Matt Schroer & Kyle Vinagro
Final Report

FindRack

Problem

Bike riders of Boston are an oft-abused, neglected people. Forced to ride on cramped, narrow, winding streets, they often arrive at their destination frazzled and running late. Much to their chagrin, the bike racks they need to secure their bikes are often full by the time they arrive, forcing a desperate search for an alternative spot. This hunt for a bike rack takes precious time, and often results in the cyclist being late for work or class. For users of Boston's Hubway system, there a multitude of apps showing the locations of nearby Hubway stations and the empty spots at each. But for the common man, owner of their own bike, there is nothing.

The target users of our application are the bike owners of Boston. Our target users depend on bike transportation, and tend to have an established daily pattern of traffic. They most likely already have favorite bike racks, but would like a better way of tracking their availability. Other users are perhaps new to the area, and need to discover the bike racks that are most convenient to their life.

Design

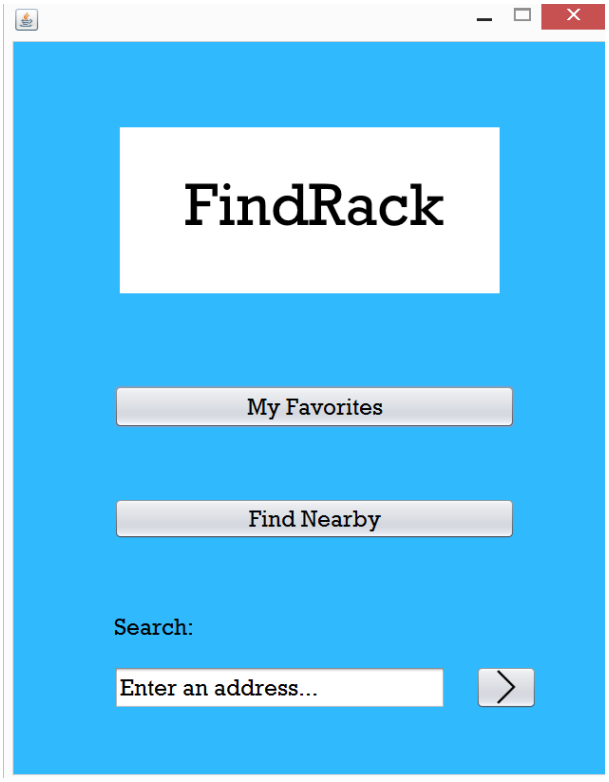
The final design of our project bares a large resemblance to our initial prototypes. We knew from the beginning that we wanted our application to be simple; something that could be operated one-handed while riding a bicycle. As a result of this decision, we did not have to undergo a radical re-design We made a few simple design changes over the course of the project such as replacing the star buttons with checkboxes to make it more obvious that they were clickable. We also modified the table's background color to better match the rest of the prototype's design, and removed the "OR's" from the home screen to reduce unnecessary clutter. Both of these design changes were based on feedback we received in the computer prototyping and user testing portions of the project.

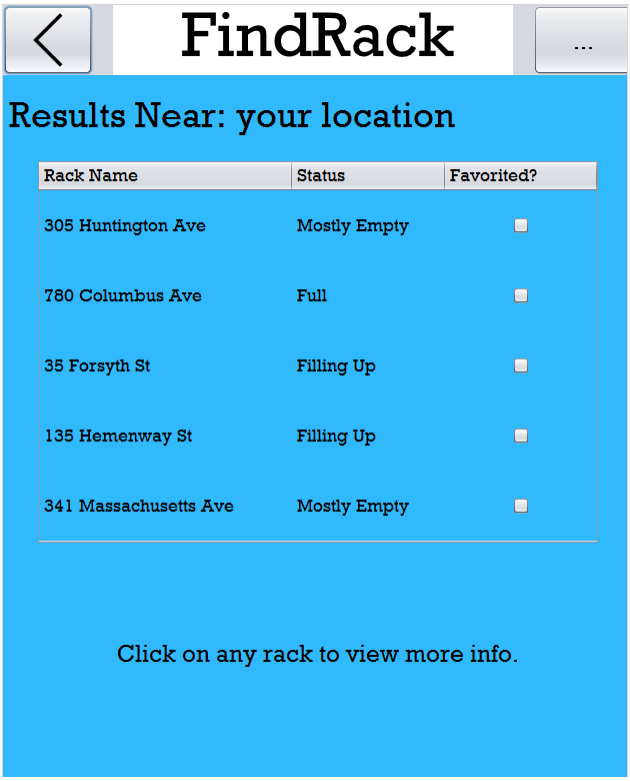
We chose the color blue as our main design color for its aesthetic properties and its readability. This is also an under-utilized color in mobile applications, which we thought would

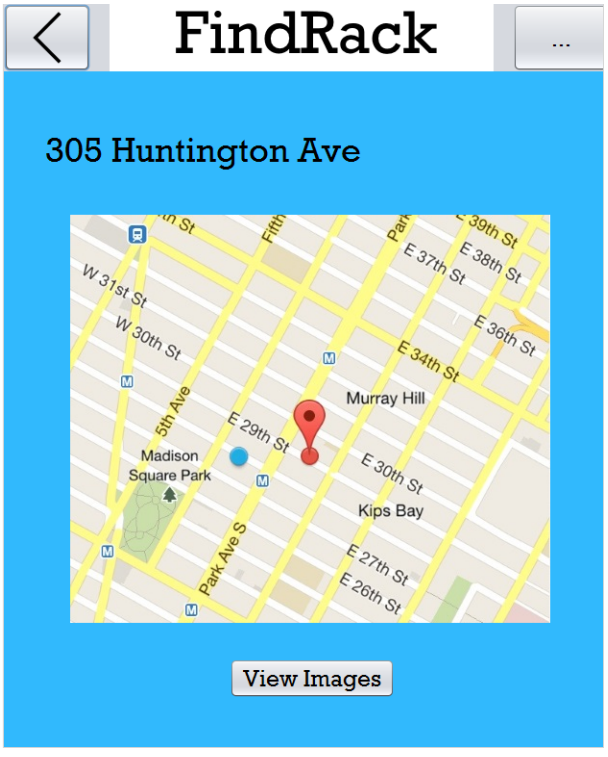
allow our application to stand out. Likewise, we chose the font Rockwell for its distinctive style and overall readability.

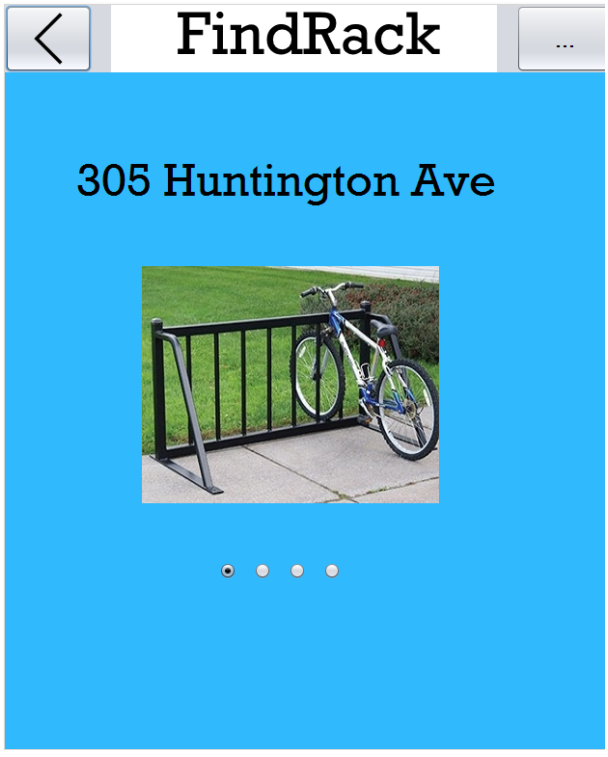
We ended up sticking with the same basic layout and structure throughout the whole project. We did not really consider any alternative designs throughout the process because the one we had been using was very simple and easy to follow. It also falls in line with many other mobile applications and provides external consistency which is very important when designing new applications.

The final prototype:









Implementation

Our application was initially designed to be a mobile application, ideally on the Android operating system. Based on past experiences with creating Android applications, we decided the easiest approach was to mock-up our design as a Java application. The final application was built using Java Swing, in a form factor meant to resemble a mobile application. As a result of this implementation decision, some features work slightly differently than initially envisioned. The most notable difference is on the Images page, where the images would have been swipeable on an Android device. This behavior was not recreatable on a desktop application.

Evaluation

As part of the development process, we underwent three rounds of testing focusing on our product in different stages of development. The first type of testing we did was paper prototyping. During this stage we had multiple people use a paper version of our application while on of the team members played computer. We learned a great deal about what worked and what did not work with our prototype. We learned that it would be important to make it very obvious what was clickable and what was not clickable in our application. There were a few instances of our subjects trying to click on the logo at the top of the application, but then when they were faced with a list of bike racks to choose from, they did not immediately recognize that they could click on them. We also originally used stars and colored dots to represent the fill level of racks and the favorite status, but users were unsure if these were just icons or if they were buttons, so we dropped this idea for our first computer prototype. Overall, we were able to keep our basic application design from our paper prototype, but we had to make some usability tweaks to make it more user friendly.

The next round of testing involved giving our computer prototype to classmates and allowing them to go through the heuristic evaluation steps. This is where we received most of our feedback and this gave us the most ideas and recommendations for changes. Some users were still having trouble recognizing that the different racks were clickable, so we added a message at the bottom of the list view screen informing them that they could click on the racks. We also were informed of a bug in which the search text field would not re-populate automatically when if you went to a different screen and then came back, which confused some users or allowed them to forget what they should be entering in that field. We also were informed of a number of cosmetic issues were fairly easy to fix. We fixed an issue with radio

buttons moving around on the imageview screen and we also fixed issues related to window resizing, and the rearranging of columns on our listview screen. This round of testing was immensely helpful to us and we were able to focus our efforts on the issues that the most number of users reported to us.

Our final round of testing involved another round of user testing like we did with our paper prototype, but this time we gave the users a current version of our application. While we did get a decent amount of feedback from this round of testing. The problems that were reported largely mimicked the problems from the heuristic evaluation that we had not gotten around to fixing yet. We also made some cosmetic changes based on user feedback and changed the color of the table on our listview screen to better flow with the rest of the application. We also implemented a dialog popup that would display if a user clicked on the menu button that would inform them that this was not implemented yet. We also hoped that this would help with the issue of the racks not seeming clickable.

At the end of our testing, we were able to identify and correct many issues with our application, but we were not able to fix all of them. One major issue that remains unresolved involves favorites not disappearing immediately when they are unchecked on the favorites screen. We were not able to fix this problem because we do not have enough knowledge of Java Swing to correct it. If we were to redesign the application in a program with which we were more familiar, we may be able to fix this problem. We also neglected to add a quick button for favorites on every screen because we could not determine a good spot for it and it would break internal consistency.

Reflection

Over the course of the iterative design process, we learned a lot of details about the different stages of the process and what you should expect to gain from each stage as well as how all of the stages tie together. If we were to do this project all over again, the biggest thing that would have changed is the architecture on which we built the application. Neither of us had enough experience with Java Swing to efficiently build our application. We had to spend a lot of time researching how to do different things in Swing and this took away from our development time.

We also learned that the paper prototyping stage was very important to the process, not because it provided us with an abundance of feedback, but it gave us the basic layout and structure for our application. We were able to basically keep the same structure throughout

the entire project and just certain things about it as opposed to redesigning it as a whole. The paper prototyping phase did give us some useful feedback, but we found that we did not get the most useful feedback until we actually had a product to be tested. This began in the heuristic evaluation phase. To us, this was the most important phase because we received a lot of feedback from it and the majority of it was very useful for improving our application. We also found that organizing the feedback from this stage and prioritizing it helped immensely when deciding which issues to tackle first. Moving on to the user testing phase, we found that most of the feedback that we received was the same issues from the heuristic evaluation that we had not been able to fix yet. This was not as useful as us, but it did have some influence in how we tackled certain issues. Overall, all three of these prototype techniques were very valuable and provided us with more than enough feedback needed to improve our product.