

The conceptual sketch is an outline of the conceptual design of your app. It should be short (one to three pages in length), and should include:

- What key purposes and social needs the project would serve and why they matter;
 - Right now there is not an easy way to evaluate how safe a business is currently without going there yourself. Existing websites that rate businesses do not consider metrics associated with safety during the pandemic
 - This app would allow Cambridge citizens to see how well a business is following city guidelines and how other customers rate their experience visiting said business in terms of safety during a pandemic
 - Helps customers see which businesses are open for specific types of business (curbside only, in store, closed, etc...)
 - Help businesses reach new customers in a time when businesses are struggling
 - Help city track which businesses are following city guidelines
- Outlines of key concepts describing how the app will work, including for each: name, purpose, structure, actions and operational principle (but detailed specifications of actions not required).
 - Customer
 - Purpose: give feedback on businesses
 - Structure: customer → ! name
 - Actions: create, update, delete
 - Operational principle:
 - After create(name), a new user with that name will be created
 - After update(name1, name2), the name of the user with name1 will be updated to name2
 - After delete(name), the user with name will be deleted
 - Business
 - Purpose: share info about how they are handling covid
 - Structure: metrics ← business → ! name
 - Actions: create, update(name1, name2), delete
 - Operational principle:
 - After create, a business will have a name and standard metrics based on city guidelines
 - After update, business.name = name2
 - After delete, the business will be removed/no longer visible to customers
 - Metrics
 - Purpose: indicate what covid guidelines a business is following
 - Structure: business → metric, metric → guideline, metric → confirms, metric → denies
 - Actions: create, update, delete
 - Operational principle:
 - After create(guideline), the new metric will be in business.metrics
 - After update(old_guideline, new_guideline) new_guideline will be in business.metrics and old_guideline will not be in business.metrics
 - After delete(guideline), guideline will not be in business.metrics
 - Confirm/deny
 - Purpose: verify whether a metric is being followed
 - Structure:

- customer --> metric
- Actions: confirm(metric, customer, business),
unconfirm(metric, customer, business), deny(metric, customer, business),
undeny(metric, customer, business)
- Operational principle:
 - confirm(metric, customer, business) increments the number of confirms associated with a business' metric
 - unconfirm(metric, customer, business) decrements the number of confirms associated with a business' metric
 - deny(metric, customer, business) increments the number of denys associated with a business' metric
 - Undeny(metric, customer, business) decrements the number of denys associated with a business' metric
- Customer comfort rating
 - Purpose: summarize customer comfort with a given business
 - Structure: user – customer comfort rating → ! business
 - Actions: rate(customer, business, rating), update(customer, business, rating), delete(customer, business, rating)
 - Operational principle:
 - After rate(customer, business, rating), the rating of the business will be updated with rating
 - After update(customer, business, rating), the comfort rating of the business by the given customer will be updated to rating
- Comments
 - Purpose: customers communication with a business
 - Structure: customer !----> comment ----> !business
 - Actions: create(customer, business, comment), update(customer, business, comment), delete(customer, business, comment),
 - Operational principle:
 - After create(customer, business, comment) comment is in business's comments
 - After delete(customer, business, comment) comment is removed from business's comments
- Violation
 - Purpose: automatically flag businesses that are not following city guidelines based on their metrics votes
 - Structure: metric <---- violation --> ! business
 - Actions: create(business)
 - Operational principle:
 - after create(business), the violation is added to the set of violations reported to the city
- Statutes
 - Purpose: indicate at what capacity a business is open (take-out only, outdoor seating, etc)
 - Structure: status !<---business
 - Actions: create, update(business, status)
 - Operational principle:

- If a business updates their status from A to B then that business will be removed from set A, and added to set B
 - If a business creates a status A then `business.status = A`
- Why the project will involve interesting and substantive conceptual design work (in particular that the app is more than just [CRUD](#); that it involves at least one concept that is not already widely used; and that the design problem is non-trivial).
 - We created a violations concept that publicly indicates if a business is not following Cambridge's COVID guidelines. This is not a widely existing concept, especially in the context of COVID.
 - This app is addressing emerging problems with the pandemic, and with concepts such as metrics, statuses, and violations we are addressing these problems in a non-trivial and novel way.