LTP Bike Logger – Final Design Write-up

David Williams

Problem Statement

The problem I chose to address is one that I observe on a daily basis. Bikes are used by many of the students and faculty on campus as a means of transportation, but all too often bikes are abandoned by their owners. Sometimes they are simply forgotten, sometimes they are scrapped for parts and the owner doesn't want to dispose of the frame properly, and sometimes they are buried in snow to be forgotten. However, when too many bikes are abandoned, valuable space on the bike racks that could be used by other people is being taken up. Technically the University policy is that any bike left alone for 48 hours is supposed to be impounded, however this is rarely executed in a timely manner. Thus, I set out to solve the problem of abandoned bikes not being taken care of properly.

Solution Overview

My solution is a mobile app designed for the Logistics, Transportation, and Parking (LTP) department at the University of Michigan. It maintains a customizable list of the bike racks on campus, and automatically tracks bikes that are impounded. Each bike rack maintains a list of individual bikes and sorts them by the amount of time since each bike was added to the list,

making it easy to identify bikes which have been abandoned for 48 hours or more and are subject to be impounded.

Finalized Design

Here is a breakdown of each screen and what their interactions are:

Log In

- Two textboxes, one for uniqname and one for password.
- Log In button, tap to submit log in information.

Home Screen

- Bike Racks button, tap to go to the Bike Racks screen.
- Impound List, tap to go to the Impound List Screen.

Bike Racks

- +Create New Bike Rack button. Tap to create a new bike rack to be listed on this screen.
- A list of the created Bike Racks. Each entry can be tapped to go to that bike rack's personal menu.

Create New Bike Rack

- Two textboxes, one for the name of the new bike rack and one for a more specific description of the new bike rack. Both must be filled out before the Create button is usable.
- A Create button, which is only usable after both textboxes are filled out. Adds an entry for the new bike rack on the Bike Racks screen.
- A Cancel button, which can be tapped to go back to the Bike Rack screen at any time without the new rack being created.

Specific Bike Rack (i.e. CC Little)

- A +New Bike Button which can be tapped to add a new bike to the list on the screen.
- A grid view listing the bikes at the bike rack. Each square in the grid has a picture of the bike, and a number of hours listed underneath. The list is sorted by the number of hours (highest to lowest), and bikes with 48 or more hours having red text (as opposed to white).
- A Home button, which can be tapped to return to the Home Screen
- A Delete Rack button, which can be tapped to prompt a dialog about removing the bike rack and all of the bikes in it (not functional in the final prototype as it wasn't necessary).
- A Back button, which can be tapped to go back to the Bike Racks screen.

Specific Bike from a Bike Rack

- A picture of the bike.

- Text describing the brand and any identifying features of the bike (if they were entered when the bike was created).
- A Back button, which can be tapped to return to the specific Bike Rack screen this bike is saved in.
- A Delete button, which can be tapped to prompt a dialog asking about deleting the bike.
- If the bike has 48 hours or more on its timer, an Impound button which can be tapped to prompt a dialog asking about impounding the bike (removing it from the previous Bike Rack and adding it to the Impound List).

New Bike (before getting to the screen)

- The camera will be launched and a picture must be taken. This is to ensure that a picture of the bike is added to the entry, as it is the most reliable way to identify a bike at a bike rack.

New Bike (after getting to the screen)

- A picture of the bike.
- Two textboxes. One to enter the brand of the bike, and another to list any identifying features it has.
- A Create button, which can be tapped at any time to add the bike to the Bike Rack list it was created in.

Impound List

- A grid view listing the bikes that have been impounded. Each square in the grid has a picture of the bike, and the name of the rack it was taken from underneath. The list is sorted by bike rack alphabetically.
- A Home button, which can be tapped to return to the Home Screen.
- A location filter. Tapping this opens a dropdown menu of all of the bike racks. Selecting a specific rack restricts the list to only show bikes that were impounded from that rack. There is a default option to show bikes from all bikes rack at the top of the dropdown list.

Bike from the Impound List

- A picture of the bike.
- Text describing the brand and any identifying features of the bike (if they were entered when the bike was created).
- A Delete button, which when tapped, opens a dialog about deleting the bike from the Impound List.
- A Claimed button, which when tapped, opens a dialog about claiming the bike.

Functionality

The app is largely focused on two areas, the Bike Rack and the Impound List (Figures 1 and 2).

Bike Rack

Here is where you can see a list of all the bike racks on campus. There is a button to add a new bike rack. The only information you need to fill out when you make a new rack is the title (what to call it) and a description of its location so that it is easily identifiable (i.e. North Quad, diag side). When you select a bike rack, you'll be brought to its menu. Most of the screen is taken up by a grid showing the bikes. Each square in the grid has a picture of a bike and shows how long it's been in the list in hours. If the time is more than 48 hours, the text is red to indicate that the bike should be impounded. Here you can select any bike to see more information and impound or delete it (only bikes with a timer of 48 or more hours can be impounded), and you can create entries for new bikes. When you create a new bike you must take a picture of it so that it can be identified, and then you can enter optional information that might help identify it. Once you create a bike, it will automatically be added to the bottom of the grid on the rack screen.

Impound List

The impound list has a similar layout to a bike rack screen. All of the impounded bikes are listed in a grid, but instead of having a time under them they have the name of the bike rack they were impounded from. There is an option to filter by bike rack, so you can do things like only view the bikes impounded from the North Quad bike rack. Any individual bike can be selected to view more information about it, and to allow it to be claimed or deleted.

The only feature not implemented is the ability to delete bike racks. While it would certainly be practical as bike racks change over time (new ones are built, old ones are removed, etc.), the main purpose of the app is to track bikes. As a result, I focused all of my efforts on that, as the

ability to remove a bike rack would not have really contributed to the problem I was trying to solve.

Sketching Techniques and Approaches

From my initial sketches I didn't have very many ideas I liked. Most of them were variations on maintaining lists of the bikes at each bike rack somehow, either with paper, a website, or an app like I settled on. The only other concept that I really like, and is actually what I initially wanted to use, was to create new bike racks that could keep track of the bikes automatically. Then the only work that the workers would have to do would be to collect the bikes, however when considering the practicality of implementing the design I couldn't justify using this approach.

After the initial sketches I made, the storyboarding process is what really helped push me to realize what my design needed to do. Before this, I hadn't really considered what happens to the bikes after they are impounded, only how the LTP workers could keep track of them while they were still at the bike racks. After simulating the process of collecting the bikes though, it became clear that there needed to be a system for keeping track of the impounded bikes, and it wouldn't make sense to have it be separate from my existing implementation, so it became something I implemented.

By the time I'd settled on the features my design needed to support, I still hadn't decided between an app and new bike racks. Ultimately both would be able to do the work of tracking the bikes, and it would be a lot easier with the new racks. However, the cost of implementing the racks would be much higher and they still wouldn't be able to keep track of the impounded bikes. Making half of a solution for a higher cost was hard for me to justify, so even though the decision to make an app resulted in more work for LTP employees to keep track of the bikes, I decided it was the better option.

Design Process

I don't think I had a lot of major changes in my design once I'd settled on an app as my solution and started sketching it. Some of the layouts of screens changed after testing my paper prototype, but other than that no features were really changed. I hadn't really considered the impound list until I did the storyboarding and realized that the people claiming the bikes would have to do keep track of them, but that was before I'd started sketching my solution.

The biggest change that did happen was a result of the paper prototyping though. I knew that I wanted the bike entries to have a picture before they could be created, but originally I left a button to confirm the creation of a bike inactive until a picture was taken. It was clear from the user testing that no one knew the picture was the requirement to be able to create a bike, so I changed my approach. The button to create a new bike now immediately opens the camera on the phone, making it clear that a picture must be taken. Once the picture is taken, other

information can be added, but the button to create the bike is always available because the picture requirement is fulfilled before reaching that screen. While it's not the most drastic change, I think it was the one that most improved my design from my paper prototype.

My Role as a Designer

I've never really designed anything before so I'm not sure if I really have a design philosophy yet. Coming from an engineering background though, I did know that I wanted it to be easy to use, and that it should be simple to understand. For example, I phrased my dialog prompts for deleting bikes to be very specific yes or no questions so that the user clearly knows what is being asked of them and their available responses are logical. Overall I suppose my approach was very focused on usability, but not as much on aesthetic. It does look pretty boring, but I also didn't have a lot of time to get familiar with a prototyping tool so it was the best I could do. I wasn't really trying to embed any societal norms though. At least none that I knew about. I ended up using the plus sign to indicate buttons that created things, but I'm not sure if that would be considered a societal norm. Other than that, I'm not really sure if my design enforces any other societal norms, especially because that was never my intention.

Impact

My solution will keep the bike racks clear for those who need it all year round. By removing abandoned bikes in a timely manner, everyone will be able to use the bike racks as intended

when they need to. This will ensure that as many people as possible can ride their bikes daily, and that the bike racks will no longer be cluttered with frames of bikes that have been scrapped for parts, or by bikes that have simply been abandoned by lazy owners. This not only makes life better for bike riders, but also beautifies our campus a little bit, which everyone can benefit from. By keeping track of which rack impounded bikes come from, it will also be easy for people to claim their impounded bikes. While the process of picking up impounded property is never fun, having the app automatically maintain the bike's location and being able to filter the impound list will make it easy for someone to find and identify their bike when they come to claim it, making the process as painless as possible.

Appendix

Figure 1

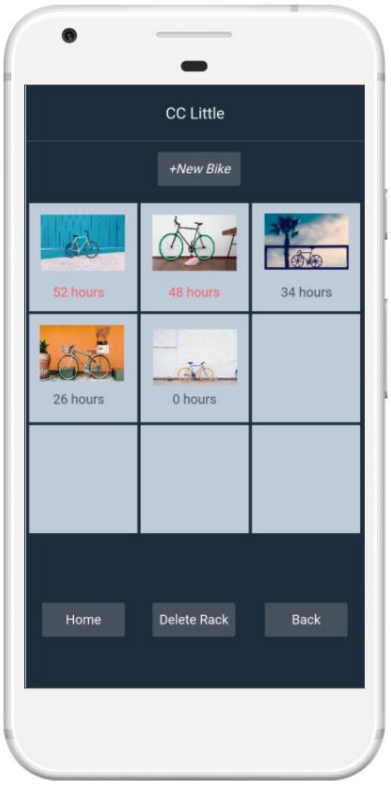

Figure 2