Peer Review: BikeLINK

This peer review will cover BikeLINK, a system intended to support a bike sharing service for Newplace. In this report, I will cover the design priorities, trade offs, and evaluations of the system, and additionally provide some insight on improving the system.

BikeLINK's main design principles are reliability and performance. The system is divided into different layers: physical infrastructure layer (hardware components), network communication layer (wireless network communication), information flow layer (communication between databases), central control layer (backend tools and databases), and application layer (app, website, kiosk). This modularization makes the system easy to understand (p. 3-4).

Most design choices are made for reliability. For example, the system will wait for a bike to finish uploading its video to the datacenter after returning for a worst case of 3.5 minutes of unavailability; the authors chose to emphasize reliability over availability to prevent video data loss (p. 5). Additionally, the authors chose to use many protocols like TCP, HTTPS, and RAID to promote reliability; this emphasizes reliability over performance because these protocols require additional time to ensure reliability (p. 8).

Performance is mentioned to be prioritized when choosing WiFi over other communication, but the justification does not cover the worst case of having a very slow transfer rate of 20 Mbps (p. 7). Rather than performance, the authors seem to prioritize cost. The authors chose to have no basic bikes as they predicted not a lot of members would opt for it (p. 4). They also chose to have a high abandonment fee of $1000 because of the missed profits (p. 12). Perhaps cost could also be mentioned as a design priority with performance and reliability.

As for specific processes:

- Reserve dock for return (p. 10-11): Members first submit a station location. The system checks that their reservation count is less than 4 and that the payment method is valid. If so, it increments the reservation count and sets the dock status to sleeping until 15 minutes before the reserved time. If the member does not drop off a bike at the dock by 15 minutes after the reserved time, the dock is reset as available. Otherwise, the user will be charged for the ride once the bike is docked.
- Surge of returns at station next to superstation: This procedure wasn't explicitly analyzed, but based on my understanding, the angel tool would ask potential angels to move bikes from the overcrowded station to the nearby superstation.

For the return reservation, it would be useful to note how a user indicates that they are making the return; right now, it seems like the user just needs to push the bike into the selected dock. However, this also brings up the question of what happens if a member who did not reserve the dock pushes their bike in. There should be an explicit check to determine that the docked bike is actually under the member that made the reservation; otherwise, the return should not go through. For the surge of returns scenario, adding the exact procedure would be helpful for future readers.

For evaluation, one section was dedicated to calculations for waiting times for uploading video data. They calculated that the average worst case (30 minutes personal video, 1 accident video) would take 2 min, 50 sec and that the absolute worst case (30 minutes personal video, 15 minutes accident video) would take 3 min, 33 sec (p. 14-15). They reasoned that since only 10% of bikes have cameras and because these times are not that long, it would be fine to wait for the upload to enforce reliability. I think it would also be interesting to see the evaluation for the different protocols for reliability (like TCP vs. UDP or RAID vs. GFS) and why these particular protocols were chosen.

Overall, the design priorities were well defined and there was plenty of justification for reliability and cost decisions; the system layers also made the design very intuitive. The procedures for certain use cases and edge cases/limitations could have been more exact, as well as additional evaluation (most of the evaluation was reasoning if the design was feasible, but it would also be useful to reiterate evaluation on design choices in this section).