

Conceptual Sketch

11/10/2021

globo-gym

Key Purposes and Social Needs:

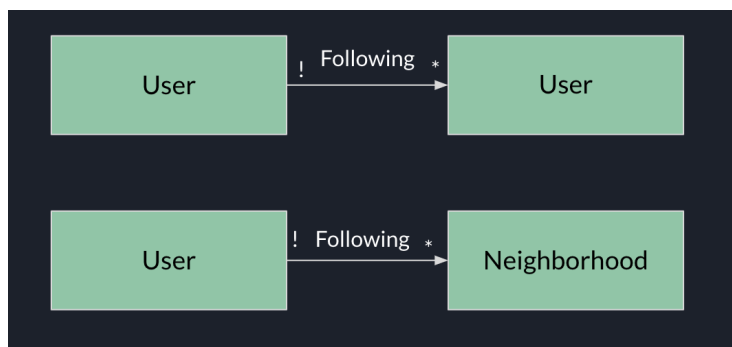
Throughout all of Covid, restaurants have been switching to a more outdoor and open style with many tables now in booths in the street where bike lanes normally are. On top of that, construction in Cambridge is booming in these post covid times. Both of these aspects have served to create more and more traffic within the bike lanes of Cambridge, while also making many bike lanes unusable as a whole. Our main purpose for making Globo Bikes is to help our Cambridge community find safer and faster biking routes to their destination. Ultimately, we aim to make the process of riding a bike as smooth and easy as possible, not only because it is the healthier option, but also because it is a much healthier form of transportation for our environment. This allows users of Globo Bikes to know that they are not only helping their community by actively supporting the safety of fellow bikers, but they are also helping themselves and the environment by doing so.

Main Concepts:

concept: Follow

Purpose: Link a user to other users and neighborhoods to let them view associated reports and biking activity

State:



Actions:

follow(u1: User, u2: User)

unfollow(u1: User, u2: User)

follow(u: User, n:Neighborhood)

unfollow(u: User, n:Neighborhood)

Operational Principle:

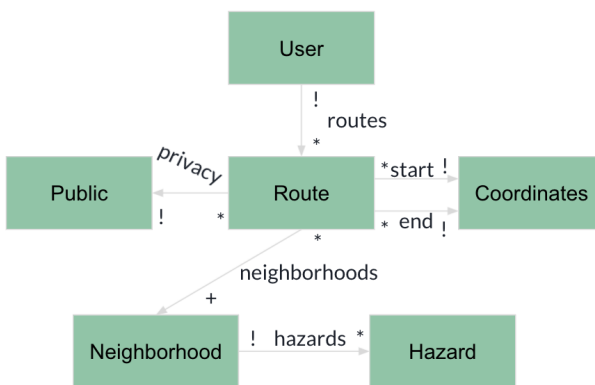
after follow(u1, u2) and not unfollow(u1,u2) =>
u2 in u1.following

after follow(user, neighborhood) and not unfollow(user, neighborhood) =>
user in neighborhood.following

concept: Route

purpose: to guide users safely from one destination to another, and ride info with followers

state:

**actions:**

route(user: User, start: Coordinate, end: Coordinate):

```

fresh route;
user -> route;
route.start = start;
route.end = end;
route -> one/multiple neighborhood:Neighborhood given start / end coordinates
  
```

publicize(route: Route, toPublicize: Boolean):

```

route -> toPublicize;
  
```

saveRoute(route: Route, user:User):

```

user -> route;
  
```

operational principle:

if route(user, start, end), then find the best route from start to end such that it avoids hazards in the given neighborhood of those coordinates, then ask for answer to publicize(route, toPublicize) upon completion

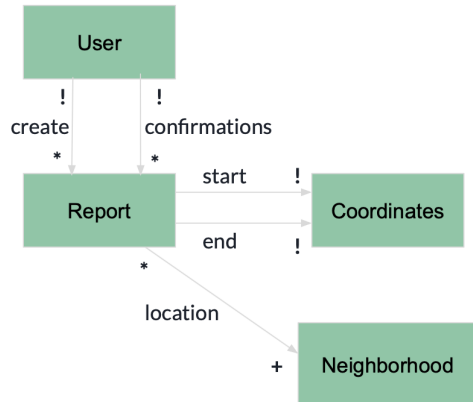
if publicize(route, true) then show users following the route's creator this route, otherwise keep it private

if saveRoute(route, user), add that route to the user's system such that they can use it in the future

Report:

purpose: identify where there are any blockages or obstructions in streets and bike paths

state:



actions:

createReport(user: User, start: Coordinate, end: Coordinate):

```
add report to Reports;
user.create = report;
report.start = start;
report.end = end;
report.location = neighborhood which Coordinates (start, end) is in the world
return report
```

deleteReport(user: User, report: Report):

```
if report in Reports and create.report = user:
    remove report from Reports and user.create
```

confirmReport(user: User, report: Report):

```
if report in Reports and create.report != user:
    user.confirmations = report
```

numberOfConfirmations(report: Report):

```
return sum of list confirmations.report
```

operational principle:

u1: User1, u2: User2, s: Start coordinate, e: End coordinate, r: Report

- 1) createReport(u1, s, e) => r; confirmReport(u2, r); numberOfConfirmations(r) = 1;
- 2) createReport(u1, s, e) => r; deleteReport(u1, r) => r does not exist;

Central Questions to Answer:

1) In what ways is the app more than just [CRUD](#)?

The app is more than just CRUD, because in addition to users creating reports and reading them, seeing routes of other users, and editing their profile information, there is a social and

geographical aspect to the system. Users can follow other users and neighborhoods, and create, read and update within these sub-networks. Additionally, we have many interactive features as well that we will allow users to interact with already existing data - users can save other routes to claim them as their own, they can interact with their following's routes to show their support, and they can confirm other user's reports to generate a community around the safety of bike riding. As such, there are a lot more interactions and features other than creating, updating, or deleting individual pieces of information from the platform.

2) How does it involve at least one concept that is not already widely used?

The report concept does not seem to be widely used, particularly in the context of mapping/routing. One of our goals for this application is to draw off of user feedback to give the best routes that are 1) safe, 2) quick, and 3) obstruction free. Thinking of cars, whenever apps give the quickest routes, they only care about the obstruction of traffic. Obstructions, lack of bike paths, and any other blockages aren't really tracked by any public data to our knowledge, so using reports will be our strongest indication of these relating to biking around cities and neighborhoods. Users can ensure that other bikers will have safe trips and update the current blockage with confirmations--another unique aspect of the report concept. Users have the ability to confirm other user's reports during a pause in their trips or after their trips if they see the obstruction. We hope this creates a community aspect and makes biking safer.

3) What particular design areas are likely to be challenging and why.

Creating a routing app like Waze is in itself not an easy task, however doing so by incorporating user-reported hazards, adjusting biking routes with that information, and allowing routes and other features to be a part of social media adds a lot of layers of difficulty. Of all of these features however, we feel as though that the most difficult part will be the routing itself - this is because we will have to outsource to something like Google Maps API, and adjust the results to not include the given hazard lanes. Another area that we think will be the most challenging is that we need to update routing and feeds in real time, rather than only upon an action by the user - this is extremely important, as not having live updates of hazards could result in some users not getting accurate information of hazards if they were recently reported during their creation of a route. To do this, we are going to need to use a more robust connection with the backend code. Aside from these, most of our other design areas really came from previously made things in the class (particularly with the social media aspects), such as authentication, following, etc.