AIR

A System for Accessible Intracity Riding

6.1800 Design Project Report

Authors: Jonatan Fontanez, Jesus Diaz, Fabian Yañez

WRAP Instructor: Kate Parsons

Recitation Instructor: Mohammad Alizadeh

May 6, 2024

1: Introduction

The city of Newplace is seeking to implement a bike sharing system to make transportation more affordable and enjoyable for its 1 million citizens. To accomplish this, we propose a system for **A**ccessible **I**ntracity **R**iding (AIR) which enables users to borrow, return, and reserve bicycles. AIR aims to solve the issue of limited reliable and low-cost transportation options available to citizens. This document outlines the process for renting and returning bicycles, defines a framework for the collection and processing of user data, and details a number of additional features implemented in AIR to improve the user experience.

In order to better serve the needs of the people of Newplace, our system will prioritize *accessibility* and *privacy*. We choose these as our primary and secondary goal to maximize the amount of users that will be comfortable using our system.

Our primary goal is *accessibility*: the ease with which people can access and use our service. It is imperative that our system can meet the needs of as many users as possible, whether that's a daily commuter, a tourist, or anything in between. Users should be able to use our service whenever possible, including during short (<2 hour) power or network outages.

Our secondary goal is *privacy*: the secure storage and transfer of personalized information that users entrust us with. From GPS trackers to High-Definition Cameras, our bikes will be equipped with powerful devices capable of recording personalized information. As such, it will be crucial for our design to carefully consider the kinds of data that should be collected, and who should have access to this data.

In Section 2, we will introduce the system overview of AIR at a high level. In Section 3, we will go into more detail about specific aspects of system modules highlighted in Section 2 and how they meet our priorities of *accessibility* and *privacy*. In Section 4, we will discuss future implications and possible changes to consider.

2: System Overview

AIR implements a bike borrowing system with a centralized storage database and processing center to collect important data used for path finding, reservations from users, and provide bike availability so users can plan around their day.

All individual components are organized into 5 different modules, which can be organized into 3 categories:

- *Central Computing Facility (CCF)*: A large, central database and processing center to store, process, and share important system data and anonymized user data.
- *Physical user access*
    - *Stations*: A collection of bikes docks with connectors for data and/or chargers for ebikes, and a kiosk for user interaction.
    - *Bikes*: Basic, standard, and e-bikes that users can borrow or reserve using the kiosks, app, or website.

- *Digital user access*
    - *Mobile app*: Another way for users to access the system, with most features reserved for members.
    - *Website*: Another way for users to access the system, though more limited in features than the app.

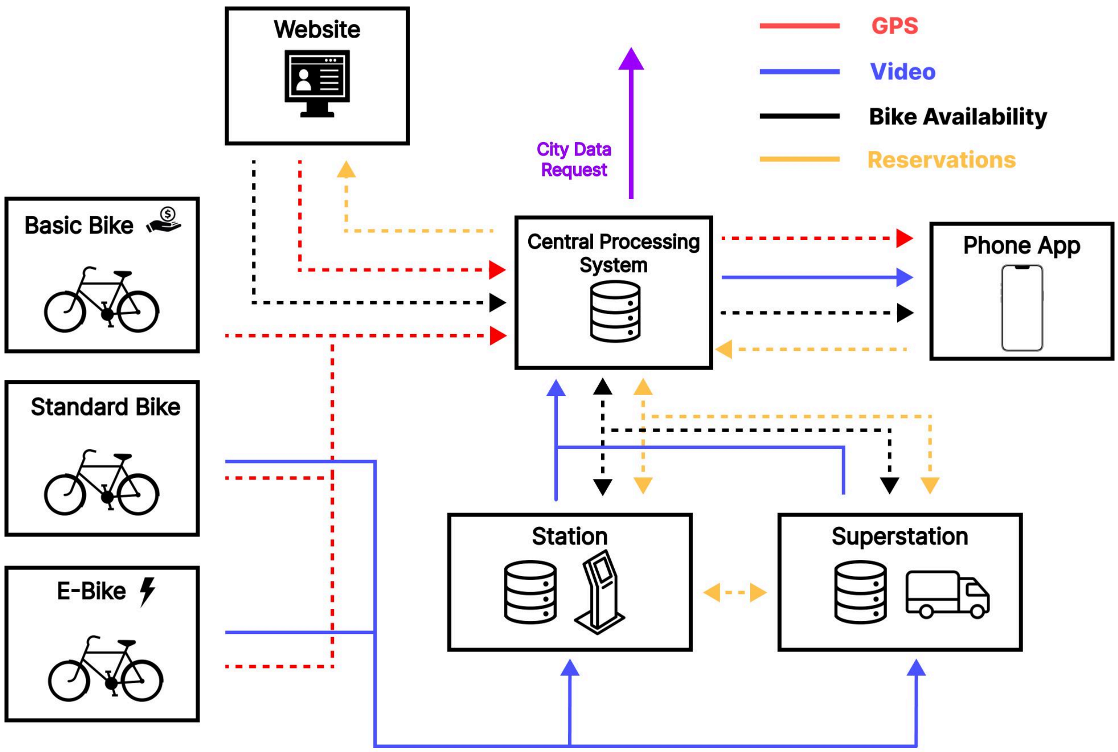

Figure 1: A High-Level System Diagram for the AIR System

3: System Design

3.1 Central Computing Facility (CCF)

As the central hub for data transfer and storage in our system, the CCF plays a crucial role in our commitment to *privacy*. Below, we analyze how our system handles incoming data and stores personalized information to prioritize this goal.

3.1.1 User Information

Member login information is stored in our system as a table of entries, *user_login*, with (1) the user email (stored as a 256-byte string), (2) a string of random data (the salt), and (3) a password that has been encrypted by the concatenation of the salt and the use of a hash function (such as SHA-256). These three methods will ultimately give us an extra layer of security to prioritize *privacy*. Although storing this information comes at a tradeoff of storage, we argue that for our target of 50,000 users the size of this data would still be negligible compared to the 100 TB of storage available:

$$(256\ byte\ user\ email\ +\ 32\ byte\ hashed\ value)\ *\ 50,000\ users\ =\ 14,400,000\ bytes\ =\ 14.4\ MB$$

Additionally, we will maintain a separate hashmap stored on disk called *user_info* to organize general member information. We assume that both bikes and stations are assigned unique, positive 16-bit integer IDs, which would easily allow us to account for all bikes and stations for our projected 50,000 users. Each user will have a 256-byte string email address mapped to an entry with the following information:

- *user_bike*: tracks the bike ID that the rider currently has rented out.
- *bike_reservation*: tracks the bike ID that the rider currently has reserved.
- *dock_reservation*: tracks the station ID that the rider currently has reserved.
- *payment_information:* an object that stores a card number (16-digits long), expiration date (4-digits long), and security code (3-digits long).
- *user_video*: stores a pointer to a video from the user's most recent trip, if it exists.
- *angel_ride:* 1-bit boolean that tracks whether or not a user's current ride would classify the user as an *Angel*. Set to 0 by default.
- *angel_extensions*: integer representing the number of 15-minute extensions the user has. Initialized to 0 when a new entry is created.
- num_reservations: integer representing the number of reservations a user has made in a day, reset to 0 every day.

All unused variables will be represented with a -1. Given the sensitive information that this table contains, such data would need to be encrypted in order to protect user *privacy*. Taking all the respective sizes into account, each entry would take up roughly 320 bytes of information. If we assume that encryption would roughly triple our storage size, we could estimate the total amount of memory we would need to store this information:

$$320\ bytes\ *\ 50,000\ users\ *\ 3\ encryption\ multiplier\ =\ 48,000,000\ bytes\ =\ 48\ MB$$

This is well within the 100 TB memory limit on the CCF, with plenty of buffer room for additional users in the future.

3.1.2 GPS Location Data

For the purposes of this system, GPS location data will be sent to the CCF in the event that a bike has been identified as a "lost bike" (see section 3.2.2).

3.1.3 Video Data

The CCF communicates with stations to receive video data. Upon receiving this data, the CCF immediately stores the video in memory. If a user wishes to access this data, they will have a week to log into their account via the mobile app or website to access this data before its data and pointer are deleted from memory. To protect the *privacy* of users, the video data will be deleted after the user downloads its contents.

Given the accumulation rate of 45 MB/s and having a majority of the original 100 TB of storage left to use for video storage, we can store approximately 600 hours of video footage taken straight from the cameras with no additional compression (see the equation below). Given that accidents are rare and that only a small percentage of the initial bikes have cameras, the total accumulation rate of video across the entire system will be low enough for the current storage to handle. If the video storage exceeds expectations, the storage of the CFF can be adjusted accordingly. The increase is expected to be approximately linear with respect to the difference in the expectations and the actual values, so storage expansion should not be a significant cost.

$$\sim 100\ TB\ * \frac{1s}{45\ MB} * \frac{1hr}{3600s} = \sim 617.28\ hours$$

Compression allows more video to be stored in the same amount of space in a lower-quality format, losing some details of the video and audio. To increase the storage space available to the users for personal video, after 12 hours we will use MPEG compression to compress the video. This will compress the video down to approximately 1.5Mb/s according to ISO 11172[1]. This results in a compression ratio of 240:1, which is extremely high. Using this, we will be able to store a maximum of approximately 148,000 hours if all video is compressed and all video stored is personal.

$$617.28\ hours\ * \frac{240\ MPEG\ hours}{1\ uncompressed\ hours} = 148,147.2\ hours$$

We chose to store the video uncompressed for 12 hours as this should give users enough time to access their full quality video. If they rode a bike to work, they would be able to access their videos during their lunch break (~4 hours after using the bike), after they get out of work for the day (~8 hours after using the bike), or when they get home. They should have gotten home before 12 hours in most circumstances, and they would be able to download it earlier if they have access to the app prior to the 12 hours.

Encryption increases (1) the minimum time between recording a video and the user having the ability to access it and, (2) the total storage size of any video. To remedy this, we do not encrypt personal videos until they are compressed. This means users can access their videos faster and store more personal videos on the CCF, increasing *accessibility*, but it also reduces *privacy*. If a malicious hacker were to be able to access the CCF, they would be able to access all personal videos that have not yet been encrypted. This is a tradeoff we are willing to make due to the extremely large file size difference between unencrypted and

---

[1] International Organization for Standardization. (1993). *Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s* (ISO Standard No. 11172:1993). https://www.iso.org/standard/19180.html

encrypted video. To compensate for the loss in *privacy*, we encrypt videos after they are compressed. This ensures older videos are more secure and newer videos do not take up too much space, while allowing users to access higher-quality versions of their personal videos before the 12 hour cutoff.

Users will not need direct access to their accident videos; only Bikes4All and any legal team handling evidence for a case involving the accident would require access to these videos. Because of this, we no longer need to prioritize *accessibility*. As such, accident videos will be automatically encrypted as soon as possible and stored in the station closest to the bike with sufficient network access and storage space. Additionally, to take advantage of the storage available on stations and to ensure the highest-quality possible for accident videos, they will not be compressed. We do not store most accident videos on the CCF; if any accident video is required, it can be requested from the stations by the CCF and stored temporarily on the CCF before being transferred to any other analysis teams, either from Bikes4All or a legal team. In this way, we can leverage the additional storage space across our stations and ensure that accident videos are kept in their original (uncompressed) quality.

3.1.4 Bike Availability

The CCF keeps two hashmaps: *available_bikes* and *total_docks*. The first maps every station to its corresponding list of available bikes, while the second maps every station to the maximum number of bikes it can hold. See Section 3.4 to see how often this information is updated. The CCF gets the data from the angel tool algorithm in the form of (bike_ID, target_station_ID) pairs and forms a hashmap to store the relation from bike IDs to a pair of stations. This pair includes the station it is currently stored at and the station it should be stored at. The second station is known from the angel tool algorithm, and the first station can be retrieved from a request to each station to see where a bike with a specific ID is currently at.

3.1.5 Reservations

When a member requests a reservation of a bike from station S_1 to be dropped off at station S_2, a request is sent to CCF to check the *available_bikes* entry for S_1/S_2, and the *total_docks* entry for S_2. If S_1 has an available bike, the number of available bikes for S_2 is less than the number of total docks for S_2, and the member's *num_reservations* entry is less than 4, the reservation is created successfully and *num_reservations* is updated. The CCF sends a message to S_1 to reserve a bike and a message to S_2 to reserve a dock. After confirming that these messages were received, the CCF will let the member know that the reservation was checked and created successfully.

3.2 Physical Access

3.2.1 Stations

Stations allow users to pick up and drop off bikes. Additionally, each station will keep track of:

- The total number of bikes it can store, as an integer.
- The bikes currently stored, as a list of bike IDs.
- Any reservations made by users through the app, the website, or another station's kiosk, as a list of pairs of user emails and reserve times; this will be two lists, one for bikes and one for docks.

- The health of currently stored bikes, as a mapping from bike IDs to bike health information.
- Its GPS location, represented with the latitude (from the Equator) and longitude (from the Prime Meridian) in degrees as a floating point number, and a time variable. The time variable is either 0 if the station is a normal station, or the time when a station session is set to end, if it is a superstation.
- A map of user emails to the number of reservations that the user has made.

When a bike is docked and connected to a station through the USB 3.1 cable, the station will request data stored on the bike in the following order: bike ID number, health information, and untransmitted video data. When the bike ID number is read, the station will add said ID to the list of bikes currently stored. When a bike is removed from a dock, the station will remove said ID from the list of bikes currently stored and set the "last locally updated" time in the bike health data storage.

Every 15 minutes, the CCF will send out the updated *available_bikes* and *total_docks* hashmaps (not encrypted) and a map of user emails to the variable *user_bike* (encrypted). Upon receiving this information, a station will decrypt the user map and add all information to a cache.

The station will have a kiosk to provide bike borrowing and reservation access to users. The kiosk will show users which docks have a bike connected, a summary of health for each bike, and the charging status of any e-bikes. When members attempt to reserve a bike at another station, the station first checks its cached information to see if the other station has bikes available. If the cached information is less than 15 minutes old, the user is shown the availability map using the cached information. When they select a station to make a reservation, a request is sent to that station to confirm the current availability. If the requested dock/station is still available, the reservation is confirmed; otherwise, the updated information is added to the cache and the user is prompted to make another reservation. This enables us to strike a balance between keeping the reservation information up to date and limiting the frequency with which we have to update its information.

For *accessibility*, the kiosks will use a high-contrast color scheme to allow those with minor vision impairments to access bikes. The kiosks will also offer an option for text-to-speech to provide better access to those who may have trouble reading the text information on the screen.

Lastly, we address the topic of *superstations*, which was introduced in the DP update. The most unique aspect of superstations is their *mobility*; a superstation's location and availability are subject to change over time which we can leverage to relocate docks and bikes to accommodate for large events, but this does require us to modify our approach for making reservations.

To account for this, by default, the CCF will set the entry for all superstations in *available_bikes* and *total_docks* to 0. When a superstation is set up, it will begin sending "delta packets" (see section 3.2.2) to the CCF which will contain the information listed above, most notably the number of bikes/docks available, and the time when the session is scheduled to end. Upon receiving this information, the CCF will update the corresponding entry in *available_bikes* and *total_docks* for the superstation, and use the GPS information to broadcast the location to users who want to make reservations.

Additionally, the CCF will use the session time sent in the delta packet to determine when users can make reservations. 15 minutes before a session for a superstation is set to end, the CCF will restrict users from making any more requests to reserve a bike. Similarly, for reserving docks, users will be able to select a time from the start of the session, to up to 15 minutes before the session is over.

3.2.2 Bikes

The bikes submodule consists of three types of bikes: basic bikes, standard bikes, and e-bikes. Each bike is equipped with a GPS tracking system. Having three types of bikes supports a variety of customer needs. Firstly, E-bikes provide a more *accessible* alternative for those who are physically impaired or who desire a less physically-straining experience. Given that less than 15% of the population is estimated to experience some form of physical disability[2] 15% of the bikes will be e-bikes. Non-members cannot use basic bikes, so we want to provide as many basic bikes as possible without compromising *accessibility* to non-members. The remainder of the bikes are standard bikes.

Each bike stores the following data, which is needed to maintain and optimize the system:
- Unique ID, in the form of an integer
- GPS location, in the form of a pair of floating point numbers for latitude and longitude
- Time at which the bike was borrowed, in the form of 4 integers

When a rider docks a bike, the CCF updates its record of bike availability and the user's activity. If the user's *angel_ride* bit is set to 1 and the user's ID is contained in the CCF's *angel_stations* hashmap, then the user's *angel_extensions* are incremented by 3. Subsequently, the user is charged based on their membership status and number of extensions used. If the user's *angel_extensions* is nonzero, the maximum number of extensions that can be canceled by the user's angel extensions are subtracted from the cost and *angel_extensions* is decremented by the number of extensions used.

E-bikes and standard bikes are equipped with a camera module, as described in section 3.1.3. To test the cost, reliability, and speed of transferring videos to the computing facility, half of the bikes with cameras will use WiFi while the other half will use cellular data. While using cellular data is cheaper, data transmission rates are slower. Testing both forms of communication will allow us to determine the severity of these risks and commit all of the bikes with camera modules to use either cellular data, WiFi, or a novel mix of the two. The bike will only use WiFi or cellular transfer methods in the case when it is unable to transmit its video data to the station via USB in time before another user rents the bike. Bikes can only store up to 30 minutes of video (the length of a default ride), so the remaining video data will be transmitted during the ride and immediately deleted to make room for video recording during the current ride.

$$Maximum\ video\ data\ =\ \frac{30min}{1\ video} * \frac{60s}{1min} * \frac{45\ MB}{1s} = \frac{81000\ MB}{1\ video}$$

[2] Leppert, Rebecca. 2023. *8 Facts About Americans with Disabilities*. Pew Research Center

$$Cellular\ data\ transfer\ time\ =\ 81000\ MB * \frac{1s}{80\ MB} = 1012.5s\ =\ \sim 17\ min$$

$$WiFi\ data\ transfer\ time\ =\ 81000\ MB\ *\ \frac{1s}{20\ MB\ low,\ 2000\ MB\ high} = \sim 68\ min\ low,\ \sim 1\ min\ high$$

In the case of cellular data transfer, the transfer can take place during the ride without requiring any data to be transferred while the bike is docked. In the worst case of the WiFi data transfer, the transfer will not be able to take place during the ride. The leftover data must be transferred using the USB connection. It will take

$$(68\ min\ -\ 30\ min)\ * \frac{60s}{1\ min} * \frac{20\ MB}{1\ s} * \frac{1\ s}{600\ MB} = 76s$$

to transfer the rest of the data over USB while docked. This is a short time, very reasonable for another user to wait until the data has been transferred.

To ensure continued *accessibility* to bikes, abandoned bikes should be recovered as soon as possible. One condition for a bike to be identified as lost is if it is approximately stationary for a certain amount of time. However, problems could arise with this condition, such as in the case where a bike is left on the metro, causing it to appear as an in-use bike.

To combat this, we instead choose to identify a bike as abandoned if it has not been docked in the last 24 hours. This can be done using the onboard processor, so the bike will be able to lock itself without needing to communicate with the CCF. However, the CCF must be able to locate the abandoned bike to support the Heroes feature. Thus, after a 24-hour ride period:

1. The bike will consider itself abandoned, lock itself, and send its GPS location to the CCF using the on-board LoRa radio system, every 30 minutes. During this time, the bike will switch to "cyclic listening mode." More information about this mode is available below.
2. The CCF considers the bike abandoned and marks the bike as lost in the system.
3. The user is charged the cost of the bike.

Charging the user is justified because *accessibility* is our priority and, as such, it is crucial that we replace lost bikes as soon as possible. Nonetheless, users will be reimbursed the cost of the bike if the bike is ever found and returned. By choosing not to share the bike's GPS location until the bike is considered lost, user *privacy* is prioritized. Abandoned bikes can be unlocked if they receive an "unlock" message through the LoRa radio system from the CCF, as well as in the rare case that a bike is mistakenly identified as abandoned. These users will have an hour after the bike has been unlocked to return it to a dock before the bike locks itself again.

In "cyclic listening mode," the bike turns the LoRa radio system into listening mode for 1 minute every half hour, on the half hour. For example, if the bike is locked at 1:23PM according to the internal clock, it will enter listening mode from 1:30PM to 1:31PM, 2:00PM to 2:01PM, 2:30PM to 2:31PM, etc. This is 1/30th of the power demand compared to the bike turning on listening mode constantly, so the LoRa radio system battery won't need to be replaced as often. After both receiving GPS data from a lost bike and a user requesting to become a Hero for this bike, the CCF will also send out a signal to unlock the bike during this 1 minute period every half hour. It will send 60 signals, one per second, during this 1 minute period. This allows the user to send a request to the CCF to rescue a bike and receive an expected time frame that the bike will be unlocked. The bike will relock itself after 10 minutes of not moving, which can be detected using the onboard GPS.

One worry about using the onboard clock is the accuracy drift over time. According to AMD[3] and the chips they use in their CPUs and other computer products, their RTC (Real-Time Clock) chip is bounded to an error of ±30.5ppm, which is approximately 2.5 seconds of possible drift per day. This means for the 1 minute period the bike is listening to drift outside of the 1 minute period the CCF is sending, the bike would need to be lost for an expected amount of 24 consecutive days. We hope this is extremely unlikely both due to the incentives given to Heroes and due to the requirement for the drift to be in the same direction during the whole time period.

Using cyclic listening mode allows bikes to be maintained less often, increasing the expected number of bikes available to users, which increases *accessibility*.

3.3 Digital Access

3.3.1 Website

When a user creates an account, they must input their email and payment information. A request to create a new entry in the database for *user_email* with *payment_information* is sent to the CCF. Users may update their account at any time. In addition to being able to borrow bikes on the website, users can also reserve bikes, explained more in detail in sections 3.2.1 and 3.1.5, respectively.

3.3.2 Mobile App

The mobile app provides the same capabilities as the website, in addition to a few other features. The app allows members to request possible bike and target station pairs that would qualify the user as Angel, or a list of nearby lost bikes, both of which the CCF stores and provides. The app also allows members to view videos recorded by them and download or request for them to be deleted before the lifetime specified in Section 3.1.3. Lastly, the app supports requesting the location of abandoned bikes to perform a *Hero* operation, as described in section 3.2.2.

3.4 Dealing with Power Outages and Network Failures

Stations are designed with local *accessibility* in mind; they should be able to provide all or nearly all features with limited resources, such as communications to the CCF and continual access to power. To ensure this, we handle power and network outages in the following way.

[3] AMD, 2023, *Versal Adaptive SoC Technical Reference Manual (AM011),* https://docs.amd.com/r/en-US/am011-versal-acap-trm/RTC-Accuracy (accessed 04-18-2024)

3.4.1 Power Outages

When a station loses continual access to power and starts running on the backup battery, it switches into "low-power mode." Some features, such as, sending video data to the CCF, and e-bike charging, are limited or disabled to save power for more important features, such as continued bike access and sending more important information to the CCF. To ensure users in the area have continued access to bikes, the station will send out a radio signal to nearby bikes to pause their lock timers; both the 30 minute and 24 hour timers mentioned in Section 3.2.2 will be paused. This reduces the inconvenience of returning bikes while a power outage is occurring. Because stations can still be used to some extent, this increases *accessibility* to those users in the area affected by the outage. However, when the battery backup is completely drained, the station will not be able to function anymore. All features requiring electronics, including those enabled during low-power mode, will be disabled.

Stations affected by an outage will not allow bikes to be borrowed when less than 30 minutes of battery remains. Users returning bikes are not affected by this, and may return bikes as normal to affected stations. This is a precaution to ensure there are enough bikes at the station when the outage ends to return to normal operation.

Every 10 minutes, the station will send a "delta packet" to the CCF with any changes made to the values it keeps track of since the last successfully sent packet. By only sending changes, the CCF still receives the same information while requiring less actual information to be sent. To ensure these "delta packets" have been received and processed by the CCF correctly, we will use the TCP confirmation and retransmission protocols.

In case of a station system failure, the station will re-request information regarding bike IDs and health from all bikes connected via USB to regenerate the information lost in memory.

3.4.2 Network Failures

When a station loses continual access to the network, it switches into "no network access mode." Some operations (sending delta packets) are postponed until network connection is reestablished. Because the station does not have access to the network, some operations on other stations or the CCF will not work.

- Reservations cannot be made to or from stations without network access.
- The CCF will not have updated information about stations without network access.
- Angel algorithm information may be outdated depending on the length of disconnect.
- Payment information cannot be confirmed, so bike rentals will be restricted to members.

Bike rentals for members will be made possible by the cached data described in section 3.2.1. Members will be able to type in their email address at a station and if the user is listed as not having a bike rental in the cache, they will be able to rent a bike. The cache will then be updated with the timestamp at which the bike was rented out. Once the network is back on, the station will check the messages from the CCF regarding (1) user data, to ensure that any user who rented a bike during the network failure is not listed as having rented another bike, and (2) the *available_bikes* hashmap, to check if the bike has been returned within the 45 minute window. If the stations find that the user has rented additional bikes, or has not returned their bike on time, it will notify the CCF to charge the user accordingly.

Despite the loss of some features during network failures, there are some operations that are not affected, namely:

- Bikes can still be returned
- Ebikes can still be charged
- Bikes can still be borrowed by members

This means some *accessibility* is still retained, as bike rentals can still take place for the most part. During longer outages, the same operations are postponed or disabled until network connection is reestablished.

4: Evaluation

In this section, we analyze the communication overhead and expected runtime and costs of certain operations. To evaluate the data storage and operation costs of these modules, we must first analyze the number of bikes and stations that we require in order to sustain the expected amount of traffic.

4.1 System Size

Based on the statistics of Boston's bikeshare system, Bluebikes[4], we can estimate the number of stations and bikes needed. Since distinguishing between the percentage of rides that come from members and non-members is irrelevant, we scale by the ratio of the expected number of rides as follows:

$$(2M\ expected\ rides/year \div 3.7M\ Bluebikes\ rides/year) \cdot 5,300\ Bluebikes\ bikes\ =\ 28,648\ bikes$$

28,648 bikes is an overestimate of the number of bikes needed, as scaling by the ratio between members of the two systems yields a much lower result. However, since we prioritize accessibility, we scale the system in this way. In order to have an average of 15 bikes per station, we choose to have 1,909 stations.

Since the population density is not uniform, the number of docks at each station will vary. Under the assumption that bike-riding frequency across the city is approximately uniform, we choose the number of docks at each station to be directly proportional to the population density of the respective area. This is done to ensure accessibility for areas with high demand. Paired with the angel tool, this consistently ensures an appropriate distribution of bikes between stations.

[4] Bluebikes, 2023, *System Data*
https://bluebikes.com/system-data (accessed 05-06-2024)

4.2 Storage evaluation

4.2.1 CCF

As mentioned in section 3.1.1, the user login information takes up approximately 14.4 MB of storage in total for all 50,000 expected members. The remaining pieces of data attached to each user take up a total of 48 MB of storage for all members, as calculated in section 3.1.1, and the CCF's *available_bikes* and *total_docks* hashmaps take up about 2 MB of storage. All together, this is well under the 100 TB of storage available even after encrypting the data. The excess storage allows for our system to scale easily, as the ability to expand is a priority in order to ensure accessibility to bike riding as demand increases.

As mentioned in section 3.1.3, the remaining storage allows for approximately 600 hours of uncompressed video footage to be stored at once. Since the expected number of rides per year is 2,000,000, the average number of rides per day is approximately 5,400. Only 10% of standard and E-bikes have cameras, which means that, in the worst case, 540 user videos are expected to be stored at once. This is well under the 800 users that can be accommodated at once since we can store approximately 600 hours and each user requires at most 45 minutes of video data.

4.2.2 Stations

The storage usage of each station is dominated by the count of reservations made by each user, which is limited to 4 per user every day. The rest of the data stored in each station is negligible in size compared to this data component, so we calculate an upper bound for the amount of storage allocated to user reservation count as follows:

$$(2048\ bits/user\ email\ +\ 2\ bits\ for\ num\_reservations) \cdot 50,000\ members\ <\ 1.3\ GB$$

Taking the rest of the data into account, this is well under the terabyte of storage that each station has.

4.3 Performance evaluation

To show that performance of our system is not hindered by any bottlenecks, we first look at the number of connections that the CCF can handle simultaneously. Since the number of expected rides per year is 2,000,000, the expected number of rides per day is 5479. The number of rides per day is not uniform throughout the year, so a conservative upper bound on the number of rides in a single day is 15,000. Suppose that, out of these rides, half of them are concentrated in a span of 4 hours during the day. This equates to an average of 30 rides per minute during this timeframe, or 30 simultaneous CCF connections in the worst case, in other words. The CCF can have up to 5000 simultaneous connections, which leaves a plethora of open connections for other user and station requests, such as reservations and video transmission requests sent to the CCF. This ample supply of open connections enables our system to scale up easily.

Now, we analyze the average case and worst case latency of several requests. When connected to a dock, videos are transmitted at a rate of 600MB/s. Since each video is approximately 81,000MB, this comes out to be a rate of 2 minutes and 15 seconds per video. As per section 3.2.2, if a user rents a bike during this time, the remaining video data is transmitted via cellular or WiFi. However, most bikes will be docked for at least 2 minutes in expectation. Therefore, despite low transmission rates with WiFi and cellular, the average video transmission time is very low due to the majority of transmissions occurring via USB. Due to the excess number of connections available in expectation, this is an upper bound on the video transmission rate, as each video being transmitted should likely have its own connection. All other packets of information, such as reservation requests, take less than a second to send and receive a response since all packets are significantly smaller than 1 Tb and the network capacity is 1 Tbps.

5: Conclusion

The AIR bike-sharing system delivers an *accessible* and *privacy-oriented* user experience for the people of Newplace. We prioritize *accessibility* through various design choices in our system, ranging from how we choose to deal with power outages to the kinds of bikes that users will have access to. Additionally, throughout every phase of our development process, we have carefully restricted the collection of data to minimize intrusion and ensured that its utilization never compromises user *privacy*. All in all, we address the need for users to easily access and use our bikes through a robust system designed to resist network failures, power outages, and large amounts of simultaneous network traffic.

6: Author Contributions

The authors of this project collaborated closely throughout the entire design process—from conceptualization down to the granular design decisions for each component. For the preliminary report, we divided the project into Station, CCF, and Bike modules, which were primarily authored by Jonatan, Jesus, and Fabian, respectively. For this final report, we collectively addressed all the feedback we received over the past few weeks and did a final editing pass.

7: Acknowledgements

We would like to thank Mohammad Alizadeh for his invaluable and thoughtful technical feedback. Additionally, we would also like to thank Kate Parsons for giving us valuable feedback and teaching us ways in which we can better communicate our design. Lastly, we would like to thank Grace Song, Katrina LaCurts, and the rest of the 6.1800 staff.