TigerMeet Design Document

Table of Contents

0. Team

- Andrew Lin: arlin@princeton.edu
- Bora Kiyan (Team Leader): bkiyan@princeton.edu
- David Wu: dpwu@princeton.edu
- Raymond Park: rkp2@princeton.edu

I. Overview

TigerMeet is a responsive web application that will enable Princeton University students to connect with other students for recreation, transportation, and academic coordination. It is intended to be subdivided into sections of general interest (i.e. Sports, Transportation), with the ability to create specific events. Within sections and events, we plan on having chat capabilities for users to connect.

II. Requirements and Target Audiences

Problem

Students on campus do not have a unified platform for coordinating events with people they do not know. Current means of getting people of similar interests together are restricted in their scope and generally are very hard to compile information.

Intended Users

Any student on campus who may want to coordinate activities or transportation.

Existing Solutions

Previously, this has been mainly done through the use of the Princeton email system. There are multiple hassles and problems with this method:

- There is no all-school email listserv, meaning that students have to send out multiple requests every time they want to connect with students. Furthermore, some students unsubscribe from their residential college listservs.
- Most recreational events like ping pong or pool rely on leagues. Leagues require much organization, and are plagued by no-shows and forfeits. TigerMeet will provide a means through which students can connect without the overhead or elimination of leagues.
- For school work, most classes with collaborative problem sets do not provide a means through which to find partners. This makes it difficult for first year students or students who do not come in with initial friends in the class to find collaborators.

III. Functionality

Features

The application will have the following features:

- User authentication using Princeton's Central Authentication Service
- Navigate through the sections
    1. Sports
    2. Working Out
    3. Video Games
    4. Transportation
    5. Problem Set Groups
    6. Miscellaneous
- Land on one section and showcase its chat room
- A chat room with messages tied to users where any authenticated user can leave a message
- Event creating capability for users
- Having a stream of events listed on the top of the chat room enabling people to see them and join if wanted
- A private message chat linked to the event

Sample User Case

A sample user case would follow as such:
A user wishes to find someone to play ping pong on the upcoming weekend.

1. The user accesses the app website and authenticates their username via the Princeton University Central Authentication System (CAS)
2. The user clicks the Sports section on the menu occupying the left of the screen

3. The click to Sports reveals a chat room on the right of the screen where the user could look at the chat history related to the section of choice
4. The user can also look at the stream of events on the top created by other users and joins a ping pong one if it exists and works
5. The user then can chat privately with the owner(s) of the chat for more specific communication
6. If the user fails to find an event posted meeting the criteria such as the time or the location, they can create a new event using the event feature of the chat room

IV. Design

Front-End (Client Side)

Our front-end will be built with HTML (linked to Python with the Django Template Language), CSS (with the Semantic UI library), and JavaScript (with jQuery). Our app will feature a splash page from which users can log in using their Princeton netid and password, authenticated by CAS. After authentication, the main page will display, from which users will be able to enter different chat rooms by selecting a section from a collection of sections of general interest. Within the section chat rooms, users can send messages as well as create events. Events created will be displayed prominently at the top of each section and will have a limited lifetime, disappearing an hour after they end. Users can select an event to enter a chat room specific to that event.

Back-End (Server Side)

Our back-end will be built with the Django framework in Python. The app will be hosted on a Heroku server. It will have the following models:

- User, which contains the user's netid, user's email, created events, joined events, and created messages
- Events, contains an event chat room and joined users
- ChatRoom, contains different messages
- Section, contains a section chat room and events created
- Message, posted from within chat rooms

Users will have control over their events and messages, as these will each have create, read, update, and delete (CRUD) capabilities. Users will be able to join any events they do not own and access those message capabilities.

Database

A PostgreSQL database will be used to store data using the models specified in the back-end.

V. Timeline

Deliverables

March 10, 2019 - Design Document
March 24, 2019 - Basic Website Running
March 25, 2019 - TA Hours
April 12, 2019 - Project Prototype
April 26, 2019 - Alpha Test
May 2, 2019 - Last Class
May 3, 2019 - Beta Test

Milestones

Sunday, 3/17/2019 (After Week 6)

- Get a server up and running
- Create one section
- Deploy app on Heroku

Sunday, 3/24/2019 (After Spring Break)

- Add database
- Create an About page
- Create the chat room => Have functional app with one chat room
- Style section

Sunday, 3/31/2019 (After Week 7)

- Create event functionality => Have functional app with one chat room and event functionality
- Expand to multiple sections and chat rooms => Have functional app with multiple chat rooms and event functionality

Sunday, 4/7/2019 (After Week 8)

- Add Authentication
- Flesh out user interface
- Project Prototype on Friday, 4/12/2019

Sunday, 4/14/2019 (After Week 9)

- Testing and major debugging
- Find initial users

Sunday, 4/21/2019 (After Week 10)

- Share with entire school
- Alpha Test on Friday, 4/26/2019

Sunday, 4/28/2019 (After Week 11)

- Beta Test on Friday, 5/3/2019

Sunday, 5/5/2019 (After Week 12)

- Victory

VI. Risks and Outcomes

To reduce the risk of not having a fully functional product with all of our originally planned sections, we are going to start small and build up. That is, we will create a pathway into a section first. Then, for that specific section, we will implement a chat function such that people interested in that specific section will be able to chat. This will then be scaled to other sections. This allows us to ensure that specific parts of our project are functional. We will make sure that there is a viable pathway towards a section. Then the chat. Finally the event creating feature will be added and enhanced with message capability. Then perhaps other aspects such as design, login page, etc. will be implemented. By building up, we reduce the risk of having a product that spans many different sections but is also riddled with bugs. It is better that we cater towards a specific group of people on campus and make sure that our app works than to cater to many different people but have the product essentially broken.