You will be penalized for including irrelevant points, for repeating yourself, for saying things in text that are already said in diagrams, and for lack of structure

Overview

This is an outline of what you plan to build. It should include the following:

- Brief description of system to be built
- Key purposes (What problems does it solve? Why should it exist?)
  The purposes could be clearly delineated, each with a name and summary in a short sentence, and then explained more fully.
- Deficiencies of existing solutions (if they exist, and if not, why you suspect not)

Written by: Jianna Liu

We plan to build a mapping web app for general bikers in the Boston area to be able to

1. **Find and filter for all safe biking paths from a start point to a destination**. Users will have the ability to mark two locations as a start and end point. From there, our system will generate all available biking paths in between these 2 locations, and a user can filter based on the tag associated with each path.
2. **Inform other users of concerns on a path through tags.** Users will have the ability to post about the status of a path or intersection, delineating it with a photo, comment, and tag (e.g. Busy, Blocked, etc.) about the current situation.

Existing solutions include Open Streetview and Google Maps, which are able to accurately pinpoint bike paths, but fail to inform users about the most “safe” bike path. We suspect they do not provide such information because “safe” is determined subjectively. As a result, we seek to build this social mapping web app where users can publicly interact and determine this information as a collective.

Conceptual Design

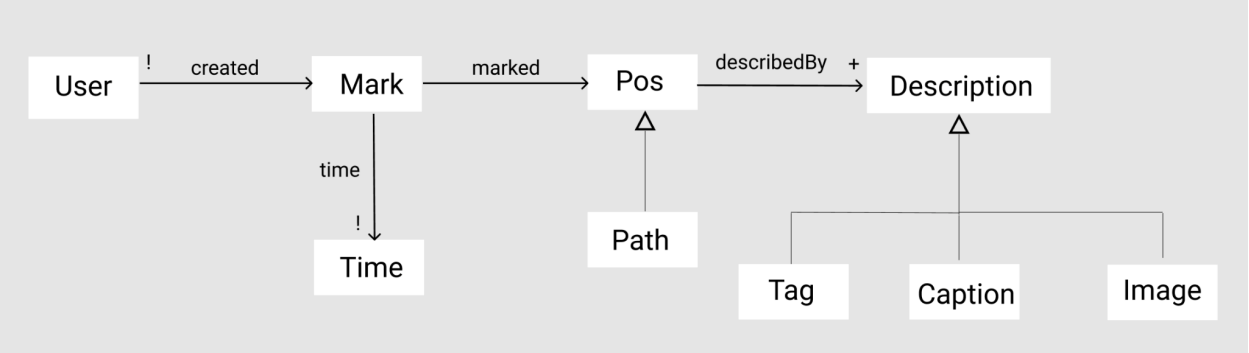
Written by: Mufaro

Marking Concept

Purpose

To add descriptions about conditions of sections or intersections on roads that contain bike lanes.

State



Actions

create(u: User, pos: {st: Pos, end: Pos}, img: Image, tag: Tag, cap: Caption, time: Time, **out**: Mark)

mark := fresh Mark
mark.pos = pos
mark.img = img
mark.tag = tag
mark.cap = cap
mark.user := u
mark.time = time

modify(u: User, m: Mark, new: {pos: {st: Pos, end: Pos}, img: Image, cap: Caption, tag: Tag, time: Time})

m.{pos, img, tag, cap, time} = new

delete(u: User, mark: Mark)

u.creates -= u -> mark

Operational principle

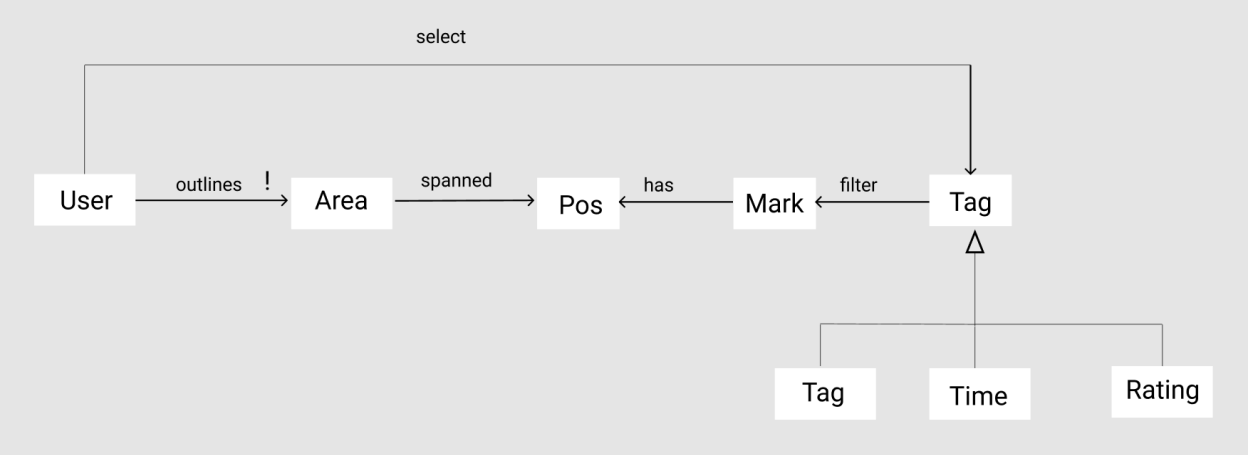
after create(u, pos, img, tag, cap, time, mark), modify(u’, mark, new_properties) => u = u’
after delete(u, mark), no modify(u, mark, new_properties)

Planning Concept: [Mark, Rating]

Purpose

To know paths and intersections in an area that have some inconveniences to find the most optimal bike path.

State



Actions

getMarksInSpannedArea(u: User, pos1: Pos, pos2: Pos, **out** mark: []Mark):
    polygon := circle with pos1 and pos2 on the diameter : **set** Pos
    result := **set** all mark: marks | mark.marked in polygon

filterByTag(u: User, marks: []Mark, tag: Tag, **out** result: []Mark):
    u.selects += {tag}
    result := **set** all mark: marks | mark.marked.describedBy = tag

filterByTime(u: User, marks: []Mark, time: {start: Time, end: Time}, **out** result: []Mark)
    u.selects += {tag}
    result := **set** all mark: marks | time.start <= mark.time < time.end

filterByRating(u: User, marks: []Mark, rating: Rating, **out** result: []Mark)
    u.selects += {tag}
    result := **set** all mark: marks | mark.rating >= rating

removeFilter(u: User, filter: Filter, marks: []Mark, **out** result: []Mark):
    u.selects -= {filter}
    result := **set** all mark: marks | mark.~filter in u.selects

clearFilters(u: User, marks: []Mark, **out** result: []Mark):
    u.selects := []
    result:= marks

Operational Principle

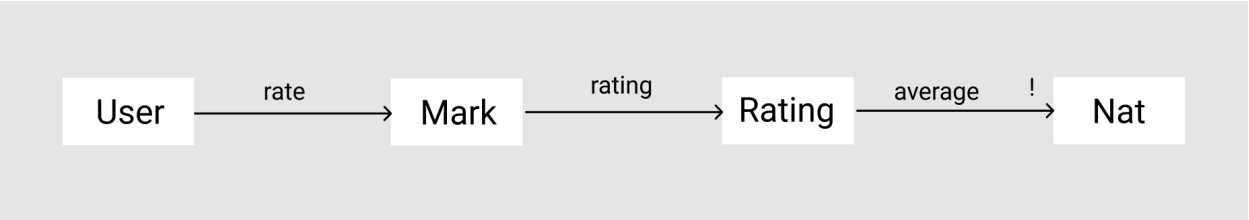
after getMarksInSpannedArea(u1, p1, p1, **out** marks1);
    after filterByTag(u, marks1, tag, marks') then removeFilter(u, tag, marks1, marks'')
        => marks'' = marks1
    after filterByTime(u, marks1, time, marks') then removeFilter(u, time, marks1, marks'')
        => marks'' = marks1
    after filterByRating(u, marks1, rating, marks') then removeFilter(u, rating, marks1, marks'')
        => marks'' = marks1
    after filterByTag(u, marks1, tag, marks') then filterByRating(u, marks', rating, marks'')
        removeFilter(u, rating, marks'', marks''') => marks' = marks'''
    after filterByTag(u, marks1, tag, marks') then filterByRating(u, marks', rating, marks'')
        clearFilters(u, marks1, marks''') => marks1 = marks'''

Rating Concept

Purpose

Metric for measuring a user's credibility as an information sharer on a scale of 1 - 5, 1 being least reliable and 5 being highly reliable.

State



Actions

rate(u: User, mark: Mark, rating: Rating)
    ratingObj = fresh Rating
    ratingObj.user := u
    ratingObj.rating := rating
    u.rate += {mark}
    mark.rating += {ratingObj}
    mark.average = average of all the ratings from all users

edit(u: User, mark: Mark, rating: Rating)
    ratingObj := set all obj: mark.rating | obj.user = u
    ratingObj.rating := rating
    mark.average = average of all the ratings from all users

delete(u: User, mark: Mark)
    u.rate -= {mark}
    mark.rating := set all obj: mark.rating | obj.user != u

Operational Principle

after rate(u, mark, rating), then no rate(u, mark, rating')
after rate(u, mark, rating) then can edit(u, mark, rating')

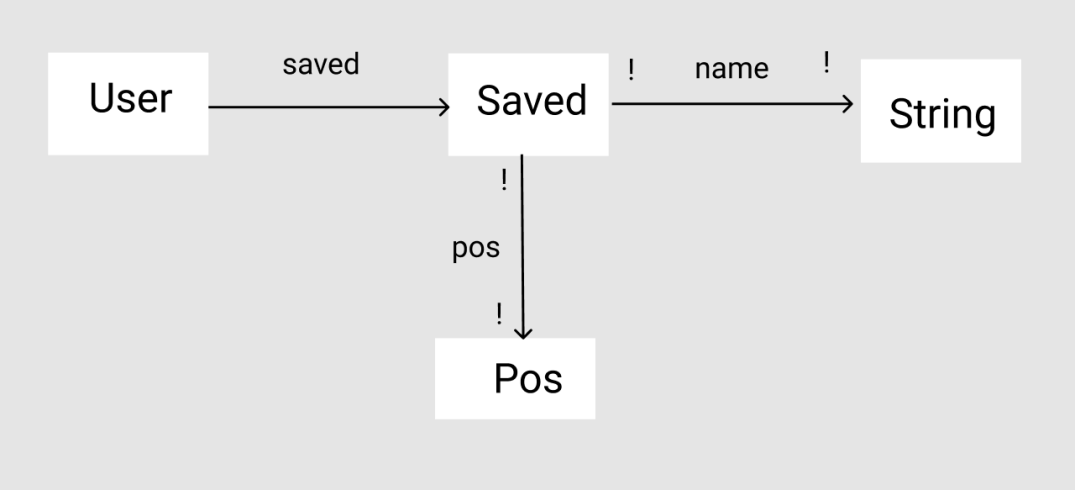
after rate(u, mark, rating) then can delete(u, mark)
after rate(u, mark, rating) then delete(u, mark) then no edit(u, mark, rating')

Saving Concept

Purpose

Allow users to attach names to an area surrounding markings that can be used to retrieve that area

State



Actions

save(u: User, pos: {start: Pos, end: Pos}, name: String, **out** result: Saved)
    result := fresh Saved
    result.pos := pos
    result.name := name

modify(u: User, name: String, new: {name: String, pos: {s: Pos, end: Pos}}, **out** result: Saved}
    savedObj := **set** all saved: u.saved | saved.name = name
    savedObj.{ name, pos } = new
    result := savedObj

delete(u: User, name: String, **out** result: []Saved)

```
        u.saved := set all saved: u.saved | not saved.name = name
        result := u.saved

    get(u: User, name: String, out result: Saved)
        result := set all saved: u.saved | saved.name = name
```

Operational Principle

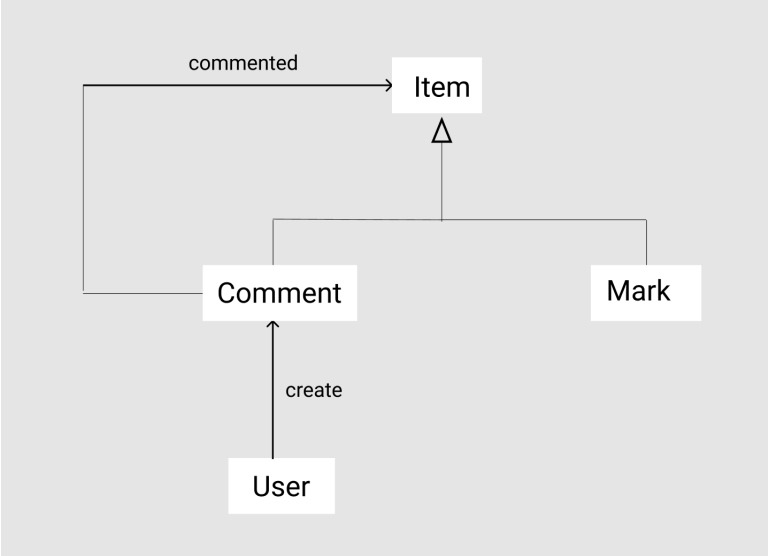
After save(u, pos, name, saved):
- no save(u, pos', name, saved')
- get(u, name, saved') => saved' = saved
- modify(u, name, new_name, saved') => no get(u, name, saved')
- delete(u, name, saved) => no get(u, name, saved')

Commenting Concept: [Mark]

Purpose

To allow users to follow up on a marking or comments to the marking made by other users as a means of following up on the validity of the marking

State



Actions

addComment(u: User, content: String, time: Time, item: Item, **out** result: Comment)
result := fresh Comment
result.content := content
result.user := u
result.time := time
item.~commented += {result}
u.creates -= {result}

editComent(u: User, com: Comment, content: String, time: Time, item: Item, **out** result: Comment)
item.~commented -= {com}
u.creates -= {com}
com.content := content

```
        com.time := time
        item.~commented += {com}
        u.creates += {com}
        result := com

    deleteComment(u: User, comment: Comment, item: Item)
        item.~commented -= {comment}
        u.creates -= {comment}
```

Operational Principle

```
    after addComment(u, content, time, mark, comment):
        deleteComment(u, comment, mark) => no edit(u, comment, content, time, mark)
        then can edit(u, comment, content, time, mark)
        then addComment(u', content, time, comment, comment')
            => no addComment(u'', content, time, comment, comment''')
```

Sketches

Written by: Hillary

https://www.figma.com/file/LPwuzcgFjiRKDXf0semKv5/Class-Project?node-id=0%3A1

Design Commentary

Written by: Jianna Liu

The main design decisions we collected through our planning is accumulated below:

1. **Prioritization of Displayal of Marks**

Many key design questions came up when deciding how to display our planning concept was the order in which we would display the marks. Our options were:

1. Displaying based on the time of the mark. We want to prioritize the relevance of the marks. By placing more recent marks at the top, it is more likely that the issue discussed in the mark is still happening and will improve the accuracy of the information a user receives.
2. Displaying based on the credibility of the user who posted. One aspect we want to minimize is the impact spam content has on our platform. Our concept, rating, discusses this. By prioritizing users with more credibility, we can trust that the information is accurate. However, this alternative will prevent newer users from developing a rating and using this platform effectively.
3. Displaying marks based on whether it is a favorite or not. By prioritizing marks that have been favorited, we would allow users to easily see the details of a frequent path. However, the marks which have been favorited may not contain relevant information anymore.

4. Displaying marks based on its associated tag. By prioritizing marks which are marked as safe, we can allow users to easily determine the most “optimal”/safe path. *** write however here

When analyzing these choices, we decided to display based on the time of the mark. However, a user also has the option to filter the provided marks based on credibility of the user (2), favorited marks (3), and its associated tag (4).

2. Creation of Mark

Key design questions that came up when discussing the user flow of the creation of a mark include:

1. Unique vs. related post. When a user is in the process of creating a mark, we don’t want a duplicate mark, in which the same content (e.g. the intersection on Mass Ave and Memorial is busy) is posted multiple times by different users. Our final decision was to allow users to see all related posts before continuing to add information to their mark in order to reduce marking traffic. We determined this would not be effective because a user may not check beforehand whether or not a post has had similar content.
2. Prevention of spam content in a mark. Our final decision in attempting to prevent spam content (and through the credibility of a user) was to have every mark incorporate a rating. It starts from 5 stars and users are given the ability to rate the upvote once. This rating allows users to determine whether or not to trust this rating based on the collective opinion. One alternative we considered was constraining the amount of marks a user can make depending on their credibility (a person with lower credibility can only post e.g. 3 times, incentivizing them to post more quality content). We determined this would not be as effective because the collective may poorly spam the rating of a mark as well, hurting the user’s credibility in the process. By further constraining it, this would make this process more susceptible to spam.

Ethics Protocol Analysis

You should create an ethics protocol analysis for your project, with discussion of three value-laden design decisions. An explanation of analysis with the ethics protocol and examples may be found in the slides for the lecture here (TBA). You should address every point outlined in the examples from the lecture in a concise and thoughtful way, and you will be asked to update the protocol for the submission of MVP and Finished Product with changes you've made to the protocol after having iterated on your design.

Written by: Hophin

Identify Stakeholders (Anyone or anything that can be affected by your project)

- Our Users
- The government/local municipality
- Restaurants/Property owners
- Third party software (Open Street Maps, Leaflet, Google, Bluebikes)
- Cyclists

Preliminary Design Choices

For our preliminary design choices, some of the questions we discussed are below:

- Who can post markings? (our users ?)
- Who can see markings ? (anyone ?)
- How are users rated, that’s a design choices

- How are markings made?
- How do we sort posts?
- Who has access to what?
- How is the user authenticated / verified?
- What determines the validity/relevancy of a mark?
- How is the user a credible source of information?

We also want our users to be able to have selective access to features as well as a seamless login through third-party authentication.

Envision Possible Futures (Guiding Questions)

When envisioning possible futures, we aim to think about the best case, worst case, and potential complaints.

In regards to the best case scenario, we would like our app to be adopted by the local municipality. This would increase the number of users on our platform and more potential data and funding. The app will hopefully facilitate awareness of various traffic related issues in the community which will eventually speed up government response in fixing these issues. Examples of this increased government response would include the construction of wider bike lanes, new bike lanes where restaurants have impeded on old ones, and the blocking of bike lanes near dangerous construction.

In regards to the worst case scenario, our app may be taken over by users who misuse the app through spamming of invalid posts. Currently, our app gives users the power to add marks without validating. Through spamming, our app will not be trustworthy and will lose reliability. Our app also heavily relies on the crowd. If no one is on our platform or no one is engaged, resulting in no posts, then our app will not have any functionality. If our app crashes, this is another worst case scenario as our users will not be able to access any information in this time period.

Some anticipated complaints we have from our users is that the ability to choose a location on a map may not be accurate enough.

Identify values at play

Fairness

- Every user interacts with the app equally. User's rating/credibility in the app solely depends on the marks that the user creates and therefore no user is treated unequally in that sense. Moreover, only users who are authenticated can interact with the platform

Transparency:

- We are committed to be transparent about the way in which we use our user data. We are not using user data in any malicious way such as selling it to advertisers.

Safety

- We are also committed to ensuring that our app is safe for all users. Safety is also defined in regulating misinformation on the state or conditions of road networks/
-

First Decision **Selective access to features**

1. Our Users

Process lens: Authenticated Users and non-registered users will understand the reasons behind the selective access to features. Authenticated users already have a valid identity therefore can be easily held accountable on our platform. That's why only authenticated users are able to "write" to our platform, i.e. make marks, comment on marks, rate marks. However, non-registered users can benefit from the marks that authenticated users have published. This also upholds our value of **fairness** that certain responsibility/privileges can only be given to people who are capable of being held accountable when they fail to uphold them.

Outcome lens: The best case scenario for users with the decision is that unregistered platform users can reap the fruits of authenticated user's labor in marking areas. The best case for authenticated users is exclusivity of the platform i.e. there is **safety** in that posts by users on the platform are more reliable. The worst case scenario for non-authenticated users is the lack of authenticated users who can mark roads, intersections on the platform.

2. The government/local municipality

Process lens: The government is explicitly left out in the design decision. There's no portal/window that they can interact with the platform from an administration point of view. So they can interact with the platform as authenticated users with ordinary profiles or salvage the features such as planning meant for non-authenticated users. However, this may compromise our fairness values as they could have a greater impact in the distribution of the application, and can be more informative on the conditions of road networks.

Outcome lens: The best outcome for the government is that they can rely on validity of marks that are done by authenticated users.

3. Bluebikes

Process lens: BlueBikes are included with incorporated features such as the searching of Bluebikes stations. This is fair, as they get to become an integral part of the platform by providing much needed transit services to our users.

Outcome lens: Best case scenario is that more Cambridge citizens are aware of Bluebikes as the feature to search for bluebikes is available to both authenticated and non-authenticated users. As a result, Bluebikes can have increased revenue in terms of bikes rented for trips. Worst case scenario for blue bikes is that the platform has no users, therefore they won't be able to benefit

4. Property Owners

Process lens: They are being included in the design decision. However, since we are going to be displaying images of their properties e.g. a restaurant blocking bike lane, people may have negative sentiments towards a certain restaurant. Moreover, since the planning feature is available to all users, negative reviews towards a restaurant may be magnified.

Outcome: Worst case scenario/Best Scenario : Restaurant may get publicity, bad or good publicity.

Second Decision **Adding tags, ratings, to marks**

- Our Users

  Process lens: We made this feature to explicitly include the voice of our users and provide clarity to our users. This upholds our value of **transparency.** Our users understand that to obtain credibility, they must put content in marks that are reliable.

  Outcome lens: The worst case scenario for our users is that the tags are misleading and our users do not find an optimal path for their situation. The best case scenario for our users is that the tags filter marks properly and correctly for our user's situation.

- The government/local municipality

  Process lens: We made this feature universal for all users to see, in particular the government, to increase the impact of **safety** that our app can have to the Cambridge community.

  Outcome lens: The worst case scenario for the government is that the tags, in particular negative ones, indicate the need for action, but the importance of an issue is miscaled and energy allocated into fixing the issue is disproportionate. The best case scenario for the government is that the tags .

- Property Owners/ e.g. restaurant

  Process lens. We made this feature universal for all users to see, including the property owners, to increase the quality of the users to these restaurants.

  Outcome lens: The worst case scenario for the restaurant is that the tags, in particular negative ones, push traffic away from that certain location, reducing the amount of customers coming to that restaurant. The best case scenario for the restaurant is that because of positive tags, more bikers come and restaurants have increased users.

Third Decision **Implementing third-party authentication**

- Our Users

  Process lens: This is meant for our users because we can't handle any personal information about them. This is safe and private because we don't handle any sensitive information therefore users can rely on us. This is a part of our transparency. We included our users Authenticated Users and non-registered users will unders

  Outcome lens: The best case scenario is that the users have a fast way to authenticate rather than creating an account and password. The worst case scenario is that the Google authentication API is broken or no longer in use, through upcoming restrictions.

  The government/local municipality

Process lens: They are not included in authentication because they cannot create an account. This is potentially unfair because they will not have an equal say in the matters pertaining to the overall Cambridge community.

Outcome lens: The best case scenarios and the worst case scenarios don’t necessarily apply in this design decision because they cannot create an account.

Property Owners

Process lens: Both restaurants and property owners can enjoy that users are authenticated by a more reliable source that is not us.

Outcome lens: The best case scenarios and the worst case scenarios don’t necessarily apply in this design decision because they cannot create an account.