Team Proposal

Team WCJ

Christopher Coyne ctc2141
Phu D Pham pdp2121
William Chiu wnc2105
Julie Song js5360

Part 0:

We met with Asif on 11/5/2020

Part 1:

1) Our project will allow users to create online lion avatars and navigate between different campus-themed chat rooms. Their chat messages will float on top of their avatars to give the appearance of talking.
2) Our users will consist of freshmen and other incoming students who have never seen Columbia's campus first hand due to COVID-19. Other students who miss campus may also be interested. All users will be required to have a columbia.edu or barnard.edu email. Users will need to log in every time they join the site and will automatically be logged out when they leave, or are inactive for some duration of time.
3) We will be able to show a website where a lion avatar (or multiple) could walk to the coordinates of a cursor click. Upon typing a message, a chat bubble will appear above the lion. The above two actions committed by other users will also be visible to all users connected to the same chatroom.
4) We will store which email addresses are in which chat rooms and at what coordinates are they located.
5) http://www.purgomalum.com/ will provide us with quality profanity checking so we can prevent users from cursing and using other inappropriate language.
6) In order to make sure our users have a columbia.edu or barnard.edu account, we will either:
    a) Use Google authentication to retrieve email address and check if it ends in columbia.edu/barnard.edu
7) In order to authenticate users, we will have them login through their email account (from a columbia.edu or barnard.edu domain)

Part 2:

1) FRESHMEN: As a freshman, I want to view the Columbia campus and spatially understand its layout so that I will be less disoriented when I eventually come to campus for the first time.

   My conditions of satisfaction are:
   Common cases:
   a) The chat rooms must be linked to different locations on Columbia campus, and link to one another in a way consistent with campus's layout.
   b) The chat rooms will describe why the locations/chatrooms chosen are locations of interest.
   c) Locations that freshmen typically spend time at will be marked as such.

   One special case is:
   a) If the user tries to access a location, or a chatroom associated with a location which we do not support, we will reject any such request.

2) CLUB MEMBERS: As a member of an in person club on campus, I want a virtual meeting place where I can meet up with other members so that we can continue to get to know each other and have a fun way to socialize, even though we are not meeting on campus.

   My conditions of satisfaction are:
   Common cases:
   a) There must be a basic chat feature that we can use to communicate with other people in the club chat room.
   b) There must be private chat rooms where a club can meet up without interference from non club members.
   c) There must also be a way to send direct messages to individual club members in these chat rooms, as opposed to sending messages viewed by all.
   d) Club founders should have some kind of indicator next to their avatar so that they are readily identifiable.

   One special case is:
   a) If a user tries to enter a club chat room, via an API request, but is not a member of the chat room, and was not invited to the chat room, they will not be allowed to enter the chat room.

3) INCOMING STUDENT: As an incoming student (freshmen/transfer student), I want to be able to communicate with other new students so that I can build an auxiliary social network before physically meeting them.

   My conditions of satisfaction are:
   Common cases:
   a) There must be a communication functionality in the form of chat that allows me to send messages to other people on the platform

b) There should be a way to add ‘friends’ to your network so that you may keep track of the people you have conversed with

c) There should be an option to continue conversations with ‘friends’ in private chatrooms as opposed to the public interface.

One special case is:

a) If a user tries to private message another user which they are not friends with, their message will result in an error, and will be rejected (the message won’t be sent).

Part 3:

1) FRESHMEN: As a freshman, I want to view the Columbia campus and spatially understand its layout so that I will be less disoriented when I eventually come to campus for the first time.

   Common cases:

   a) A demo testing user will go to the website and login.
   b) They will confirm that they can see Butler library and can immediately identify their avatar.
   c) Taking into consideration the perspective of the image, Lerner, John Jay, and College Walk will be reachable by right clicking on the edge of the background graphic and waiting for the avatar to walk towards the clicked point.
   d) Each location will come with a popup on the top left with a “DId you know?” fact about the location.
   e) Each “Did you know?” fact will describe relevant usage of the location depicted during years where there isn’t a global pandemic.
   f) Chatrooms corresponding to freshmen dorms will additionally have a popup on the top left saying “Hot with freshmen”.

   Special cases:

   g) If a user tries to send a POST request from scratch that sends their avatar to a location/chat room which is not in our set of locations/chat rooms, they will remain in the same place and the request will be rejected.

2) CLUB MEMBERS: As a member of an in person club on campus, I want a virtual meeting place where I can meet up with other members so that we can continue to get to know each other and have a fun way to socialize, even though we are not meeting on campus.

   Common cases:

   a) A demo testing user will go to the website and login.
   b) They will confirm they are logged in by seeing their avatar and surroundings.

c) The user will navigate their avatar by clicking on the virtual map of columbia.

d) They will have the option of joining a club chat room by entering the name of the chat room (and if it is a private one, a password). After hitting "join" the user's avatar should now be inside of the club chat room with other club members.

e) Whatever the user now types into his chat bar should only show up in this club chat room, and should not be visible to anyone outside.

f) By clicking the button "return to public chat" the user should be transported back to the main public chat room, the map of columbia where they first logged into.

Special Cases:

g) If a user tries to send a POST request from scratch that adds their avatar to a club chat room/location they are not a part of and were not invited to, the request will be rejected.

3) INCOMING STUDENT: As an incoming student (freshmen/transfer student), I want to be able to communicate with other new students so that I can build an auxiliary social network before physically meeting them.

Common cases:

a) A demo testing user will confirm their login credentials and be able to access the platform.

b) All online users in the vicinity should be visible on the interface as well as in an interactable state.

c) Sending a chat from one user in the public chatroom should send a message in real time that is visible to everyone on the site.

d) Sending a friend request to a specific user should alert the other user with a popup notification as well as an option to accept or decline.

e) Accepting the request should automatically open a private chatroom that allows both users to send messages that are visible only to each other.

f) Taking in the history of all friend interactions, a menu of all friends made on the site should be readily available to the user.

Special Cases:

g) If a user tries to send a POST request from scratch that sends a message via a private chat with another user they are not friends with, the request will be rejected.

Part 4:

For Python, we will not be using a compiler. For Javascript, we will be using babel as well as webpack. We will be using VSCode or Sublime Text or Vim as our IDE. For Python, we will store our requirements in a requirements.txt file. For javascript, we will use create-react-app and set up the frontend as a node project. Building, and installation of javascript dependencies will be handled by node (npm). For JS testing, we will use Jest in conjunction with Travis and ESlint for style checking (linting). For coverage, Jest also provides coverage reports using the --coverage tag, but we may also use it in conjunction with Codecov.io for formal coverage reports. For Python style checking, we will use pep8. For Python unit testing, we will use unittest. For Python coverage tracking, we will use coverage. For Python bug finding, we will use ipdb. For persistent data storage, we will use MongoDB.