Concepts

User

Purpose: keep track of username, password, posted reviews and favorited businesses

Structure:

username: User -> String
password: User -> String
reviews: User -> Review
favorites: User -> Business

Actions:

getReviews(u : User):
    result = u.reviews
getFavorites(u : User):
    result = u.favorites
getUsername(u : User):
    result = u.username
getPassword(u : User):
    result = u.password
changeUsername(u : User, n: String):
    n not in n.~username
    u.username = n
changePassword(u : User, p: String):
    u.password = p
addFavorite(u : User, b : Business):
    b not in u.favorites
    u.favorites += b
removeFavorite(u: User, b: Business):
    b in u.favorites
    u.favorites -= b

Operational Principles:

if addFavorite(u,b); getFavorites(u):us then u in us
if no addFavorite(u,b); getFavorites(u):us then u !in us
if removeFavorite(u,b); getFavorites(u):us then u !in us

Business:

Purpose: keep track of description, tags, and ratings

Structure:

name: Business -> String
password: Business -> String
reviews: Business -> Review
location: Business -> Float[]

Actions:

getReviews(b: Business):

result = b.reviews

getBusinessName(b: Business):

result = b.name

getLocation(b : Business):

result = b.location

getPassword(b: Business):

result = u.password

changeBusinessName(b: Business, n: String):

b.name = n

changePassword(b: Business, p: String):

b.password = p

changeLocation(b: Business, f: Float[]):

b.location = f

Operational Principles:

if postReview(u,r); getReviews(b): rs then r in rs

If no postReview(u,r); getReviews(b): rs then r in rs

Review

Purpose: Gives information about a given business to define the business "goodness"

Structure:

business: Review -> one Business

author: Review -> one User

Actions:

postReview(u: User, r: Review):

u.reviews += r

deleteReview(u: User, r:Review):

r in u.reviews

u.reviews -= r

editReview(u: User, old: Review, new: Review):

old in u.reviews

u.reviews -= old

u.reviews += new

OperationalPrinciples:

if postReview(u,r); then r in u.reviews

if no postReview(u,r); then r !in u.reviews

if deleteReview(u,r); then r !in u.reviews

if editReview(u, r1, r2); then (r1 !in u.reviews) and (r2 in u.reviews)

Tag

Purpose : Organize business by categories for filtering

Structure:

tags: Business -> Tag

type: Tag -> String

Actions:

addTag(b: Business, t: Tag):

b.tags += t

removeTag(b: Business, t: Tag):

b.tags -= t

OperationalPrinciples:

if addTag(b,t); then t in b.tags

if no addTag(b,t); then t !in b.tags

if removeTag(b,t); then t not in b.tags

Map

Purpose: To show business in a given location defined by a map bounding box

Structure:

boundaries: Map -> Float[ ]

Actions:

setBoundaries(m : Map, bounds : Float[ ]):

m.boundaries = bounds;

displayBusinesses(m : Map):

result = Business in m.boundaries

This project would give businesses across Cambridge a way to communicate information to members in the community. It would help them get word out about their status and the types of services they are offering with regards to safety in the pandemic.

Most concepts we have are CRUD operations, but we anticipate the Map concept to take a considerable amount of design work. Our vision for this is to have a “bounding box” map, which dynamically responds to users panning and zooming by changing the businesses that are rendered on the map. This would allow users to discover new businesses in Cambridge more naturally by navigating the map without the explicit need for a search box.