Progress Report

We have a basic UI, but it is not completely hooked up to the backend yet. We also have all basic backend functionality working. This includes the following:

- Create, edit an account
- Add and remove a bike
- Edit a bike's location, availability, price
- See all bikes unfiltered, filtered by location, filtered by availability, filtered by both location and availability
- Create, cancel, reserve, use a ride

Achieved/~~Missed~~ Milestones

All Proof of Concept milestone tasks were completed. This milestone focused on the backend, with a basic frontend to test requests and responses. This includes the following tasks:

- Implement accounts
- Implement bike functionality
- Implement ride functionality
- Implement basic UI and integrate with backend

Difficulties encountered

Availability tracking

Our biggest struggle right now is trying to find the best way to keep track of bike availability -- when they are free, being used, when the time slot opens up again, etc. We thought about using a linked-list data structure to store that information, but it wouldn't be compatible with a database. We're looking into an API that might be able to manage it for us, but any suggestions would be fantastic!

- Have a representation with multiple entries for each bike to represent each 30-minute block (or smallest unit of time for renting a bike) a bike is available or being rented out; basically lots of entries for every bike
    - Smallest unit of time for renting a bike: 30 min?
    - Query: Available WHERE type of bike, time values for start and end, etc.
- Limit so that users can only create bike listings available for the next few months (since we're making tons of entries for each bike)
- If we want to explore linked list idea further, can try using NoSQL db, but will be harder to implement

Changes to design/project plan

We received feedback about our security process and review concept, and will make the following adjustments.

Security

Our original plan was to have some sort of GPS tracking system and alarm for each bike, so it could be found if stolen. Feedback suggested that this was a bad idea, so we'll be implementing the following instead (suggestions welcome about which approaches we should keep/remove):

1. Ask for a full name and phone number. This should be enough information for people to contact the authorities about damages or stolen bikes.
2. Late fees for returning a bike late, that adds up to more than the cost of the bike if not returned. Alternatively, a hold charge like hotels do, in case a bike is damaged or stolen.
3. Bike renters must take a picture of the bike when it's returned as a precaution.

- Taking pic of bike for when it's returned:
    - for MVP, users store their own photos and provide a link to their photos, and add the link to (Google Photo Link) and we can just display it
    - Uploading photos for every ride isn't super user friendly, but it's sufficient for our purposes!
- Note about photos for our app: we can't just upload photos to Heroku without an AWS instance, so maybe shouldn't upload straight to Heroku (otherwise, it would reset)
- Name + phone number is fine
    - Ignore gov't IDs for now, maybe a reach goal? (We wouldn't have to worry about verification - just uploading a pic of ID would be sufficient)
- Late fees as another deterrent is good!
- For returning a bike, implement an electronic handshake system where both the renter and rentee must click the "hand-off" button within a certain period of time: the renter when they're done with the bike, and the rentee when they see that the bike is where it should be

Reviews

This idea was originally built into our Account and Bike concepts. We will move the concept of a review into its own separate concept after receiving feedback. We will still be doing a weighted rating system and reviews for both users and their bikes. One feature we'll add is to allow users to upload pictures to support their reviews.

- In posting a bike, remove "latitude" and "longitude" fields and just turn it into a map thing where people can pin locations - check Google Maps API for pinning locations

- Figure out query into db to easily get when bikes are available (or not available - so that if there’s a date/time in which a bike is not available, we can easily grey it out on the map when filtering)
- EPOC timestamps for availability - maybe too detailed?