SUPER: Seamless Urban Pedaling – Efficient and Reliable

6.1800 Design Project Report

By Abhay Basireddy, Francisco Colon, Tae Wook Kim

WRAP Instructor: Emily Robinson
Recitation Instructor: Larry Rudolph

May 6th, 2024

1. Introduction

In urban environments worldwide, the demand for sustainable and efficient transportation solutions has intensified, prompting the development of innovative bike-share systems. However, existing systems often grapple with inefficiencies stemming from disjointed information flow between system modules, resulting in inconvenience for both users and the city. In response to this challenge, we introduce SUPER: Seamless Urban Pedaling: Efficient and Reliable, a pioneering bike share system designed to revolutionize urban mobility.

SUPER addresses the shortcomings of traditional bike share systems by seamlessly integrating information flow across its modules, optimizing user experience and system efficiency. At the heart of SUPER's design philosophy are two primary principles: reliability and availability. Reliability, in the context of SUPER, entails the system's ability to consistently deliver a dependable biking experience to users, free from unexpected failures or disruptions. Meanwhile, availability signifies the system's commitment to ensuring that accessing a bike requires minimal effort for users at any given time. SUPER’s main objective is to achieve these two properties for commuters, who are our target customers. The system puts commuters at the center of our customer base considering their frequent use and significant portion of the revenue stream from their memberships.

This paper presents a comprehensive exploration of SUPER, organized into several key sections. First, we provide an overview of SUPER's overarching architecture and operational framework. Subsequently, we delve into the details of SUPER's software modules, demonstrating how each module contributes to realizing the system's overarching goals of availability and reliability. Following this, we examine various use cases to illustrate the real-world applicability and impact of SUPER in diverse urban contexts. Finally, we conclude with a discussion on the future implications of SUPER and potential avenues for further development and refinement.

2. System Overview

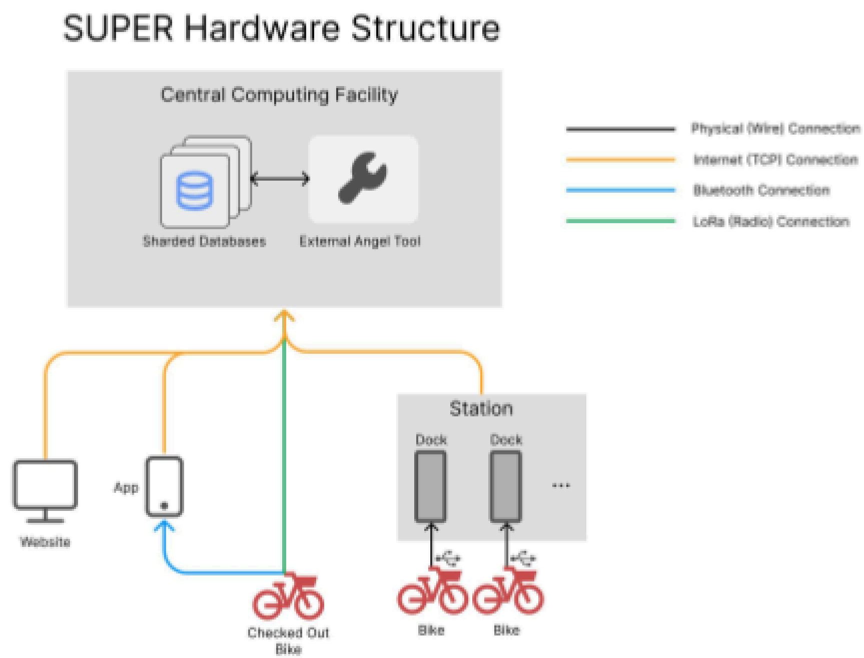

Figure 1: The hardware infrastructure and connections of SUPER

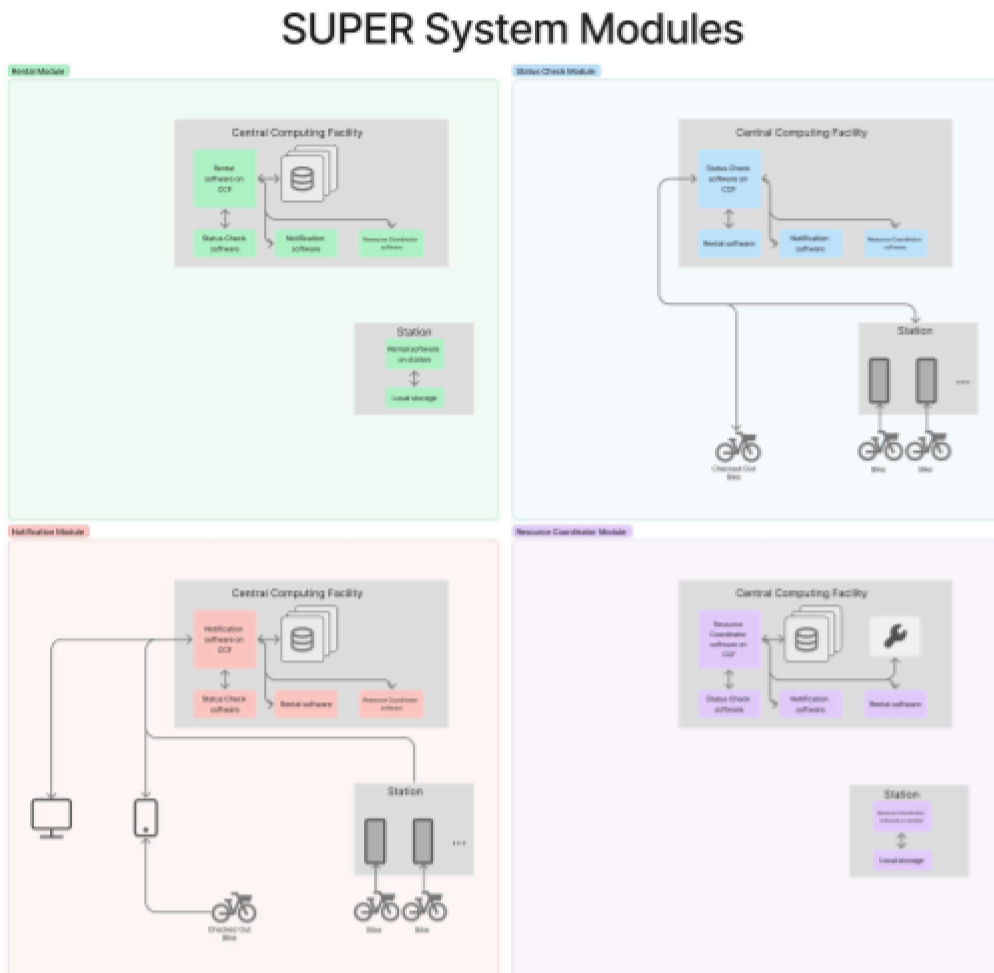

Figure 2: Modules of SUPER and their Corresponding Hardware Components

SUPER aims to provide the required bike-share functionality while prioritizing the reliability of the system for users and the availability of bikes. It does so through its overarching modules: the rental module, status check module, notification module, and resource coordinator module. Figure 1 displays the hardware across the SUPER system and the various modes of communication between them. Figure 2 displays the hardware used by each module separately and the components each respective software module communicates with. Figure 2 omits the specific modes of communication for clarity, but references the same connections as Figure 1.

- *Rental Module*: The rental module manages all bike rentals and reservations for the system, incorporating the angel and hero functionality as special cases of rentals (different users checking in and out, for example). The majority of the functionality is in the central computing facility (CCF), but we prioritize the reliability of the system by allowing stations to take over some rental/return functionality during communication outages. This module interacts with the sharded database to speed up data access times, which allows for reliable growth of the number of users without overloading the database with requests.

- *Status Check Module*: This module enforces the reliability of the system by consistently checking that each of the hardware components is functioning properly. It determines if bikes are abandoned, pings from stations for their data after rides, and maintains regular contact with stations to have the latest information about the bikes and docks available. This makes it easier to determine communication outages and start the relevant worst-case functionality as well as expediting new reservations, promoting the reliability and availability of the system. This module uses encryption and TCP to maintain confidentiality and integrity. After a communications outage, it also uses a deficit round robin to reconcile data between the CCF and the disconnected station.

- *Notification Module*: The notification module is the core of communication between the different hardware components of the system, sending one-directional information between components (as opposed to the bidirectional communication in the status check module). This module aids in data transfer between bikes and stations and between stations and the CCF. It is also the only module communicating with users through the app and website, processing requests internally in the system and responding with the results. The module bolsters the availability of the system by sending live location data from bikes to the CCF via users' phones so the resource coordinator module can activate the angel tool if necessary. The module uses TCP connections to maintain data reliability and certificates to connect with the app/website to maintain data confidentiality.

- *Resource Coordinator Module*: In general, this module ensures the resources are properly allocated throughout the system and user requests do not overlap. This includes making sure reservations are only made when bikes and docks are available, heroes are appropriately considered and notified to rescue bikes, and angels are appropriately contacted to ride the bikes. In particular, this module implements a queue system to manage the possible heroes per bike and angels per station, and sets a short time limit to ensure enough users get the chance to be heroes and the bikes are successfully moved/rescued quickly, promoting the overall availability of bikes in the system.

3. System Design

The SUPER consists of four core software modules that collaborate to make the system reliable and available: rental, health check, notification, and resource coordinator. This abstraction creates modularity for scalability and ease of maintenance.

3.1 Rental Module

The rental module is the core module of the SUPER system that provides the main functionality of the system. In a general sense, this module handles renting out and returning the product which includes managing stock, corresponding location with available stock, tracking rental durations, payment, mapping members to the product, and recording histories. This can be directly applied to our bike-sharing system to handle bike rental and returns. The following subsections will talk about the details of this software module including states and transfer of information.

3.1.1 Data Storage

The two tables that will be used in this software module are *Trip* and *Station_Availability*. Both of these data are primarily stored in the CCF but temporarily stored locally in each station until it gets an acknowledgment that the CCF has received the data.

Attribute	Size	Description
Renting User	16 bytes	UUID of User
Renting Time	8 bytes	Timestamp
Renting Station	16 bytes	UUID of Station
Returning User	16 bytes	UUID of User
Returning Time	8 bytes	Timestamp
Returning Location	16 bytes	UUID of Station
Payment Status	1 byte	enum {paid, not paid, n/a}
Bike Type	1 byte	enum {basic, standard, ebike}

Universally Unique Identifiers with 16 bytes will be used for users and stations to allow for scalability and flexibility within the system. As shown in the table above, each record will have a size of ~66kB bytes corresponding to each trip. Assuming there will be 20 million completed rides per year, we estimate:

$20\ million\ \times\ 66kB\ =\ 2\ GB$ of total storage

Because it is an insignificant amount, the redundancy of local data in the station and usage of UUID will increase the reliability of the system without sacrificing a noticeable amount of storage.

To seamlessly prioritize the availability of the system, the *Station_Availability* table will store the current status of the station including health and available bikes with corresponding types.

Attribute	Size
Station Health	1 byte
Number of Basic Bikes	1 byte
Number of Standard Bikes	1 byte
Number of E-Bikes	1 byte
Number of Docks	1 byte
Number of Reserved Docks	1 byte
Number of Reserved Basic Bikes	1 byte
Number of Reserved Standard Bikes	1 byte
Number of Reserved E-bikes	1 byte

In the CCF, we will also use database sharding to partition database records by station ID. Following our assumption that most of the customers are regular commuters who use the same stations daily, the smaller partitions will increase the efficiency of database reads and writes.

3.1.2 Functionality

This generic software module seamlessly handles all the necessary functionalities of the bike system: angel, rescue, reservation, and superstations. In most cases, the same user rents and returns the bike, and the timestamp is recorded synchronously. For rescue service, on the other hand, the renting user will be the customer who rented out the bike, and the returning user will be the hero who rescues the abandoned bike. For angel service, both the renting and returning users will be the angels with pre-set values of returning station and renting time, as well as payment status of ‘n/a’ since the trip will not cost the angels. Similarly, reservation functionality for members will simply entail the pre-set values of renting or returning time.

By using UUID as a station ID, we can safely add or remove superstations on demand without modifying data about existing stations or concerns about conflicts of IDs. With the new station ID, the module will handle the superstation just like any other station.

3.1.3 Data Transfer

This Rental module will utilize the Status Check module to communicate between bikes, stations, and CCF. As mentioned, there will be temporarily two copies of the data while the locally stored data on the station is being transferred to the CCF. In order to keep accuracy and consistency, a heartbeat of 5 minutes is used to update the data on CCF and station. The 5 minutes of heartbeat is a frequent communication on the network especially to a centralized node of CCF, but to achieve availability which is our second most important property, the heartbeat is kept as frequent as possible without congesting the CCF.

3.1.4 Computation

The primary computation for this software module will be calculating trip price using member tier, bike type, trip duration, and video usage. Although this calculation will be frequent, this simple arithmetic computation will not affect the performance of the system. Another computation is managing different stations such as adding or removing stations, which requires updating nearby stations to efficiently activate angel and rescue service and rerouting users in case of unavailability of a station. This computation is algorithmically heavy considering it can be up to $O(n^2)$ where n is the number of stations. However, the system is not expecting to add or remove stations as often, even with the superstations.

3.1.5 Worst Case

In case of power or network failure, data might not be able to synchronize between different components. By allowing asynchronous updates and redundancy on the local station, the system will keep the eventual consistency. Since most of the data, except for the availability, is not time-sensitive, this eventual consistency will not significantly impact the system as a whole.

Considering the significant size of the videos, there might be cases where local storage at a station is full during the network outage due to a lack of communication with the CCF. In this scenario, we will disable the station from renting out more bikes to avoid freeing up docks for more bikes. We are sacrificing the availability of a station for reliability since users who pay additional charges for the video service should be able to rely on the system to get their videos. We expect communication outages to be fairly rare, and only 10% of the bikes are video bikes, so we do not suspect this will occur often.

In another case where the payment system is not functioning because of network failure, the system will only allow members to rent the bike and process the payment with stored credit card information on the member’s account asynchronously. The impact of this will fall mostly on non-member customers' availability, but given our priority is reliability for members/commuters, this tradeoff will be consistent with this design principle for our intended users. However, the existing reservations that were already stored in the station’s local storage will continue to be valid and operate as expected.

3.2 Status Check

With reliability as the first priority of our system, making sure all the components of our system are functioning is crucial. SUPER achieves this utilizing a status check software module that pings different components and receives responses about the component’s health and additional data about the status.

3.2.1 States

This software module is for bi-directional communication, and therefore, it does not store any data in the module but requires states to enable communication seamlessly. The states include sender, receiver, and data. The sender and receiver are the two components that are communicating and data is a flexible format that can range from text representing health to video buffers. This flexibility of data allows various modules within the system to utilize this Status Check software module for different purposes.

3.2.2 Functionality

This module partakes in multiple functionalities required by the system. This includes communication between CCF and station, station and bike, and CCF and bike.

The communication between CCF and stations is crucial to managing the latest status of stations for reliability and availability. As described in the rental module, CCF pings the station for a regular status check every 5 minutes and the station responds with the status of the station including available bikes, health, etc. This allows the system to detect station failure or congestion quickly to respond correspondingly. Additionally, whenever new reservations are made through the mobile app or website, CCF will ping the stations to fetch the most updated status to guarantee no conflict or over-booking.

The CCF also communicated directly with bikes using LoRa radio in case of abandonment. The CCF pings the bike every 30 minutes and the bike will respond with the GPS data to activate rescue service. The pre-set interval of 30 minutes was to prioritize reliability over availability. Even though checking every 30 minutes can slow down the rescue process with not as frequent, accurate GPS data, we prioritized the battery life of the LoRa radio on the bike to guarantee eventual rescue of the bike. Given the assumption that the abandoned bike will be locked and not moving around, an infrequent GPS update was a reasonable tradeoff.

Lastly, the communication between the station and the bike occurs after each trip. The station pings the bike through a USB cable on the dock for the health of the bike and trip information, including GPS and personal videos, which will be responded to by the bike. While the communication is happening, the bike will be unable to be rented out by the next customer. In the worst case, it can take up to:

$$45\, MB/s \times 60\, s/min \times 30\, min \times 2\, lenses = 160\, GB$$

$$160\, GB \div 600 MB/s = 266.67\, seconds$$

The potential 4 minutes of wait time at the station sacrifices availability, but it will guarantee the successful transfer of video and enough storage in the camera for the next customer, prioritizing reliability.

3.2.3 Security

Protecting customers' privacy and storing accurate data are expected by the customers and the security is directly related to the reliability of the system. Consequently, the module implements secure and reliable TCP with public key encryption over UDP. This will ensure both the confidentiality and integrity of the data over the network. Additionally, all the components that communicate will always communicate to the same components, so exchanging symmetric keys through computationally heavy private key encryption will be infrequent, making the overhead of the tradeoff not as significant.

3.2.4 Worst Case

In the case of a communications failure, the status check module will recognize the disconnect between a station and the CCF. The station will support limited functionality detailed in section 3.1.5 and act independently during the outage. Once the station comes back online, it notifies the status check module about its reconnection, at which point this module will use a deficit round robin scheduling algorithm to reconcile the station data with the CCF data while accepting requests and data from other stations and users. The proportion of quantum between the disconnected station and the rest of the network is 1:1 so the CCF can update its internal data more quickly, prioritizing availability since the rental module and resource coordinator modules can allow users to make reservations from this station more quickly after it comes back online.

3.3 Notification Module

The notification module of SUPER covers all the one-directional communication between the components of the system. This mainly covers the data transfer between bikes and stations, stations and the CCF, the CCF and the user interfaces (app and website), and bikes to the mobile app/the app to the CCF if possible during rides. The notification differs from the status check module primarily in that the information transmitted does not need a response back from the receiver. We also choose to immediately transmit this information as soon as possible to keep the most up to date data in the CCF for use in other modules that depend on real-time availability information (as opposed to the status checks in regular intervals).

3.3.1 Data Transfer + Overview

The notification module covers sending bike availability and ride data from the bikes to the CCF when bikes are rented/returned, interacting with users through the app and website to display the most up-to-date information, and sending live location data from the bikes to the CCF through the mobile app for users who keep the app online during rides. Since our system prioritizes availability and this information is used to determine the availability of bikes, the data is sent immediately after receiving/processing it so users can make reservations without delay and reliably know if there are bikes or docks open at the moment they open the app and website.

Due to the one-directional nature of this transmission, we choose to send all data over the network using TCP so it reliably reaches the receiver. There is a tradeoff between the speed of the transmission and the reliability of the transmission, and we choose to prioritize reliability here since the accuracy of our data in the CCF is more important than the speed it takes to get there as users cannot rely on the system if there are inconsistencies when they make a reservation and cannot access a bike/dock and a quick, but corrupted transmission may entail further transmissions to confirm the validity of the data anyways, leading to more time overall.

3.3.2 The functionality required by the spec

The notification module is the main method of communication between users and the system since it manages the flow of information to the mobile app and website. Whenever users try to check station availability, make a reservation, access their videos, get angel or hero information, or check routes, all of the information they request will be delivered to them using the notification module. When users make reservations or act as angels/heroes, the corresponding modules will process this behavior, but the notification module will send the results of this processing to the user so they can act appropriately. This module will ensure the users can access their ride data and videos as well after the ride, as detailed in section 3.3.3.

This module also supports the accuracy and efficiency of the reservation system. When a bike is docked after a ride, the bike sends a notification to the station that it will start uploading data, which in turn notifies the CCF that a bike is uploading data and there is a dock with a bike uploading data. This is important since the data transfer may take some time, so the system should propagate this information to the CCF. Once the bike finishes uploading data to the station, the station sends this data to the CCF (at faster speeds since it is not restricted to the slower USB connection), and there is a new dock on the station with an available bike. Stations similarly send the most recent bike availability information to the CCF as soon as bikes are rented and removed, ensuring the CCF has the latest accurate information to use in the resource coordinator module so that it can quickly and properly respond to users making new reservation requests.

Finally, this module facilitates bike availability by managing the flow of location data from bikes to the live angel tool in the CCF via users' mobile apps. For users who do not keep their apps online during rides, this module simply does not activate. Otherwise, this module first sends one GPS reading per second to the mobile app via Bluetooth (about 1/10th of the max frequency of the GPS sensor). The app transmits this data across the Internet using TCP to the CCF, which accepts it in the resource coordinator module, which appropriately deals with the data and can activate the angel tool if it detects a high enough demand in a certain area.

3.3.3 Storage

The notification module exclusively manages users accessing their ride videos from the app since it covers the app-system interface. The CCF must store user videos and accident videos, which can be up to 160 GB (45 MB/s * 60 s/min * 30 min * 2 lenses), so this has the potential to grow large quickly. To prevent too much storage use, personal videos will be available for users to download for 24 hours and then removed from the system. Accident videos will be saved on the system automatically and can only be manually deleted if Bikes4All deems it appropriate.

Once deleted, the notification system will only send the user's bike data for the ride to the app when the user views their previous rides.

3.3.5 Security

Similar to section 3.2.3, maintaining confidentiality and integrity of the data and requests sent by users is important in ensuring our system is reliable and only provides the functionality/transactions requested. To achieve this, SUPER uses an HTTPS connection and a certificate from a certificate authority for any communication with users through the application and website. For internal communication between components of the system, SUPER uses the encryption scheme described in section 3.2.3.

3.3.6 Worst Case

Since the notification module uses acknowledgments and checksums with TCP, the notifications should reliably reach their destinations under normal network conditions. In the case of a communications outage near a station, the station will collect the updates to send to the CCF for when it comes back online again and maintain its local storage of the status and availability of the docks, reservations, and bikes. This local storage should suffice for the limited functionality during network failure detailed in section 3.1.5. Once the network comes online again, the station sends a notification to the status check module about its reconnection to the network, at which point it will reconcile the station data with the CCF data internally.

3.4 Resource Coordinator Module

The resource coordination module is deployed to prevent overlapping requests by users. Its purpose is to ensure that different users are not simultaneously assigned the same action, such as reserving or rescuing the same bike. This module maintains cohesion between different user actions and eliminates potential conflicts.

3.4.1 Data Transfer + Overview

This module will utilize the information currently used on the CCF and will communicate with different hardware components and users using the Notification Module and the Status Check module. The module will integrate two queue systems: one for the Angel Tool and one for the Hero Tool, and it will establish a protocol for reservations that prevents inconsistencies between users.

3.4.2 Functionality

3.4.2.1 Reservation

When a user wants to make a reservation, the CCF will obtain the latest information from the station with the Notification Module and it will present the user with the amount of available bikes and available docks. If there are any available bikes or docks, the user will be allowed to make a reservation. Once the user decides to make the reservation, we will use the Status Check Module to get the latest information as there is a chance that the remaining bikes or docks were claimed by other users in between the time spent of the user checking for a reservation or making it. If the bikes or docks were not claimed we will enact the reservation, otherwise we will send the user an error message.

3.4.2.2 Hero

In order to prevent conflicts we will be using a queueing system for the Hero Tool. Every abandoned bike will have a queue associated with it. When utilizing the live location feature, users within 1000 an feet radius of an abandoned bike are added to a queue. The first user in the queue for the abandoned bike will receive a pop-up prompt whether they wish to be designated as a hero. If the user decides to decline, waits for 30 seconds or leaves the 1000 ft radius they are removed from the queue and the next user on it will receive the prompt. This queue ensures that no two users will be assigned to rescue the same bike at the same time. Once a user decides to become a hero, the two nearest available docks will be reserved for them.

3.4.2.3 Angel

The angel system will also work with a queue. A queue will be created for a station once the angel tool is activated for that station. Potential angels for that station will be added to the queue. Users who are at the top of the queue will be prompted to become angels as long as there is available space on the stations. If a user waits for too long or declines to be an angel, the following user in the queue will receive the prompt. Once a user becomes an angel, we will enact a reservation for a bike and a dock at the specified stations.

3.4.3 Storage

Queues for the Hero and Angel tool will be stored on the CCF. A queue made for the Hero tool will last as long as the bike to be rescued remains abandoned. A queue made for the Angel tool will only last as long as there are angels needed for the specified stations. We evaluated the choice of making the queue stored on the affected stations but that can bring about issues, especially in communications outages since stations will have to communicate through the CCF which will not be possible. Information regarding reservations will be kept on both the station holding the reservation and the CCF.

3.4.4 Worst Case

In cases where two users attempt to reserve the last bike or dock at a station, the user that had the first successful reservation will keep it. The other will receive an error message. This ensures that no two users get a success message for their reservation when in reality one has overridden the other.

In the case of a communications outage, making reservations is not possible, however, reservations will be kept locally on the stations. This will allow users, angels, and heroes to still use a reserved bike or dock if the reservation was made before the outage occurred. If a communications outage occurs on a station that is using the angel tool, all current reservations will still be active, however, the angel tool will deactivate for following users that have not decided to become angels.

4. Evaluation

SUPER aims to provide a reliable and available bike-share system for the citizens of Newplace, so it is appropriate to evaluate it in this context. This section will cover some validation metrics to test the efficacy of the system in these design principles and explain design choices when looking at the system at a higher level.

SUPER aims to provide quick, accurate information throughout the system to allow users to make reservations/request real-time information from the system, but there is some communication overhead. We set the LoRA radio to awaken and respond to the status check module every 30 minutes in order to enforce reliability. Although this slows down recovery if a bike is abandoned, it preserves battery life and ensures the bike is able to be located for a longer period of time and is eventually rescued, so we sacrifice temporary availability for long-term reliability.

Additionally, there is some delay and overhead in transferring the ride data to the CCF. We chose to have the bike send its live location to users' phones using Bluetooth if the app is open and active, and have the app send it to the CCF to improve availability since this information helps with traffic estimation and can be used by the city to improve its roads as well as the angel tool to update its model/start rerouting bikes if it detects a peak in demand. Regardless of whether the user has the app open, we choose to send the ride data through the USB at the end of the ride to the station, which sends it to the CCF. Altogether, this means the ride data transmission is slower since it's limited to the USB capacity (4 minutes as calculated in section 3.2.2) and we do not take advantage of users already sending data from the app to the CCF. However, we made this choice to keep costs lower for users by using Bluetooth, since we expect the majority of users to be commuters who prefer lower costs overall but have a few minutes of flexibility when waiting for a bike. We transmit the bike's location data instead of the user's phone's location data to ensure it is consistent with our system since the accuracy of the phone data may vary by phone. Finally, not all users may have phones as the city aims for the system to be accessible to all, so this ensures the data is transmitted regardless of an app and in a lower-cost manner.

We make different design choices throughout the system to prioritize availability and reliability, so to evaluate the effectiveness of these choices, the length of different system operations should be considered. In general, since SUPER sends ride data through USB and this may cause some delay, we should consider the average wait time at stations to check out a bike at the kiosk as well as reserve a bike online. These also serve as general markers of the availability of the system for users and can be used as a baseline if we need to vary the specifics of other design choices.

We also made the choice to use a sharded database in our CCF based on station to quickly access user data assuming commuters mainly use the same stations. This would also increase the reliability of the system as the number of users go up as we have decentralized tables. To confirm this assumption, we will record the average user data access time using the sharded database as the number of users increases and observe the trend to ensure this time does not go up too quickly/linearly.

We chose 2 hours as the limit after which to declare a bike abandoned to ensure bikes stay relatively close to the city and are able to be reliably recovered. To evaluate this choice as well as the choice to turn on the LoRA radio every 30 minutes, we should record the average bike recovery time for abandoned bikes. Depending on this value, we can change these time constants if necessary to decrease the recovery time. Similarly, we should record the average delay in making reservations and checking out a bike during normal traffic and right after a disconnected station comes back online to evaluate the 1:1 quantum proportion we set for the deficit round-robin algorithm to reconcile station data after a communications outage (detailed in section 3.2.4). We should ensure that the focus on reconciling this data does not lead to significant delays for users in the app, website, and at stations and we can adjust this proportion accordingly to find the optimal value to balance accurate information for the station and minimal delays.

Apart from the metrics above to measure the availability and reliability of the system, we can also consider the worst-case behavior of the system and record the number of failed reservations (due to a communications/power outage) to measure the availability of our system. We should also compare the rate of bikes checked out and docked at stations during communications outages and compare this to the rate during normal communication to check the reliability of our system through failure and ensure users can still use the system.

The main part of the SUPER limiting scale is the video transfer protocol. Since we choose to store videos locally during communication outages and send them through USB to the stations, this takes up space on the station storage and increases the wait time to access a bike. We did not prioritize this functionality since we don’t expect commuters to use this functionality, who are our target audience, so we used Bluetooth on the bikes to lower costs and chose USB transfer instead. We also assumed communications outages and accidents would be relatively rare and took into account that cameras were on 10% of bikes, so we did not prioritize this functionality since the probability of too many videos at a station during a communications outage would be decently low. This may change if more cameras are added and more accidents occur. We can calculate the percentage of rides on camera bikes where users download their videos to check our assumption about commuters, record the average video size to determine if enough videos would fit in local storage during a failure, and record the number of accidents/outages to confirm our other assumptions. As described above, we can check the scalability of the sharded database structure by observing the average delay in making a reservation or checking out a bike as the number of users increases.

We specifically designed the system for commuters in Newplace, as they make up the populace and would use our system more than recreational or tourist riders. As discussed above, our choice of using Bluetooth on bikes and sending data through USB might make the wait longer for a bike but ensure the costs are low, which we suspect is important to commuters as they pay for a longer-term subscription. We also assume commuters have a few minutes of flexibility on their commute, so they can handle such a delay, or make a reservation if necessary. We also do not prioritize the video capability of the system since we assume commuters will not use this often, as they repeat their commute daily. This led to our choices above such as deleting videos after 24 hours and taking some time to reach users (as it goes through the station to the CCF), since it is not as important to us. We will confirm this assumption by tracking the percentage of camera bike rides where the video is downloaded.

In particular, since we assume commuters do not often use the camera, we do not expect there to be many citizens captured on personal videos from the camera. Since there are laws allowing recordings in public places, and many bikers use GoPros and action cameras to record their rides privately, we do not foresee any major differences in the camera functionality from the existing norm of recording on rides. These videos follow the same privacy laws as any other content if uploaded online, and we choose not to review and blur out parts of these videos to save resources and compute power for our modules that increase availability like the resource coordinator module and the angel tool.

We collect lots of ride data, so we expect Newplace officials to have an abundance to choose from when considering policies and initiatives to support their populace. Specifically, the ride location data is stored and helps update the angel tool model, so officials can have a better idea of the different high-traffic areas and times during the week. We also record data about the users like how many rides they have to classify them as commuters, tourists, or recreational riders, which the city can use for logistics. We similarly can offer the city high-level statistics by stations to identify regions that have higher demand for transportation, so they can act accordingly.

5. Use Cases

SUPER's reliable and accessible implementation of the bike-share system provides seamless experiences for several use cases:

- Commuters: Sharding will allow for quicker access to a user's information in areas they most frequent. This will improve reliability and will help commuters have more seamless trips.
- External Research: By using frequency and their deviation to most efficient routes we can classify routes as for commute, errands or recreation. Most frequent and efficient will be labeled as commuting, less frequent and less efficient will be labeled as recreational,

and those in between as errands. These can be used by the government and other external researchers to better improve road management and recreational facilities.

- Tourists and recreational riders: Through the recording of rides, we have the opportunity to introduce features that enhance the overall recreational riding experience. These will include an option for members and non-members to start a recreational ride, which will have a premade route based on average recreational rides between both stations.
- Underserved communities: In order to maintain fairness and uniformity throughout all stations, information such as the number of bikes, their types, and times of congestion will be used by the Angel algorithm. Using bike type as one of the parameters will ensure that everyone has fair access to all types of bikes.

6. Conclusion

The SUPER bike share system gives its users a reliable and accessible system via the implementation of our modules. The rental module enables users to reliably rent and return bikes. The notification module guarantees timely updates to our data, ensuring transparency and trust. The status check module continuously monitors system health, proactively addressing any issues to maintain seamless operations. Lastly, our resource manager ensures users are able to reserve docks and bikes as desired.

Moreover, we are committed to reliable performance despite communications and power outages. Our enhanced protocols ensure that users can continue enjoying the convenience of our bikeshare system for extended durations during these disruptions. Our enhancements to data transmission and availability in the angel and superstation systems will ensure that all users receive fair treatment and consistent access during peak hours. The use of database sharding and splitting into partitions allows for increased speeds when fetching user data and can make the experience for those who rely on the system the most even more convenient.

Features that can enhance the user experience are also planned, such as giving users using Bluetooth the ability to show up on a map with personalized icons (similar to Waze), and a friend/party system for group rides. SUPER's robust protocols, including enhanced availability measures and innovative features, ensure a reliable and accessible bike-share system, allowing users to seamlessly navigate Newplace even during disruptions

6. Author Contributions

All members of the team worked together in designing and creating components for the design. Abhay wrote the System Overview, Notification Module, and Evaluation section. Terry wrote the Introduction, Rental, and Status Check Modules. Francisco wrote the Resource Management Module and the conclusion.

7. Acknowledgments

We would like to give thanks to Larry Rudolph for his meticulous and incredibly helpful feedback on our project, Emily Robinson for giving us encouragement and valuable feedback regarding our presentation and writing, Phoebe Shi for her valuable input as a TA of the class and Katrina LaCurts for her very informing and enthusiastic lectures. We extend our sincere gratitude for their invaluable assistance and unwavering support throughout the project.