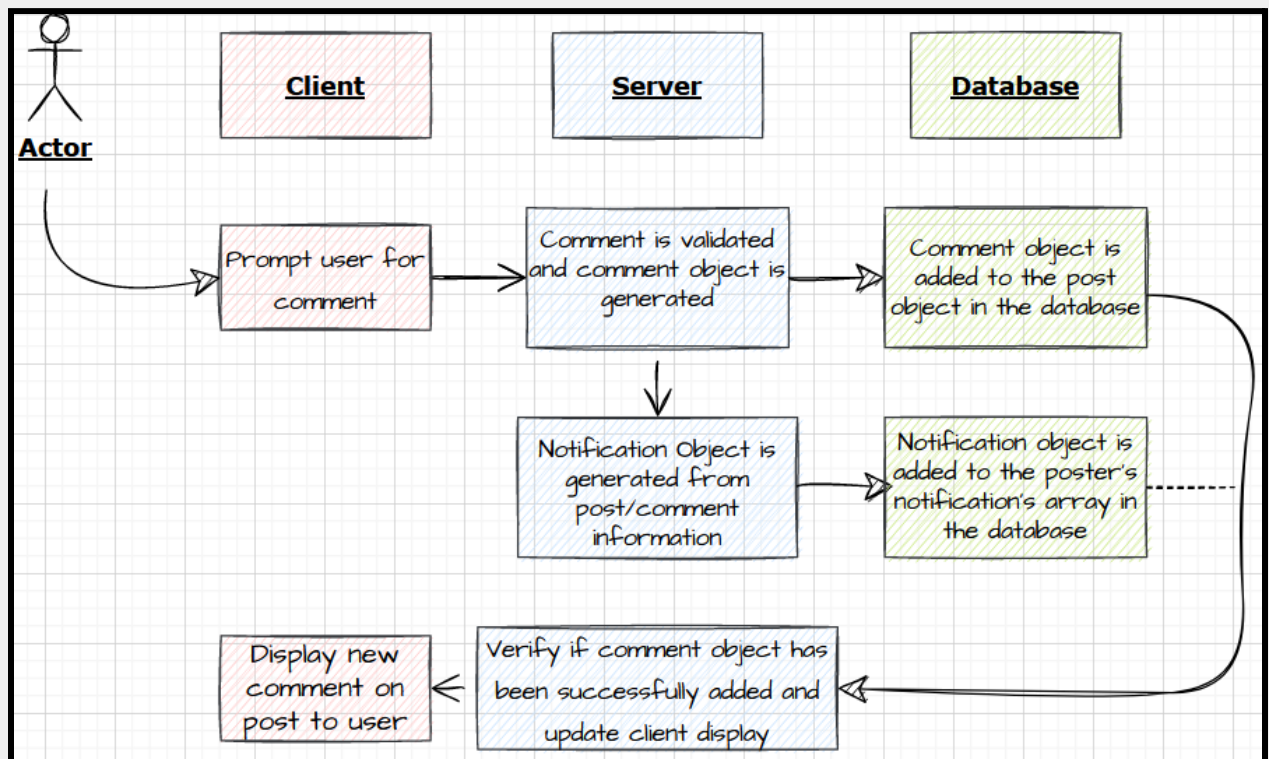
#### 4. Create Post

A user goes to create a post on a class forum page. They are then prompted to enter a post title and post text. The inputted information is then checked for validity and an event object is created. The object is then stored into the database. After the post creation is successful, the class forum page will be refreshed and the newly created post will be listed in the page.



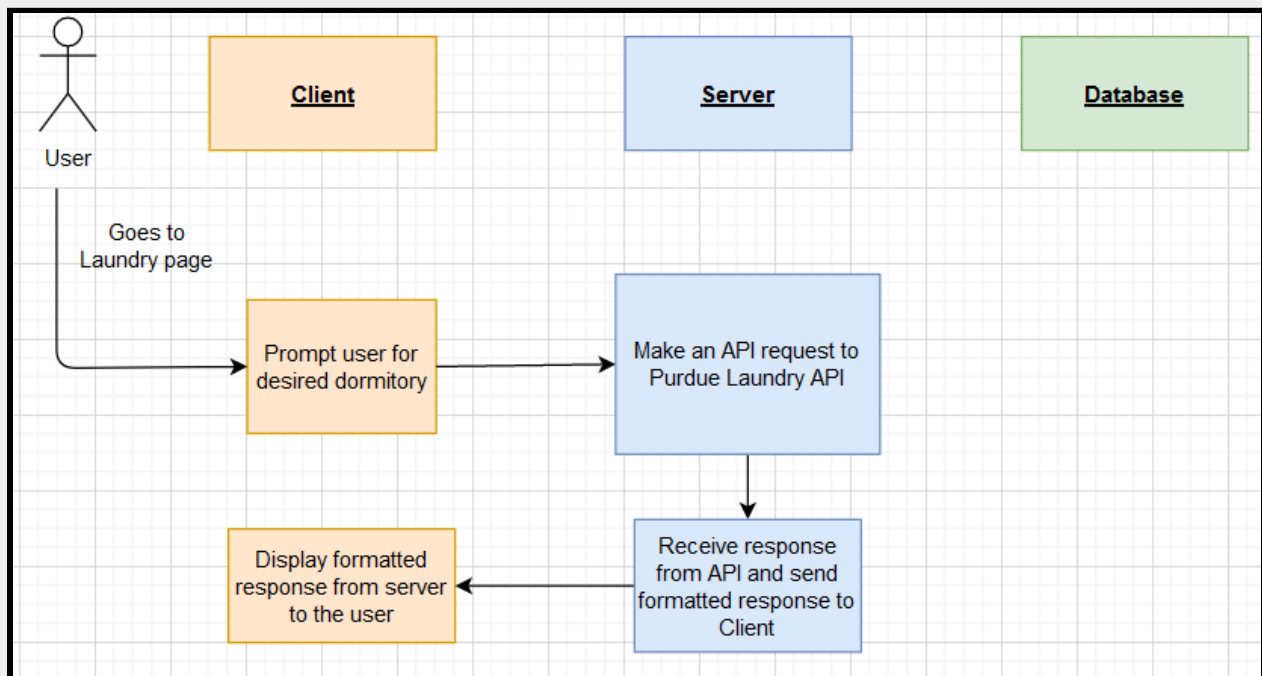
## 5. Comment on a Post

A user goes to a post on the class forum screen. The user replies to the post with a comment on the client. The comment is parsed and a comment object is created on the server once the comment is submitted. A notification object is also created. The comment object is then added to the post object's comment array in the database. The notification object is also added to the poster's notifications array in the database. The server verifies that the comment was successfully added to the post and the client then displays the new comment on the post to the user.



## 6. Display Laundry Status

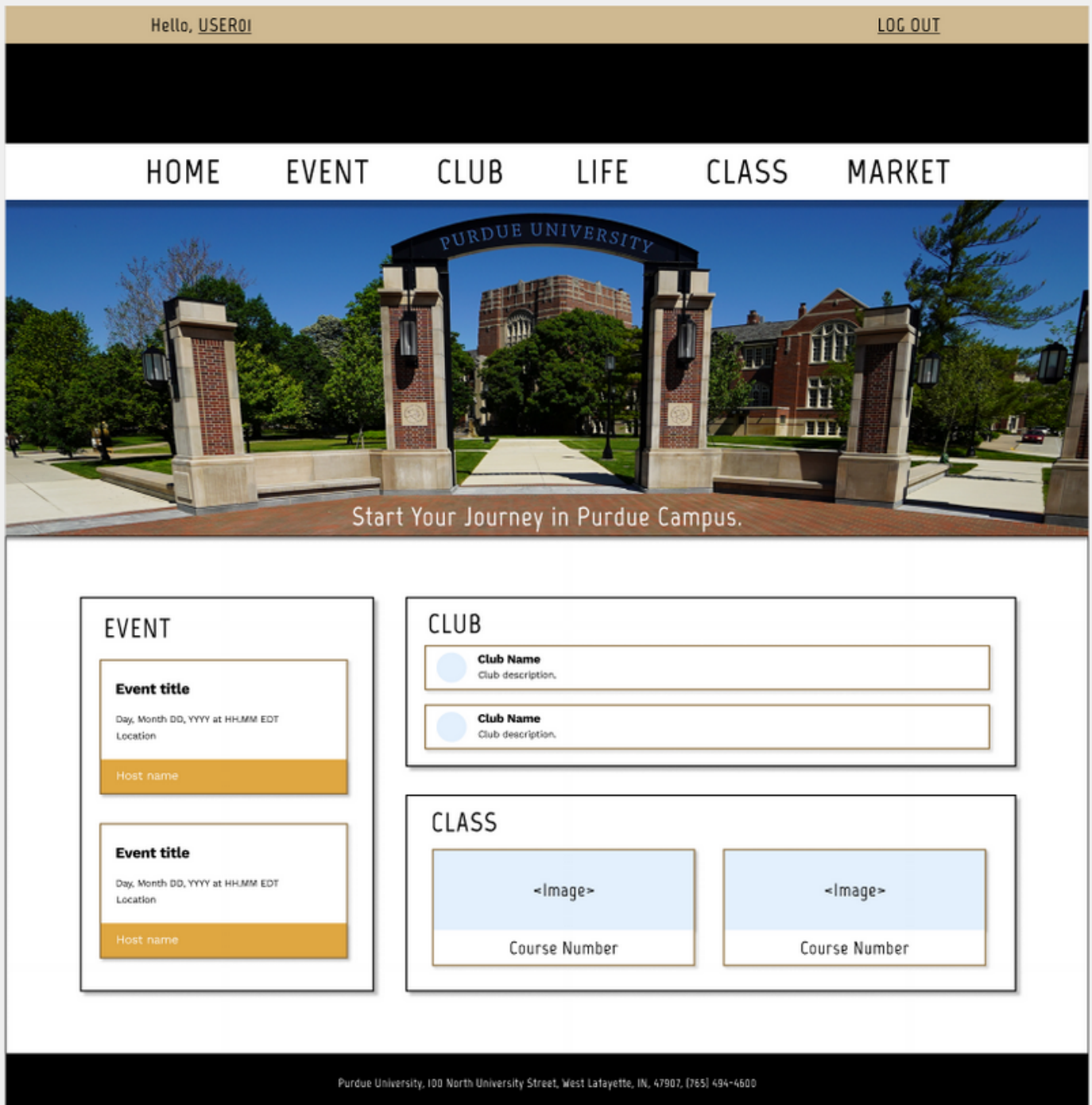
A user goes to the “Laundry” page on the Living screen. They are then prompted to select the dorm hall of interest from a list. The client then sends the user’s selection to the server where an API request is generated to a 3rd party Purdue Laundry API service. The server will then receive a response from the API and will send the response to the client. The client then displays the information on available machines to the user.



## d) UI Mockups of Core Features

### *Landing Page*

This page shows users a general overview of the features that PurdueParty.io offers and how to navigate to those features.



## Club Listings

This page shows users the list of clubs that are registered with PurdueParty.io. This page also allows users to create a new club.


Hello, [USER01](#)


[LOG OUT](#)


[HOME](#)[EVENT](#)[CLUB](#)[LIFE](#)[CLASS](#)[MARKET](#)


### Clubs


Create


**Club Name**  
Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis...


**Club Name**  
Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis...

**Club Name**  
Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis...

**Club Name**  
Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis...

**Club Name**  
Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis...

**Club Name**  
Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis...

**Club Name**  
Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis...

Load more

Purdue University, 100 North University Street, West Lafayette, IN, 47907, (765) 494-4600



## Club Info Page

This page shows users the information about a specific club and lists the public events that they will be hosting. This page also allows privileged users to edit the club information.

Hello, USER01

LOG OUT

HOMEEVENTCLUBLIFECCLASSMARKET

Club Name

EDIT

Club description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contact Information

Address line 1

Address line 2

City, State, Zip code

Country

john.doe.club@gmail.com

123-456-7890

Additional Information

Primary Advisor Name

John Doe

Primary Advisors campus email address:

john.doe@purdue.edu

Primary classification of the majority of the organizations current members.

Select the option which represents 50.3% of the group.

Undergraduate

Organization Meeting Day:

No Response

Organization meeting times:

Afternoon

Public Events

View More Events

< Image >

Event title

Day, Month DD, YYYY at HH:MM EDT

Location

Host name

< Image >

Event title

Day, Month DD, YYYY at HH:MM EDT

Location

Host name

< Image >

Event title

Day, Month DD, YYYY at HH:MM EDT

Location

Host name

< Image >

Event title

Day, Month DD, YYYY at HH:MM EDT

Location

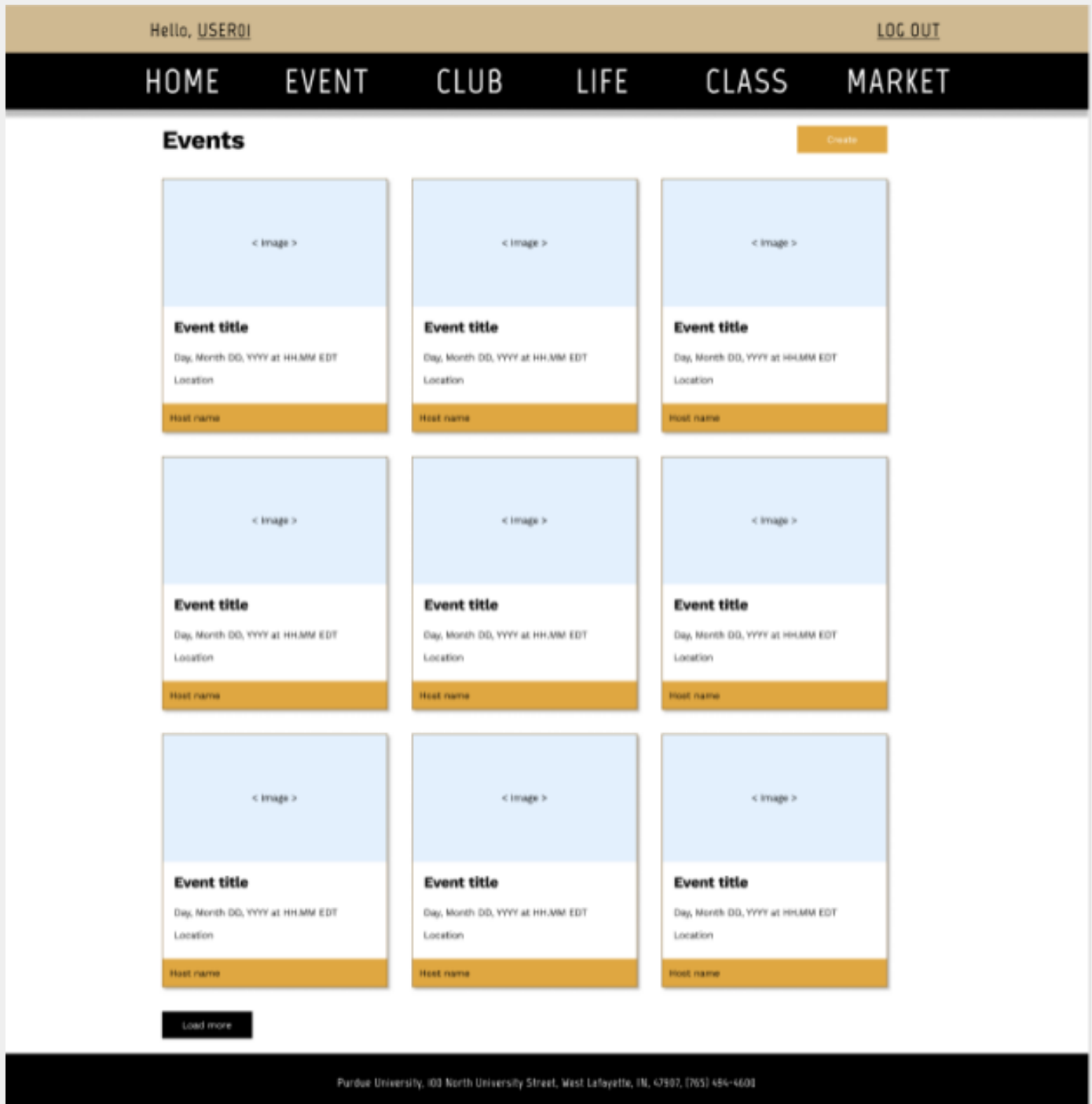
Host name

Purdue University, 100 North University Street, West Lafayette, IN, 47907, (765) 494-4500

25

## Events Page

This page shows users the lists of events that are happening at Purdue. It also allows users to create a new event.




## Event Info Page

This page shows the full information about a specific event. Privileged users can also edit the event information here.

Hello, USER01

LOG OUT

HOMEEVENTCLUBLIFECCLASSMARKET



< image >

Event Title

Edit

Date and Time

Day, Month DD, YYYY at HH:MM EDT to  
Day, Month DD, YYYY at HH:MM EDT

Location

Event location


Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Tags

Event TagEvent TagEvent Tag

Host

Club Name

Purdue University, 630 North University Street, West Lafayette, IN, 47907, (765) 494-4500

## Marketplace Listings

This page shows the lists of items that are currently up for sale in the marketplace. The user can click on any of the listings to see more information about the item or the seller.


Hello, [USER01](#)

[LOG OUT](#)

[HOME](#) [EVENT](#) [CLUB](#) [LIFE](#) [CLASS](#) [MARKET](#)


### Market Place

Search More

**Test**


Product description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ashsdshakosdeli ahupeldi iusthdi bla jorci oiax nmszpw lnsad kuh sd aow ziczxcv eev.

\$00.00

**Test**


Product description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ashsdshakosdeli ahupeldi iusthdi bla jorci oiax nmszpw lnsad kuh sd aow ziczxcv eev.

\$00.00

**Test**


Product description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ashsdshakosdeli ahupeldi iusthdi bla jorci oiax nmszpw lnsad kuh sd aow ziczxcv eev.

\$00.00

**Test**


Product description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ashsdshakosdeli ahupeldi iusthdi bla jorci oiax nmszpw lnsad kuh sd aow ziczxcv eev.

\$00.00

**Test**


Product description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ashsdshakosdeli ahupeldi iusthdi bla jorci oiax nmszpw lnsad kuh sd aow ziczxcv eev.

\$00.00

**Test**

Product description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ashsdshakosdeli ahupeldi iusthdi bla jorci oiax nmszpw lnsad kuh sd aow ziczxcv eev.

\$00.00

**Test**

Product description. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam ashsdshakosdeli ahupeldi iusthdi bla jorci oiax nmszpw lnsad kuh sd aow ziczxcv eev.

\$00.00

Load more

Purdue University, 600 North University Street, West Lafayette, IN, 47907, (765) 494-4680

## Class Forum Page

This page shows the list of posts / threads that have been created in a specific class forum. The user can expand any post to view the replies or to engage with it.

Hello, USER01

LOG OUT

HOMEEVENTCLUBLIFECCLASSMARKET

### CS 407: Software Engineering Senior Project

**Post Title**

Post Detail, Wikipedia has received praise for its enablement of the democratization of knowledge, extent of coverage, unique structure, culture, and reduced amount of commercial bias, but criticism for exhibiting systemic bias, particularly gender bias against women and alleged ideological bias.[1][12] its reliability was frequently criticized in the 2000s, but has improved over time and has been generally praised in the late 2010s and early 2020s.

Reply

**Post Title**

Post Detail, Wikipedia has received praise for its enablement of the democratization of knowledge, extent of coverage, unique structure, culture, and reduced amount of commercial bias, but criticism for exhibiting systemic bias, particularly gender bias against women and alleged ideological bias.[1][12] its reliability was frequently criticized in the 2000s, but has improved over time and has been generally praised in the late 2010s and early 2020s.

Reply

**Post Title**

Post Detail, Wikipedia has received praise for its enablement of the democratization of knowledge, extent of coverage, unique structure, culture, and reduced amount of commercial bias, but criticism for exhibiting systemic bias, particularly gender bias against women and alleged ideological bias.[1][12] its reliability was frequently criticized in the 2000s, but has improved over time and has been generally praised in the late 2010s and early 2020s.

Reply

**Post Title**

Post Detail, Wikipedia has received praise for its enablement of the democratization of knowledge, extent of coverage, unique structure, culture, and reduced amount of commercial bias, but criticism for exhibiting systemic bias, particularly gender bias against women and alleged ideological bias.[1][12] its reliability was frequently criticized in the 2000s, but has improved over time and has been generally praised in the late 2010s and early 2020s.

Reply

**New Post**

Title

Content

Submit

Load more

Purdue University, 600 North University Street, West Lafayette, IN, 47907, (765) 494-4600

29

## Living Section Homepage

This page shows users real-time information pertaining to gym, dining courts, residence halls, and city bus. Users can view specific information about each category in the appropriate subpage.

