# 6.1800 Design Project, 2024: An Enhanced Bike Share System

V1.0; See also DP FAQ and DP Errata

There are five deliverables for this design project:

1) **DP Prep (DPP):** In order to help you prepare with your team design effort, this assignment will require some guided analysis of the DP specification below. This assignment will be written by each student individually, will be approximately 750 words, and is due at **11:59pm on March 1, 2024**.
2) **DP Preliminary Report (DPPR):** This preliminary report will lay out your key design decisions, including both a functional system design and a sketch of any data structures, storage management, and/or network protocols required to achieve your design. It will not include any significant evaluation. It will be written your team as a whole, will be approximately 2,750 words, and is due at **11:59pm on March 22, 2024**.
3) **DP Presentation:** This presentation will address the feedback received on the DPPR, and any corrections or updates to the design project specification. It will also outline evaluation criteria and use cases you will use later for evaluating your design. All team members must participate in the presentation. It will be delivered live and in-person with your recitation instructor. It will occur during the week of **April 16 - 23, 2024**.
4) **DP Report (DPR):** This will be your full report. It will include your final design, all diagrams appropriate for that, your evaluation of your design and a review of how effectively your design addresses the specified use cases. It will be written by your team as whole, will be approximately 6,000 words and is due at **11:59pm on May 6, 2024**.
5) **Peer Review:** In Tutorial your team will have done an early "review," providing informal feedback to another team on their design. For this peer review, you will individually review a few specific sections of that (same) other team's final report and will address some specific questions about that report. It will be approximately 250 words and is due at **5:00pm (NOT 11:59pm) on May 10, 2024**.

Your assignment for each of the five parts above will be distributed in separate "assignment" documents on the dates stated in the course calendar.

As with real-life system designs, the 6.1800 design project is under-specified, and it is your job to complete the specification in a sensible way given the stated requirements of the project. As with designs in practice, the specifications often need some adjustment as the design is fleshed out. Moreover, requirements will likely be added or modified as time goes on. We recommend that you start early so that you can evolve your design over time. A good design is likely to take more than just a few days to develop. A good design will avoid unnecessary complexity and be as modular as possible, enabling it to evolve with changing requirements.

Large systems are never built by a single person. Accordingly, you will be working in teams of three for this project. Part of the project is learning how to work productively on a long-term team effort. **All three people on a team must be in the same tutorial.**

<u>Late submission grading policy</u>: If you submit any deliverable late, we will penalize you one letter grade per 48 hours, starting from the time of the deadline. For example, if you submit the report anywhere from 1 minute to 48 hours late and your report would have otherwise received a grade of A, you will receive a B; if you submitted it 48 hours and 5 minutes late, you will receive a C.

**You must complete the three team design project components above to pass 6.1800. For the other two (individual) components of the design project, the contribution to your overall grade will be whatever grade you receive on that component. Thus, if you choose not to do one or the other of them, you will receive an F for that component only as a contribution to your overall grade.**

# I. Introduction

Bike share systems, like the Bluebikes you see around Cambridge and Boston, can make transportation less expensive and more enjoyable for citizens, and reduce pollution. In this project, your team will design an enhanced bike share system for the city of Newplace (pop. 1 million). Based on existing systems in other cities, they are expecting approximately 50,000 citizens to become members of the bikeshare, and to have around 20 million completed rides per year. Newplace is currently accepting bids for this system. Your team works for Bikes4All, who would like to make a successful bid to design, build, and operate the system.

In addition to allowing riders to rent and return bikes, this enhanced system will include several features that you don't see in standard bike share systems:

- The ability to (potentially) support at least three distinct types of bikes.
- The ability for riders to reserve a bike for pick-up at the start of their ride, and to reserve a dock for drop-off at the end.
- Support for an experimental camera module on certain bikes.
- The ability for the system to remain working through certain types of communications and/or power failures.

The objective of the system is to meet the needs of the city and as many citizens as possible, while giving them an enjoyable and effective experience as well as respecting their privacy.

In this document, we first provide an overview of the whole bike sharing system including both the core modules or elements of the system and an overview of the information flow and functionality. We then provide more details on the functionality to be supported in Section III. Section IV gives more details about the existing infrastructure. The document concludes in Section **Error! Reference source not f ound.** with a high-level review of what your system should achieve and further questions to consider as part of the design.

# II. System Overview

We begin this by highlighting the underlying modules of the system, on top of which you will be designing your own system. We then briefly summarize the functionality of the whole system.

## A. Modules

The bikeshare system consists of bikes, stations containing a kiosk and set of bike docks, the central computing facility, and riders. In addition, there is a website and phone app that provide rider access to the system. The company that wins the contract – hopefully Bikes4All! – will handle the operation of technology on the bikes, stations including the kiosks and docks, the central computing facility, and the capabilities of the phone app.

We'll start with a high-level description of each of these components; Section IV includes more details.

### 1. Bikes

Bikes4All utilizes three types of bikes in their systems:

- **Basic bikes:** These bikes are less expensive for the company to operate and have a lower failure rate, but they have only extremely limited communications capability. They are only available to riders with accounts (more about this below).
- **Standard bikes:** These bikes communicate with the rider's phone running the app using either Bluetooth, cellular data or WiFi to provide common functions. They have a more sophisticated electronics system than basic bikes.
- **Ebikes:** These bikes include a pedal-assist electric motor. They are significantly more expensive, dependent on being charged, heavier to ride, and the motors have a non-zero failure rate. They are otherwise identical to standard bikes, but some riders are very passionate about them.

Each bike is uniquely identified and has at least a limited capacity to report its current GPS location via a very low-capacity long range radio system. For the purposes of this project, location information will have three components, the latitude (from the Equator) and longitude (from the Prime Meridian) in degrees as a floating point number, and the time at which it was take. Bikes are equipped with remotely controllable locks, and a small number of bikes have experimental camera modules.

At a minimum, your system must support the use of standard bikes. Support for the other two bike types is optional, though it will improve user satisfaction (and possibly other qualities) if the system supports multiple bike types. Your design is more likely to be accepted for the contract with more types of bikes included in it.

## 2. Stations

Bikes are stored in **docks** at a **station**, and each station also has a **kiosk** at one end containing an interactive touch screen, a card slot for credit/debit and bike cards (see below), and a keyboard. The screen is used both for specifics about borrowing a particular bike and additional useful information such as the number of bikes and docks available at nearby stations.

Bikes are locked into a dock unless unlocked through confirmation of an account or payment. Once payment or an account is confirmed, the system will cause the dock to unlock the appropriate lock (the system does this, so you do not need to design this bit) and can be adjusted for fit and ridden. To terminate a ride the bike must be inserted into a dock. Docks are design so that the rider sits the bike snugly into the dock and the dock automatically locks and connects the bike to the USB and power plugs At that point any exchanges between the bike and the system will need to occur – you are designing those exchanges.

In case of power and/or communications failures, the stations should be prepared to continue some level of rider service. In support of that, each station will include battery back-up, but part of your responsibility is to design for the information required at a station for it to continue with "isolated" service.

As a contribution to the system, Newplace will build enough stations for the bikes so that every station is within 1,000 feet of at least one other. Newplace is also prepared to support the electricity and communications infrastructure to connect the stations to a central location.

## 3. Riders

There are two broad categories of **riders**: **members** and **non-members**. Members have accounts and can be identified by either an app running on their phones, or a *bike card* provided by Bikes4All (only members can receive a bike card). Members may have a credit or debit card associated with their accounts, but the city would also like to allow members to pay with the equivalent of cash, by adding "money" to their bike cards. This can be done with cash payments at several public locations such as banks, post offices and public libraries. Thus, people can be members without having to have a phone or credit card.

Members can borrow a bike either with the phone app or a bike card at the kiosk. At present, they cannot use their credit/debit cards to directly access their membership. If they have only a credit/debit card at a kiosk, they will be treated as non-members. Non-members – visitors, tourists, etc. – can only borrow a bike with a credit or debit card at a kiosk.

Bikes4All will support website login to manage accounts and make reservations. A separate team will be designing the user interfaces for this site, but your team will work with them on the information being exchanged between the app or website and the core of the system.

It should be noted that although the city of Newplace requires that all bike riders on city streets wear helmets, enforcement of that is outside the purview of the bikeshare system itself.

### 4. Centralized Computing Infrastructure

The centralized computing system will keep track of several types of information including (but not limited to):

1. **Basic management information.** For example, user accounts, data about individual bikes and docks, etc.
2. **Information needed to make long-range decisions.** For example, decisions about the (re)placement of docking stations, the numbers of bike docks in those stations, the number of bikes in the system, etc. Bikes4All may also want to develop profile models of types of riders to better serve them.
3. **Information required by the city** both for monitoring the efficacy of the system and decisions about road and traffic management. To achieve this, the city may require reporting on ridership among different categories of riders as well as locations. The city will also need to understand riding routes including times of day and week.

You will need to specify the size and nature of each of the sets of data and consider the storage of accident videos (see Section IV.A.3).

### 5. Communication between Riders and the System

For your bikeshare system to be useful, riders must be able to communicate with the system. For members the situation is a bit more extensive than for non-members. There are three ways that members can communicate with the system: the app, the website, and kiosks at the stations. Members can use any of these three to interact with the system and do things such as borrow a bike. When on a ride, the only option for direct communication is through the app.

Non-members have two options for communication: the website and kiosks. On the website, they can only receive information about the system, bike availability, etc. At a kiosk, they can provide a credit/debit card and borrow a bike as well.

Your team is not responsible for designing the user interfaces for any of these modules but will handle the information flow (both sending data to these modules and collecting data from them). In many cases, you will be required to display particular information to a rider (for example, that they cannot borrow a particular bike because it has been reserved by another rider); see Section IV for more details.

# III. Expected features of the system

Your system is meant to enable riders to easily share bikes effectively at a reasonable cost, while respecting their needs, expectations, and personal privacy. This section addresses the functionality of the whole system in more detail.

## A. Basic Functionality

At the most basic, riders should be able to rent bikes for a ride and return bikes when their ride is done. The system prioritizes shorter rides: members can ride a bike for up to 45 minutes for free, and non-members can ride a bike for up to 30 minutes for an initial flat rate. There is a fee schedule for longer rides, as shown in Table 1. This table also contains fees for additional features discussed further below.

*Table 1: Bike rental rates*

| | Membership | Non-membership with credit/debit card[1] |
|---|---|---|
| Base ride time | 45 min | 30 min |
| Base ride cost | Free | $2.95 |
| Each additional 15 min | $2.00 | $4.00 |
| Electric motor bike unlock fee | Free | $2.95 |
| Electric motor bike surcharge | $.10/min | $.25/min |
| Camera surcharge fee | $2.00 | $5.00 |

To borrow a bike at a station, the rider will either interface with the phone app, notifying it of the station ID at which they wish to rent a bike, or use their bike card and the kiosk. Riders can indicate the number of the dock from which they wish to get a bike or let the system assign them one. If they choose a bike that is not available – for example, one that has been reserved by someone else – they can either choose an alternate or let the system choose for them.

While riding, standard and ebikes must notify the rider with a 5-minute warning on the bike screen that the ride will incur a charge to extend the period. The company is not sure that it is possible to do this for Basic bikes, but if you can include that in your design, they would like it.

Finally, with respect to these charges, the company understands that for riders using money on their bike cards, it may turn out that a ride lasts longer than there was money on the card for it. To handle that, the rider will not be able to borrow another bike on their account without first paying (into the bike card) to cover the extra charges and then adding enough more for another ride.

## B. Reservations

In popular areas of Newplace, commuters often head to a station to rent a bike, only to find that there are no bikes there when they arrive. Similarly, on bike rides home, commuters may arrive at a station only to

[1] In the future there may be longer-term plans for non-members, such as multiple hour plans, but those are not considered here.

find that there are no available docks. To address this, the city wants to enable riders to make *reservations*.

Reservations can be for pickup, dropoff, or both. For pickup reservations, using either the app or the website (see below for more detail), a customer can look at the map of available bike locations and reserve whatever is convenient. The customer then has fifteen minutes to pick up the bike at that location; if they arrive within fifteen minutes, the bike should be there for them to rent.

Dropoff reservations work similarly: customers select a dock to reserve, and if they arrive within fifteen minutes of their reservation time (either fifteen minutes early or fifteen minutes late), the dock will be available for them to use. Dropoff reservations can also be made at a kiosk, using the member's bike card.

Each rider can make no more than four bike reservations and four dropoff reservations per day (midnight to midnight) and reservations can only be made by members.

## C. Improving Bike Availability

To help improve bike availability, Bikes4All wants to incentivize riders to move bikes to high-demand locations. An *Angel* is a rider who rides a bike from a designated low-demand location to a designated high-demand location. For each such transfer, the Angel is given a credit worth an additional 45 minutes on a ride in the form of three 15-minute ride extensions to a ride that can be used together or separately.

To enable this feature, another team at Bikes4All has designed an algorithm and written the code to determine which bikes should be moved where. This algorithm takes into account the demand for bikes as well as the actual availability of bikes and docks. Your system will receive information from this *angel tool* that indicates which bikes are candidates to be moved, and to where. If a member wants to be an Angel, your system should assign them a specific bike to pick up, and a specific location for delivery.

Your system will also need to provide regular information about usage patterns, so that the angel tool can update its model. The tool needs to build a model of bike traffic and demand, to determine the target occupancy of bikes at specific stations. It also needs to identify a plan for moving particular bikes for which an angel can sign up. To do that, you will need to decide which information is provided to the angel tool. Again, you are not designing the angel tool, only exchanging information with it.

## D. Abandoned Bikes

Bikes4All also wants to incentivize riders to rescue abandoned bikes (bikes that have been left by their rider and not returned to a dock). *Heroes* are riders who perform this operation. Heroes receive $10 in credit toward their next membership renewal. These credits can accumulate to future years. A *Superhero* is a hero who has returned 20 bikes and receives a life-time membership for free.

To support Heroes, your system will need a way to identify when a bike has been abandoned. You will need to decide which kinds of information available in the overall system contribute to defining a bike as abandoned, and explain and justify your choice of information, particularly in light of balancing user privacy (see Section G).

Once your system has determined that a bike is abandoned, the system should lock the bike (via its remotely controllable lock). After that, any member should be able to access the abandoned bike information to locate these bikes and sign up to retrieve them. Once a Hero arrives at the bike, the system will need to unlock the bike; this is done via the phone app, by either typing in the bike's ID number or scanning its QR code. See Section IV.A.1 below for more detail.

## E. Experimental Camera Modules

Bikes4All is going to begin testing the inclusion of cameras on a small number of standard bikes and ebikes. The cameras are meant to allow riders to record personal videos along their ride and transfer those videos to their phones, and to record accidents. Accident videos will help Bikes4All understand riding skills and riding conditions, as well as estimate possible bike repairs needed.

Your system must send any accident videos to the central computing facility, though there is no time constraint on this requirement. Note that a user might return a bike with an accident video still stored on it, and another user may try to borrow that bike quickly thereafter (before your system has sent the video to the computing facility). You will have to choose how to handle this situation, but the video should not be deleted from the video storage on the camera (see Section IV.A.3) until the transfer to the computing facility is complete.

This scenario is possible with personal videos as well, but personal videos are not required to be sent to the central computing facility. You will need to decide how to handle their transfer, deletion, etc.

## F. Failures

The city of Newplace has specified that the stations must remain at least partially available during a power or communications outage. The stations are dependent on the central computing system both to receive and provide information. Your system will need to make sure that any information required for basic operation of the station is available locally on the station in case of failure. In addition, your system should be able to support the docks and bikes at least to some extent during such a failure. The city does not understand how much functionality can continue, so you will need to specify that. Minimally, riders should be able to return their bikes from rides. Any additional functionality would be welcomed, such as letting riders borrow bikes during such an outage. You should also think about what information can and should be displayed on the kiosk during these types of events.

## G. Data Collection and Privacy

There is a lot of data that your system can collect that would be useful to both Bikes4All and the Newplace government. For example:

- How long are the rides? Where do they start and end? Where do they go?
- Is there a way to estimate how much of the riding is commuting vs. errands vs. riding for joy and entertainment? How frequently is each rider doing each type of ride?
- Is the system reaching underserved or lower-economic communities?

By gaining a better understanding of the bike usage of the roads, Newplace may be able to improve the safety and traffic management of roads, and Bikes4All may be able to adjust and prepare for further growth. Bikes4All will also need to understand the level of staffing needed for maintenance, how to make decisions about Angels moving back from less needed to more need locations, etc.

Many of these questions can be answered by using data collected from the GPS trackers and/or cameras on the bikes. However, GPS trackers and cameras often reveal personalized information. You will need to decide how much of which sort of information should be provided to both improve the city and company operations while balancing users and privacy.

### H. Summary of Rider Communication with the system

In order to enable the functionality discussed above, riders will need to interact with the system. We summarize the core communication functionality below; Section IV presents them in more details. In some cases, there will be pieces of information that the system needs to communicate to riders; this is also discussed in Section IV. In addition, the Appendix provides a broad sampling of interfaces.

*Table 2:Types of functions available through different interfaces for members and non-members*

| Functionality | App | Kiosk | Website |
|---|---|---|---|
| Borrow a bike | Members-only | All riders | None |
| Make a reservation | Members-only | Members-only | Members-only |
| See map of station information, including bike and dock availability at all locations | Members-only | All riders | All riders |
| Manage accounts[2] | Members-only | Members-only | Members-only |
| Manage being an Angel | Members-only | Members-only | None |
| Manage being a Hero | Members-only | None | None |
| Manage videos | Members-only | None | None |
| See route maps before and during a ride | Members-only | None | None |
| Manage routes during a ride | Members-only | None | None |

## IV. Detailed System Components

Here, we present more details about each component of the system, as well as some more detailed requirements needed to support the functionality described above.

### A. The bikes

#### 1. Identifiers

Each bike has a system-wide unique *identifier* (ID). The ID will be written on the bike itself as well as encoded in a QR code that can be read in from the bike system app. You are responsible for designing these identifiers.

#### 2. Electronics System

Each bike has an on-bike electronics system, of varying complexity depending on the type of bike.

[2] Registering for accounts and billing are outside the purview of this system, although they can be done from the website or app.

### *a) Basic Bikes*

Basic bike system contain only:

- A GPS receiver
- A LoRa two-way radio system. This radio system has a long range (1-2 miles) and a very low capacity (average 10Kbps, with a range of 0.3 – 25Kbps). The LoRa radio is supported by an internal battery and is off (not listening) for most of the time. If it wakes up to send and receive one message every 30 minutes, the battery will last for at least two years. If it wakes up more frequently the battery will last a shorter time. The LoRa system on the bike can also put itself into "listening" mode, so that it is listening continuously, but that takes yet more power; this mode should be used sparingly. The LoRa systems available on basic bikes include a non-rechargeable battery and must be replaced by the bike maintenance crew.
- A remotely controllable lock. The lock can be engaged or disengaged based on data it receives from the LoRa radio.
- A simple "health" monitor for the bike, to report any issues that may require maintenance or repair. This consists of two sensors that will measure tire pressure and breaks and notify the on-board processor of the status of the tires and breaks.
- Two indicator lights that can be on or off, which can be used to convey information to anyone near the bike.
- A minimal processor that can manage the storage of GPS information, operate the indicator lights, receive and process the tire and brake "health" sensors, and operate the remotely controllable lock.
- A simple processor with enough storage to save up to one hour of GPS recordings from the receiver, the bike's unique ID in the system, and its current "health".
- A connector used to "plug" into a dock. For the Basic bike this supports only USB 3.1 (a limited capacity protocol, providing a maximum of 600MBps). When a rider puts a bike into a dock, the bike's ID and health will be reported through this interface. In addition to any further GPS information not previously reported can transmitted.

You will need to determine which GPS information is stored on the bike and transferred by either the LoRa or USB 3.1 connecter and on what schedule. There are many trade-offs to consider: How much accuracy do you need? What happens if the ride lasts longer than one hour? What happens if another rider takes the bike out again in quick succession? Can all the data you wanted to retrieve from the bike be retrieved without significand delay, if you only send it at the dock through the USB 3.1 connector? Etc.

### *b) Standard Bikes*

Standard bikes contain a more sophisticated electronics system. In addition to the functionality that the Basic bikes have:

- A small screen, so that additional information can be provided to the rider during a ride. At a minimum the screen should report the amount of time remaining in the current ride period and the total accumulated charges.
- Increased storage, so more information can be stored on the bike, such as route information, perhaps simple pointers for tourists, etc.
- An extended set of lights: up to four lights that can be red, yellow, green or off. These lights can also be used to signal information to the rider and are likely less distracting than the screen.
- Higher capacity wireless communication of your choosing. This may be Bluetooth, cellular data, or wifi. You will need to decide which of these options best suits your design. You will be making

one choice for your whole design, so all the standard and ebikes in your system will use your single choice for higher capacity wireless communications.

Since standard bikes have multiple wireless communications options, it's important to understand the differences between them:

- Bluetooth is used for short-range communication (<= 30ft). It does not transmit through metal and many other building materials but has higher capacity than LoRa (packets can be up to 64 kilobytes long and 2.1 megabits per second). Because of the short range, Bluetooth data can only be sent from the bike directly to a user's phone. There is no dollar cost for using Bluetooth and it can only be used to communicate directly with the phone.
- Cellular data can be used for longer-range communications; cellular devices can be up to 1-3 miles from a tower, although this is line of sight, so in urban areas cellular companies place "towers" much closer together, and there still are dead spaces. Bikes that use cellular communications will communicate directly through the cellular network; they cannot peer with a phone and communicate through the phone. Data can be transmitted at up to 80Mbps, and costs $50/month per bike. This service provides only direct communication with the central computing facility.
- WiFi (802.11ax) will achieve between 20Mbps and 2000Mpbs (depending on location). The ISP has promised that they will provide enough hotspots that for all intents and purposes there are no dead spots, only lesser capacity areas. As with the cellular service, the devices cannot peer with phones directly using this Wifi. The local ISP has equipped the city with hotspots and contracts for this service are $75/month per bike. This communication can only support communication directly between the bikes and central service.

There are trade-offs among all of these options. For instance, there is no cost for a bike to communicate directly with a phone using Bluetooth, but any data that then needs to get from the phone to the central service through the app on the phone may be an added cost for the owner of the phone. In contrast, the cellular data and WiFi options will require that Bikes4All pay per bike for either of those services. If you do not opt for Bluetooth, you will need to consider how much of the data that is going initially to the central server also needs to be delivered to the phone through the app. There is no perfect answer here, so your design will need to explain the tradeoffs you have considered and justify your choice.

#### c) Ebikes

The highest end bike option is the ebike. Ebikes have the same electronics system as the standard bikes but come with a long-lasting pedal-assist electric motor. The motor has a separate indicator of how much battery capacity remains, so the rider can decide whether it is adequate for their intended ride. This bike is heavier so riding it without an operational motor is more work. In addition, there are surcharges for using this type of bike, but some riders are passionate about them.

### 3. Cameras

Bikes4All is going to begin testing the inclusion of cameras on standard and ebikes. The cameras have two purposes:

1. To allow riders to record things along their rides and transfer them to their cell phones. After a bike with a camera is returned to a dock (within some limited time period to allow for extraction of a final recording), these videos should be cleared from the bike.

2. To record accidents. These will be saved on the bike until they have been transferred to the system (either station computer or central computer – your choice) for analysis. Depending on the protocol for transfer, these accident recordings may remain in the camera for longer. You will need to decide whether a rider can access such accident recordings of their own rides.

Because the cameras are a new and untested feature of shared bikes, Bikes4All has decided that they will provide these on only 10% of the standard and ebikes. The cameras that Bikes4All are considering are medium quality, but highly ruggedized to withstand frequent use. Each camera will have two lenses, pointing forward and toward the rider (backward). They can store 30 minutes of video in any sized time periods, with additional capacity reserved for recording accidents. The camera will be equipped with an "accident" sensor (gyroscope and impact sensor). When this sensor raises an alert, the camera starts recording an "accident" report (thus terminating any ongoing recording) on both the forward and backwards lenses simultaneously at half the frame rate (so the same rate of accumulation, but two streams) for three minutes per event.

The specification for the camera includes:

- HD video: 1280x720 pixels, 30 frames per second (fps), emergency recording is two streams at 15 fps each.
- Communications when the bike is not docked: the camera is wired to the tiny onboard processor, so its data transmission utilizes the communications of that processor. You will be deciding whether this is Bluetooth, cellular data, or Wifi.
- Communications when the bike is dock: In this case, the processor uses the bike "plug" supporting USB 3.1 (600MB/s transfer rate) and communicates with the station system and from there into the rider's account in the central facility using the high speed wired network.
- Battery capacity: 45 minutes of recording time with a maximum of 30 minutes for rider recordings and the remainder of the time reserved for emergency recording.
- With compression on the camera, each frame is 1.5MB (with an accumulation rate 45 MB/s)
- Total storage is 128GB with 35 GB reserved for accident video recording

## 4. Summary of Rider Communications with Bikes

As suggested above, there are a number of ways for riders to communicate with the bikes.

- Each bike has set of indicator lights for communicating a very small amount of information to the rider directly. At a minimum, these lights should indicate whether the bike is "healthy". If it is not, the rider is encouraged to take it to a station and swap it for a healthy bike. The lights must also indicate if the bike has been remotely locked because it is deemed abandoned. You may choose to communicate other types of information with these lights.

- Standard bikes and ebikes have a small computer and screen. This screen should convey:
  - The amount of time remaining for the current charge
  - The amount of the current charge
  - Time remaining on a return dock reservation if one has been made
  - If the bike has a camera, whether all video has been transferred
  - Ebike battery level
  - Anything else of your choosing

- Members can communicate with the bike through the app. Depending on the type of bike and your design choices, this communication may happen directly via Bluetooth, or indirectly via cellular data, WiFi, or LoRa. There is much you can do with this type of communication, but at a minimum, you'll need to use it to support Heroes rescuing an abandoned basic bike.

## B. Bike Stations

Each bike station consists of several docks, the bikes in the docks, and a kiosk, all supported by a small computer.

### 1. The station as a whole

Each station is a separate subsystem. It supports 15 to 45 docks and the kiosk and has the following configuration:

- Intel Xeon single processor
- 16 GB memory
- 1 TB storage
- 1 Gbps network interface (symmetric, so 1 each way)
- Battery backup for 2 hours of operation of the whole station

Newplace has specified that unlike older bike share systems, if there is a communications or power failure, the station should continue working at least for a while. Your design will need to reflect this requirement and explain the extent to which your system can operate under failure.

### 2. Docks

In terms of the system you are designing, the docks are simply components of the station itself. Besides being a storage and locking facility for the bikes, each one will have a set of 4 indicator lights (red/green/off), a power source for charging any batteries on the bike (other than the LoRa system), and a plug carrying USB 3.1 for data between the bike and the station. The plug will allow for reporting the bike ID, any GPS information, bike health information as measured by the health monitoring system, and any video that needs to be transferred.

The plug also communicates with the various batteries on the bike including for the motor on an ebike, for the processor, and for the camera. All batteries other than for the LoRa radio can be recharged through this plug, although for the camera and motor this may take time and may only fully recharge with significantly long stays such as overnight.

### 3. Kiosks

The interface on the kiosk consists of a touch screen, a keyboard, and a card reader for both credit/debit cards and bike cards. The screen can show a map of bike stations to anyone, without a card. To begin the process of borrowing a bike, a card (either credit/debit card or bike card) must be provided in the card slot. Both the touch screen and keyboard can be used for entering information into the system, such a preferred bike ID, choices for reservations, etc.

You are encouraged to indicate additional information that would be useful to the riders and how it will be provided.

### 4. Summary of Rider Interfaces with Stations

At a dock where a bike is present, the set of indicator lights should indicate at least the following set of conditions:

- The bike has been returned but is copying videos to the previous rider, and hence is unavailable until the transfer is complete.
- The bike has been returned and is available to another rider.
- The bike is unlocked based on assignment to a particular rider and is available to be removed by the rider for riding.
- The bike has been returned, but in need of maintenance and hence is unavailable until it has received attention.
- The bike has been returned but is reserved.
- The dock is empty and available to receive a returning bike.
- The dock is empty but is reserved for a returning bike.
- The dock is empty but needs maintenance and hence is unavailable.
- If the bike is an ebike an approximate indication of the motor battery level.

Your design should allow for an extension of this list of conditions in the future.

Notice that because of the reservation system, it is possible for a rider without a reservation to arrive at a station and find a bike, but not be able to borrow it, because it is reserved. In that case, the kiosk should allow them to ask how much more time remains on the reservation. The same holds true for return or dock reservations.

## C. The central computing facility

This computing facility will support computation, storage, and communication at the center of the whole system. It contains:

- Intel Xeon CPU Max Series processor with 56 cores
- 256 GB DRAM memory
- 100 TB storage
- 1 Tbps network capacity (symmetric, so 1 Tbps each way)

The central system is expected to be handle the full load of connections to it from riders in the app or on the website, bikes, and the stations. It has the capacity to handle at least 5,000 simultaneous connections from the combination of the app, the website, the kiosks, the docks, and the bikes.

This system will both keep track of all the management information, such as user accounts, individual bikes, as well as information needed to make long-range decisions about the (re)placement of docking stations, the numbers of bike docks in those stations, the number of bikes, etc. It will also need to provide the information required by the city both for monitoring the efficacy of the system and decisions about road and traffic management. You will need to specify the size and nature of each of the sets of data (in files or a database[3]). You will also need to consider the storage of accident videos.

Although the central computing service may be at the center, it is possible that some information should either never go there, or if it does, should not be stored there permanently. You will need to design the information flow for communicating with bikes, docks, and the station kiosks with this in mind.

---

[3] We will provide you with a brief summary of the differences between a file system and a database system, to assist you in deciding whether and when to use which type of system.

## V. Reviewing your assignment

With all this detail in mind, it is important to review the basics. First, you are designing a system that will manage the flow of information needed to operate the enhanced bike share system for the city of Newplace. Information can be collected in a number of places and needs to be copied, moved, possibly anonymized, aggregated, and so forth in order to achieve the intended functionality. In the end, this is all in the service of making bike sharing available to both citizens and visitors.

Second, your system must meet not only some operational goals, such as providing as much accessibility to rides as possible, but also several societal goals. The first is to understand about the safety of their roads. Second, Newplace is focused on bike sharing becoming increasingly available to previously underserved communities. They want to understand whether and how this system will achieve that. Third, they are concerned about the privacy of the riders involved, both with respect to the information that the provider (Bikes4All) will keep about the riders, but also with respect to the information provided to the city.

To do all this, you will need to enable support for:

- At least standard bikes, and preferably, also basic and ebikes. Users should be able to borrow, drop off, and make reservations. The system should support the appropriate interactions via the phone app, website, and station kiosks. It should charge users accurately and notify them as charges will increase. The system should conform to the constraints given in the spec (e.g., network capacities). Doing this requires that the system report the required messages (things like whether a bike has been reserved, etc.) and that students choose an appropriate communication technology for bikes to transmit.
- Both members and non-members, who are likely to be visitors to the city.
- Reservations for both pickup and dropoff of bikes for members.
- Both Angels and Heroes.
- At least a relatively small number of cameras on bikes as indicated in the spec.

You will need to make many trade-offs to support the functions above. There are also places where you will need to make “big” trade-offs:

- Balancing useful data vs. privacy
- Balancing functionality during failures
- Managing getting the video off the camera effectively
- Working with basic bikes that have extremely limited communications
- Managing the remote locking and unlocking of abandoned bikes

Remember that **there is no single right design**. You will need to make many design choices, justify them, and explain them in terms of other alternatives that you rejected. You may need to make reasonable, educated guesses about functionality that the spec leaves unspecified.

# VI. Appendix: A partial list of interfaces

This is a sampling of the interfaces in the system. It should give you a sense of both the kinds of requests that parts of the system can make of each other and user interface requests. It is intentionally incomplete because the design of some of the requests is part of your challenge. Note that every element of the overall system has its own address, so these messages are all addressed to particular elements (e.g., a particular bike, dock, station, etc.). If no return values are specified, you can assume just a simple return. These are in no particular programming language. Square brackets ("[]") are used to indicate a list.

## A. Bike to station through dock (through USB 3.1 plug)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `check_in(bikeID)` | | |
| `check_out(bikeID, riderID, rideID)` | | Both rider and ride need IDs |

## B. Station through dock to bike (through USB 3.1 plug)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `assign_bike(riderID, rideId)` | | |
| `get_status()` | `tire_status, brake_status, rider_video_to_send_flag, rideID accident_video_to_send_flag, rideID` | |
| `get_accident_video(riderID, rideID)` | `accident_video` | This is incomplete. You will need to complete it for your assignment. It also may not be used at all. |

## C. GPS to central system (using LoRa)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `report_gps_location(bikeID, number_of_reports, [list_of_gps_reports])` | Number of reports received | |
| `turn_listening_mode_on(bikeID)` | | |
| `turn_listening_mode_off(bikeID)` | | |

## D. Central system to GPS (using LoRa)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `get_status()` | `tire_status, brake_status,` | All location information is in the form of latitude, longitude |

| | | |
|---|---|---|
| | `battery_levels`, `GPS_location` | (floating point degrees) and a timestamp. |
| `lock_bike()` | `True` if locked successfully; `False` otherwise | Lock bike not in dock. Note that there may be other ways to lock/unlock a bike |
| `unlock_bike()` | `riderID, rideID` if unlocked successfully; `null` otherwise | Unlock bike not in dock. Note that there may be other ways to lock/unlock a bike |

## E. Bike to central system (using cellular data or WiFi)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `send_rider_video(bikeID, riderID, rideID, video)` | `rideID` if received successfully; `null` otherwise | This may not be exactly what you need |
| `send_accident_video(bikeID, riderID, rideID, video)` | `rideID` if received successfully; `null` otherwise | This may not be exactly what you need |

## F. Central system to bike (using cellular data or WiFi)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `get_status()` | `tire_status`, `brake_status`, `battery_levels`, `accident_video_available`, `GPS_location`, `station_if_docked` | |
| `lock_bike()` | `True` if locked successfully; `False` otherwise | Lock bike not in dock. Note that there may be other ways to lock/unlock a bike |
| `unlock_bike()` | `riderID, rideID` if unlocked successfully; `null` otherwise | Unlock bike not in dock. Note that there may be other ways to lock/unlock a bike |
| `get_accident_video()` | `riderID, rideID, video` | |

## G. Central system to station

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `reserve_bike(riderID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |
| `reserve_dock(riderID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |
| `reserve_bike_for_angel(riderID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |

| | | |
|---|---|---|
| `reserve_dock_for_angel(riderID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |

## H. Station to central system

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `reserve_bike(riderID, stationID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | Reservation made at kiosk for another station |
| `reserve_dock(riderID, stationID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |
| `send_rider_video(bikeID, riderID, rideID, video)` | `rideID` if received successfully; `null` otherwise | |
| `send_accident_video(bikeID, riderID, rideID, video)` | `rideID` if received successfully; `null` otherwise | |
| `get_non_member_id (credit/debit_card_info)` | `riderID` | |
| `new_ride(bikeID, riderID, rideID)` | | |

## I. App to bike (using Bluetooth)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `get_new_video_list()` | `list_of_unretrieved_videos` | |
| `get_video(videoID)` | `video` | |

## J. App to central system

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `get_station_information()` | `stationID, bikes_available, docks_available` | This is probably incomplete |
| `login(riderID)` | | This call is followed by login process |
| `borrow_bike(riderID, stationID, bikeID)` | `True` if successful; `False` otherwise (e.g., if declined) | Bike ID is optional, if not provided, bike will be assigned by the system |
| `report_bike_issues(riderID, bikeID, rideID, set_of_issues)` | | |
| `reserve_bike(riderID, stationID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | Reservation made at kiosk for another station |
| `reserve_dock(riderID, stationID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |

| | | |
|---|---|---|
| `find_route(origin, destination, list_intermediate_stops)` | `route_information_for_mapping` | |
| `revise_route(current_route, route_deviations)` | `route_information_for_mapping` | All route information will be in the form of a series of location points (latitude and longitude as floating point degrees) |
| `get_video_list()` | `list_of_videos_on_server` | |
| `get_video(videoID)` | `video` | |
| `get_hero_information()` | `bikeID, bike_type, location` | Abandoned bikes are not in a dock, so their location is needed to find them. |
| `reserve_bike_for_hero(riderID, bikeID)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |
| `get_angel_information()` | `bikeID, bike_type, current_location, destination_location` | |
| `reserve_bike_for_angel (riderID, bikeID)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |

## K. Website interface (Note that non-members can only get station information)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| `get_station_information()` | `stationID, bikes_available, docks_available` | This is probably incomplete |
| `login(riderID)` | | This call is followed by login process |
| `borrow_bike(riderID, stationID, bikeID)` | `True` if successful; `False` otherwise (e.g., if declined) | Bike ID is optional, if not provided, bike will be assigned by the system |
| `report_bike_issues(riderID, bikeID, rideID, set_of_issues)` | | |
| `reserve_bike(riderID, stationID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | Reservation made at kiosk for another station |
| `reserve_dock(riderID, stationID, time_of_reservation)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |
| `get_hero_information()` | `bikeID, bike_type, location` | |
| `reserve_bike_for_hero(riderID, bikeID)` | `reservation_ID` if successful; `null` otherwise (e.g., if reservation declined) | |
| `get_angel_information()` | `bikeID, bike_type, current_location, destination_location` | |

| | | |
|---|---|---|
| reserve_bike_for_angel (riderID, bikeID) | reservation_ID if successful; null otherwise (e.g., if reservation declined) | |

## L. Kiosk interface (Note that only some of these are available to non-members)

| **Function call** | **Return value** (if any) | **Notes** (if any) |
|---|---|---|
| get_station_information() | stationID, bikes_available, docks_available | This is probably incomplete |
| login(riderID) | | This call is followed by login process |
| create_one_time_riderID(credit/debit_card_info) | riderID if successful; null otherwise (e.g., if declined) | |
| borrow_bike(riderID, stationID, bikeID) | True if successful; False otherwise (e.g., if declined) | Bike ID is optional, if not provided, bike will be assigned by the system |
| report_bike_issues(riderID, bikeID, rideID, set_of_issues) | | |
| reserve_bike(riderID, stationID, time_of_reservation) | reservation_ID if successful; null otherwise (e.g., if reservation declined) | Reservation made at kiosk for another station |
| reserve_dock(riderID, stationID, time_of_reservation) | reservation_ID if successful; null otherwise (e.g., if reservation declined) | |
| get_hero_information() | bikeID, bike_type, location | |
| reserve_bike_for_hero(riderID, bikeID) | reservation_ID if successful; null otherwise (e.g., if reservation declined) | |
| get_angel_information() | bikeID, bike_type, current_location, destination_location | |
| reserve_bike_for_angel (riderID, bikeID) | reservation_ID if successful; null otherwise (e.g., if reservation declined) | |