Divvy Truck Assistant
Project Report

An application to assist Divvy Van drivers by providing optimal routes of travel while keeping inventory in mind.

Prepared by Alfonso Arias, Vivek Ganesan, and Juan Zambrano
For use in CS 440 at the
University of Illinois at Chicago

TABLE OF CONTENTS

Project Description (A)

Requirements (B)

Design(C)

I. Project Description

1. Project Overview

The system is a desktop application where Divvy delivery drivers will be able to receive real time instructions on where to transfer bikes and the most efficient routes to take when making transfers. The system, which keeps track of the number of bikes at each station in real time, will use an algorithm to determine which stations require bikes as well as the quantity of bikes to be delivered. The system will also show the most efficient route a driver should take in order to reach a destination in the shortest amount of time.

2. The Purpose of the Project

The purpose of this project is to allow drivers to transfer bicycles between bike stations in an efficient manner. An efficient manner means that drivers will be able to make all scheduled bike transfers in significantly less time than the current time that it takes to conduct all bike transfers.

a. The User Business or Background

The business of the client is the *Motivate International Inc.*, the company behind the Divvy bicycle sharing system. This company is a bicycle sharing system that operates in the city of Chicago as well as a select number of suburbs in the area. The Divvy corporation operates over 5800 bicycles at 580 stations. Divvy reports an approximate daily ridership of 13,000 and an approximate yearly ridership of over 3 million. Divvy offers a set of fare options for different types of riders.

The system to be developed can be used by other companies that have a similar business model to that of Divvy. In May 2019, the city of Chicago approved a measure for a Scooter sharing system. The solution detailed in this report can also be a means for that as well.

b. Goals of the Project

The goal is to develop a system where the amount of time it takes for a bike transfer from one station to another is reduced. The actual amount of reduction should result in a 10-20% decrease from the current time.

c. Measurement

Divvy currently has a dataset of times that it takes to get from one station to any other station. One example of the data set is showing the length of time it takes to get from 856 S Morgan St to all other Divvy stations in the city of Chicago. Divvy generally collects data every few years to examine rider times. However; after a six month implementation of this system, Divvy will record the dataset again to see if there is a change. The outcome of the system will be considered successful if the changed times meet the goals specified in the goal section of this report.

3. The Scope of the Work

The scope of the work involves developing an efficient system to handle the current workload for delivery drivers.

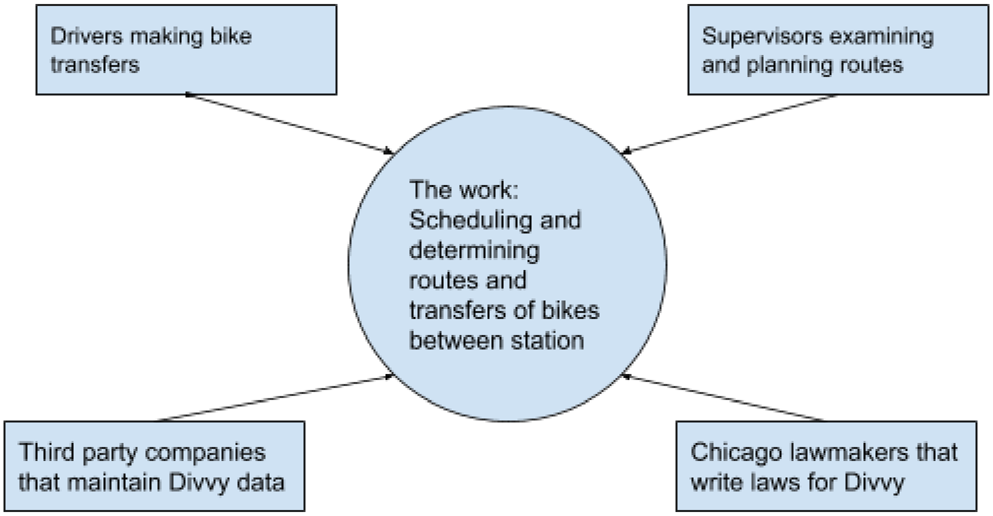
a. The Current Situation

Divvy delivery workers are responsible for getting bikes transferred from station to station. Each station can only hold a certain number of bikes. Delivery workers currently work with a system that gives updates in a real time of how many bikes are at each station.

However, workers are not properly aware of how many bikes need to be transferred from one station to another. Workers have to assume how many bikes need to go from one station to another. An example is that if one station has an excess of Divvy bikes, the worker needs to make the assumption of where to distribute those excess number of bikes.

Divvy workers also do not receive proper information on the most efficient route to get from one station to another.

b. The Context of the Work


Drivers making bike transfers
Supervisors examining and planning routes
The work: Scheduling and determining routes and transfers of bikes between station
Third party companies that maintain Divvy data
Chicago lawmakers that write laws for Divvy

c. Work Partitioning

Event Name	Input/Output	Summary
Divvy delivery workers	Input: number of bikes, location An efficient method to direct drivers on conducting transfers to another destination from current destination.	Divvy workers will report the current station they are in as well as the number of bikes at that current station.
Frontline supervisors of Divvy delivery workers	Input: The number of transfers made by delivery workers These supervisors will receive a report as to how many transfers delivery workers make per shift.	Frontline supervisors would also want to see the system interface that the delivery workers interact with in order to see if there is a gain in productivity.
Industrial Engineers and Operations Analysts	Input: Bike Transfer Analytics These engineers can determine if additional delivery workers are needed to complete transfers.	Industrial Engineers and Operations Analysts will want to be able to determine if the increased efficiency in the software can lead to reduced costs.
Executive Management	Input: Bike Transfer Analytics Executive management can make a decision on the future of Divvy distribution systems.	Executive managers can then make decisions about new policies regarding bike transfers based on the success of the system.

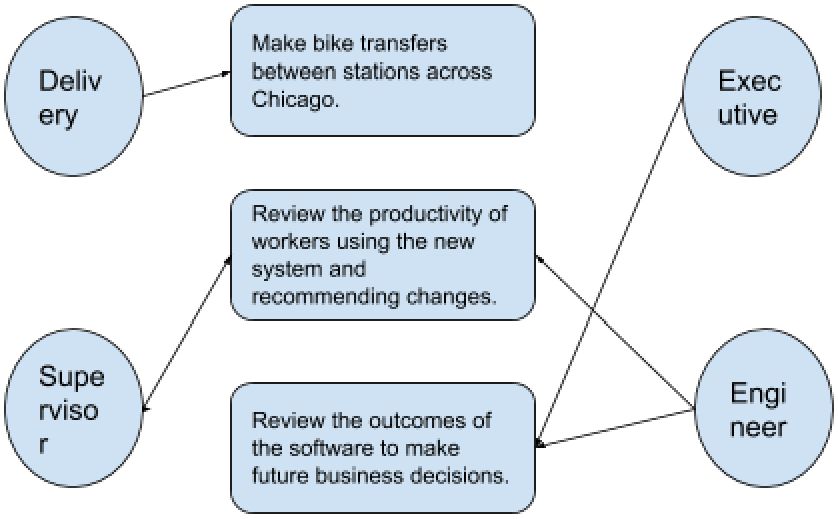
d. Competing Products

As of 2019, there are currently no competing products for this exact system. FedEx, one of the largest shipping companies in the US, currently utilizes a similar system for its drivers to make efficient routes when delivering packages. The purpose of this system is for drivers to avoid driving routes where turns and lane changes are difficult for the enormous sized delivery trucks.

While this system is similar to that of FedEx, the system will also deliver an algorithm for how many bikes should be transferred from one station to another. An excess of bikes in one station will prompt the algorithm to determine how many bikes should be distributed to other stations that experience a shortage of bikes.

4. The Scope of the Product

a. Scenario Diagram


Deliv ery
Make bike transfers between stations across Chicago.
Exec utive
Review the productivity of workers using the new system and recommending changes.
Supe rviso r
Review the outcomes of the software to make future business decisions.
Engi neer

b. Product Scenario List

1. Delivery worker making bike transfers

2. Frontline supervisor assessing productivity

3. Industrial Engineers and Operations Analysts making decisions about efficiency

4. Executive Management making business decisions

c. Individual Product Scenarios

Delivery worker making bike transfers: Ronald, who has worked with Divvy for 5 years, has transferred bikes to and from Divvy stations with the use of the existing system. Using the new system, Ronald will report the current station that he is in as well as a count of how many bikes are at each station. The system will show him all the stations that he needs to transfer bikes as well as how many bikes should be transferred to the station he needs to go to. Ronald will use this software on a laptop computer.

Frontline supervisor assessing productivity: Jillian, a frontline supervisor, wants to see if the software makes a difference in productivity for her drivers. She will be able to login into Ronald's account and keep track of the number of transfers he has made during his shift. By observing Ronald's productivity, she can use this information for performance reviews and evaluations.

Industrial Engineers and Operations Analysts: Sally, an Industrial Engineer for Divvy, is looking for ways to cut costs for next year's operating budget. She currently has data sets for average delivery times from each station to all other stations. She uses the data from before and after the implementation of the software. If she sees the software has met the stated goals, she can make a decision to cut the delivery workforce by a certain amount while trying to maintain the same level of efficiency.

Executive Management: Joe, the COO of the company, wants to decide whether Divvy should expand to the Chicagoland suburbs. He is thinking about opening a Divvy station in Aurora, IL. Based on the data sets comparing transfers before and after the software implementation, the COO can make a decision whether expanding to Aurora is a good idea or not.

5. Stakeholders

a. The Client

The client of this product will be Divvy. Divvy will authorize the purchase of the software and will be in charge of handling all contract negotiations that involve the purchasing of this software.

b. Hands-On Users

The users who are involved in the daily use of this software will be the Divvy delivery workers who deliver bikes from station to station. This means that the system must be simple enough to use for those with relevant education and experience for a delivery worker. The relevant education for a delivery worker is a high school diploma while the relevant experience is generally 1 year or less experience in commercial driving.

c. Other Stakeholders

The customers that use Divvy on a regular basis will be affected by the outcome produced by the software. The goal of this software is to improve efficiency and delivery times of bikes from station to station. As a result, some stations will experience an increase in bikes stationed while others will see little to no change or a decrease in bikes. This can affect customer satisfaction.

The city government of Chicago will also be affected by the outcome of this software. The city government, which primarily regulates Divvy and other transportation sharing companies, may have to change their regulations depending on whether this software produces the outcome desired.

The delivery workers who move bikes from station to station will be affected by this system. Delivery workers are responsible for making sure that bikes get delivered to each station in a timely manner. They are also responsible for making sure that each station has an appropriate number of bikes. This software will improve their productivity and ensure that each station has an adequate number of bikes pertaining to its usage delivered in the shortest possible time.

d. User Participation

All testing will involve all the users of the system, mainly delivery workers and frontline supervisors. Since these workers will be the primary users of the application, they must be involved heavily in testing. When testing the software, the users will use the existing and new systems in parallel to compare the changes in productivity.

e. Maintenance and Service Technicians

The maintenance of this software must be done by a third party company. The third party will be given a set of rights to the use and maintenance of the system. The third party is primarily responsible for making enhancements, fixing bugs, and releasing patches based on the client.

The third party must encrypt all personal and sensitive data. Information involving the employees of the client, the business of the client, and the customers of the client must all be kept private. In the event of a data breach, the third party must assume full responsibility for all losses involving data.

6. Mandated Constraints

a. Solution Constraints

Description: The system shall connect to a database that stores real time information regarding bike information at each station.

Rationale: Divvy corporation already maintains a database with the information stated above. There is no need to create a new database tailored for this system.

Fit criterion: The system must use Divvy's databases.
Description: The system shall maintain fault tolerance in spite of a network connection error.
Rationale: Divvy drivers are limited in their shift hours. A network connection issue can result in a delay of scheduled delivery times. Therefore, the system must be able to recover from any network connectivity issues without manual intervention.
Fit criterion: The system shall maintain internet connection in the vehicle itself to avoid any network disruption.
Description: The system must be usable only on approved Divvy's devices.
Rationale: This is to ensure that other companies, or criminals, aren't provided unauthorized access to Divvy's data.
Fit criterion: The system must be available on a device that is only accessible to the Divvy corporation.

b. Partner or Collaborative Applications

The system must adhere to all laws that are set by the city of Chicago when it comes to regulating bike sharing. This means that the system must not make route decisions and station mapping that in any way, shape, or form violates the laws set by the city.

One example is speeding. Drivers need to make stops within a scheduled time limit. However, the system must schedule routes for drivers allowing them to follow the traffic laws.

c. Anticipated Workplace Environment

Delivery workers' primary work environment is inside a delivery van that transports up to 15-20 bikes between stations. The system must be able to work on a laptop computer installed in the delivery vans when making scheduled stops to deliver bikes. The system will operate on a Windows desktop (XP or a later version) or Mac OS. The system will not be compatible with other operating systems such as Linux or DOS.

d. Schedule Constraints

The Divvy workers who deliver bikes between stations must make sure that each station has an adequate number of bikes. They must perform their tasks in a certain amount of time that does not affect customer experience. This means that Divvy workers must work during times where ridership is low so that bikes can be transferred promptly.

e. Budget Constraints

The Divvy corporation has a set number of drivers and vehicles. This means the software must be able to schedule routes and delivery times for delivery workers without having to work with additional drivers. The system must be able to ensure that drivers work in a designated amount of hours so that no driver works overtime on a given day.

7. Naming Conventions and Definitions

a. Definitions of Key Terms

Efficiency: The tool of measurement to determine whether the system has produced a significant reduction in time.
Mapping: The system that will allow delivery workers to locate from one station to another station.
Strategy: This is defined as a series of steps to take by executive management based on the success of the system. This term is mentioned to explain the use cases.

b. UML and Other Notation Used

The following models that have been utilized in the initial part of this report have been use case diagrams. The use case diagram for this project is based on the use case model that has been used in traditional software engineering.

c. Data Dictionary for Any Included Models

The table provided below serves a model for determining efficiency. The reduction of time will be measured as a percentage from the time resulting from the existing procedure to the time resulting from the new system. Depending on the percentage, a conclusion can be drawn using the table.

Time Reduced (Percentage)	Classification
1-2%	No Change
3-10%	Little Change
10-20%	GOAL - Significant Change
20%-40%	Exceptional Change

8. Relevant Facts and Assumptions

a. Facts

Divvy Annual Ridership: 3 million +
Divvy Daily Ridership: 13,000
Number of Stations: 580
Number of Bicycles: 5800

b. Assumptions

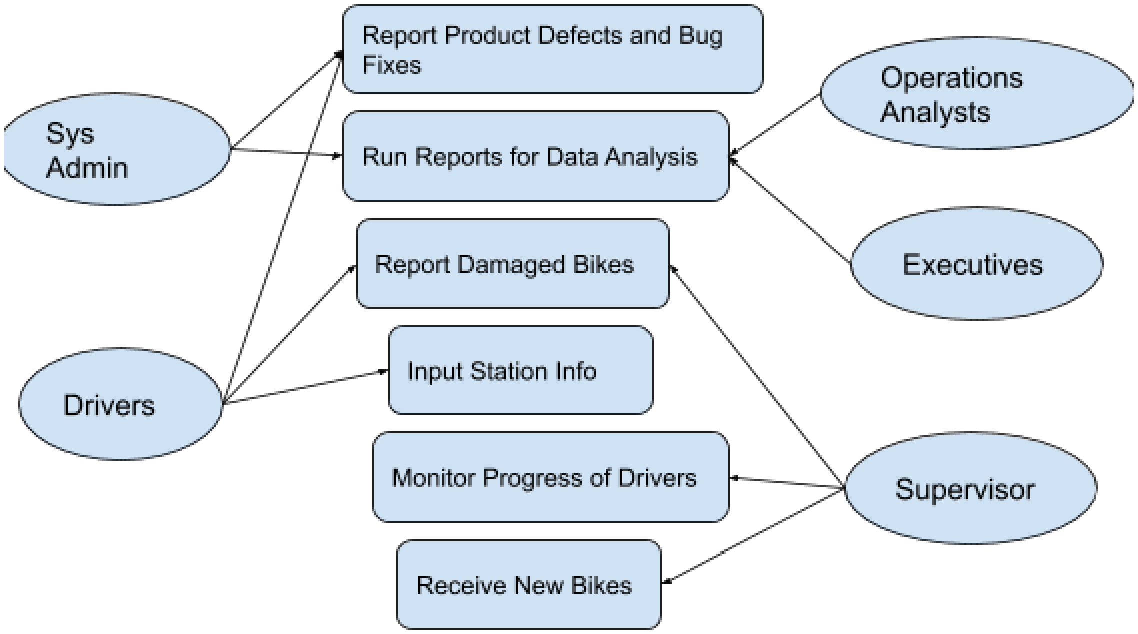
- **i.** Divvy corporation will keep their operations within the boundaries of the city of Chicago. Therefore, planning routes will be kept within a specified domain.
- **ii.** Divvy corporation will not purchase any more bikes, vans, or hire additional drivers. The system should plan routes and transfers with existing resources and infrastructure.
- **iii.** All Divvy drivers have at least a high school diploma. This will determine the complexity of the system and the level of difficulty for the Divvy drivers.
- **iv.** All Divvy drivers are familiar in using Windows operating systems (Windows XP or later vision) or MacOS.
- **v.** Divvy's computer systems either use Windows or MacOS. We can assume that there will be enough delivery drivers available during the testing phase of the system.

II Requirements

9 Product Use Cases

The individual product use cases as well as the product use case diagram are listed below. These diagrams help better illustrate how certain users will use the system.

9a. Use Case Diagrams



9b. Individual Product Use Cases

Use Case ID: 01
Use Case Name: Input station information and current bikes at station
Pre-Conditions:

-> the user has logged into the system and is currently behind the wheel making transfers

Post Conditions:

-> the user will be making a transfer of at least one bike from the current station to a station that is in need of bikes.

Triggered Event: The driver has input the number of bikes at Divvy Station.

Sequence of Events:

1. The user arrives at the Divvy station and counts the number of bikes on the rack.
2. After pressing enter, the user will get a report of which station to transfer the bikes too and how many bikes should be transferred.

3. The user can either accept to transfer the number or click on a button that is unable to complete the current task because the van is too full.

Use Case ID: 02
Use Case Name: Report Damaged Bike
Pre-Conditions:

-> The user has identified a bike that is broken and has logged into the system.

Post-Conditions:

-> The bike should be labeled as defective and the number of operational bikes should be reduced by one.

Triggered Events: The driver enters the ID of the bike that is broken and presses report.

Sequence of Events:

1. The user identifies a bike that is broken and places braces around it so others don't pick it up.
2. The user enters the bike ID of the broken bike. The total number of operational bikes is reduced by one.
3. The report is sent to a server and the mechanic at the Divvy HQ will receive the report.
4. The report will detail the bike ID as well as the station location.

Use Case ID: 03
Use Case Name: Receive New Bikes
Pre-Conditions:

-> The Divvy corporation has received a shipment of new bikes.

Post-Conditions:

-> The system should see an increase in the total number of operational bikes. This should result in the algorithm allocating more bikes.

Triggered Events: The frontline supervisor will press a button that will allow the total number of bikes to be added to the inventory.

1. The frontline supervisor receives the shipment of new bikes.
2. The system will receive the number of new bikes and add the number to its pending count for distribution.
3. A Divvy delivery worker will be responsible for dispatching these bikes using the system and to make sure each station receives an adequate number of bikes.

Use Case ID: 04
Use Case Name: Monitor Progress (frontline supervisor interface)
Pre-Conditions:

-> frontline supervisor has logged into his or her account to monitor driver progress

Post-Conditions:

-> user should receive a report regarding the number of transfers that a particular driver has made during a given shift.

Triggered Events: supervisor clicks run report to receive driver report

1. The user clicks run report to receive a driver's full report.
2. The system will obtain a dataset that contains all the transfers a driver has made during a particular shift.
3. The system will maintain other datasets of transfers the driver has made for the past week.
4. The system will organize the data in the form of a table and will allow the supervisor to electronically access the table.

Use Case ID: 05
Use Case Name: Run Reports
Pre-Conditions:

->The frontline supervisor has already logged into the system

Post-Conditions:

->The frontline supervisor will receive a report.

Triggered Events:

1. The frontline supervisors logs into the system.
2. User clicks run report.
3. A report is generated given the number of transfers for a particular driver and the reported transfer times.
4. The system fetches the data from its servers and populates the report onto the computer screen.
5. The data will be given to executives who will analyze the data.

Use Case ID: 06
Use Case Name: Report Bugs and Fixes

Pre-Conditions:
-> System enters an erroneous state

Post-Conditions:
-> Divvy developers receive a report of the fault

Triggered Event: System enters erroneous state

1. System enters erroneous state.
2. Operating System will report the problem to the development team.
3. User will close out of the system and relaunch.
4. The system will resume algorithm where it left off.

10 Functional Requirements

Determine Efficient Route:

- Description: The system must provide the most efficient route from one station to another that avoids general traffic issues and meets efficiency goals stated in this report.
- Rationale: The system must meet efficiency goals stated in this report.
- Fit Criterion: This requirement is met when the data shows that the percentages desired have been accomplished.
- Acceptance: #1

Determine Number of Bikes to Transfer:

- Description: The system must report how many bikes should be transferred from one station to another.
- Rationale: This is to ensure that each station has an adequate number of bikes pertaining to ridership levels.
- Fit Criterion: This requirement is met when the number of bikes parked at each station is proportional to ridership after all the transfers are done.
- Acceptance: #2

Report Faulty Bikes:

- Description: The system must provide a means for drivers to report the number of faulty bikes.
- Rationale: Faulty bikes cannot be included in transfers.
- Fit Criterion: This requirement is met when the total number of operational bikes is reduced by the number of faulty bikes.
- Acceptance: #3

Provide Reports:

- Description: The system must provide a report that shows average delivery times between stations.
- Rationale: Operations Analysts will need these reports to draw conclusions based on the change in data.
- Fit Criterion: This requirement is met when the system can produce a report that details the criteria listed above.
- Acceptance: #4

Receive New Bikes:

- Description: The system must provide a means for the frontline supervisors to increase the total number of operational bikes when new bikes are received.
- Rationale: The system must be able to work with an increased number of bikes.
- Fit Criterion: This requirement is met when the number of total operational bikes can be increased by the number of new bikes received.
- Acceptance: #5

11 Data Requirements

Station Count:

- Description: The system must be able to keep track of the number of bikes at each Divvy station in real time.
- Rationale: The system will use this information for its algorithm to determine which station to transfer bikes.
- Fit Criterion: This requirement is met when the number of bikes at each station after transfer hours is appropriate for ridership levels.
- Acceptance: #6

Map Data:

- Description: The system must use an API to keep track of the map location of the city of Chicago.
- Rationale: The system will try to determine the most efficient route from one station to the next while trying to eliminate the challenges of left turns and lane changes. Therefore, a map is needed when determining the route.
- Fit Criterion: The map will be implemented in the UI interface so the driver will be able to keep track of routes in real time. The driver will also

get traffic updates so the route will change in the event of a major incident.

- Acceptance: #7

Delivery Times:

- Description: The system shall store the computed average time that it takes to get from one station to another.
- Rationale: The system needs to maintain this data from Operational Analysts and Executive management to make decisions regarding the business of Divvy. Delivery Times will determine the success of the system.
- Fit Criterion: The system will record the amount of time it takes to get from station to station. This will be stored in back-end database not accessible to the user.
- Acceptance: #8

12 Performance Requirements

12a Speed and Latency Requirements

Loading the Application:

- Description: The application must take no longer than one full minute to load.
- Rationale: Speed of opening an application is needed so that the software does not slow down daily operations.
- Fit Criterion: When the amount of time it takes to load the desktop application is less than one minute, the loading of the application is optimal.
- Acceptance: #9

Data to Server:

- Description: All data that goes to a server at Divvy HQ must be returned within 5 seconds or less.
- Rationale: The system must be able to meet the efficiency goals stated in Part I of this report.
- Fit Criterion: The system should return all data sent to it with a response back from Divvy in less than 5 seconds. For example, when a driver sends the number of bikes that are defective, the server should respond with a message that a trigger has been sent to the mechanic.
- Acceptance: #10

12b Precision or Accuracy Requirements

Route Correctness:

- Description: The system must determine the most efficient and correct route to go from one station to another.
- Rationale: An incorrect route based on the map system can be dangerous to the safety of other drivers.
- Fit Criterion: When the correct and most efficient route is determined, the Route Correctness criteria is met.
- Acceptance: #11

Save Work

- Description: At the end of a driver's shift, the system must be able to save all the transfers that have been made.
- Rationale: The frontline supervisor needs a method to assess the driver's productivity during the given shift.
- Fit Criterion: The system must report all the transfer data to the frontline supervisor's account.
- Acceptance: #12

12c Capacity Requirements

Requests

- Description: The system must process at least 100 requests per second.
- Rationale: The system must be able to handle simultaneous requests that come from Divvy drivers as they are making transfers.
- Fit Criterion: A sample of 100 laptop computers will all run tests where they send a piece of data to the server for processing. The server should not report any freezes or slow downs. This can be done through unit testing.
- Acceptance: #13

Bandwidth

- Description: The system may not consume more than 10,000 GB/month in network bandwidth.
- Rationale: This is to help keep the cost of computing resources relatively low.
- Fit Criterion: This requirement is met when the internet service provider for Divvy reports the total monthly bandwidth consumption of the system.

- Acceptance: #14

13 Dependability Requirements

13a Reliability Requirements

Backups:

- Description: The system must perform weekly backups.
- Rationale: The constant examination and interpretation of data is integral to operations analysts and executive management.
- Fit Criterion: The system is successful in providing weekly backups if all data can be backed up and stored onto a data center in the cloud.
- Acceptance: #15

13b Availability Requirements

Shifts:

- Description: The system must be available during the hours of 8 AM and 6 PM all 5 days per week.
- Rationale: Divvy delivery workers primarily make transfers during these hours.
- Fit Criterion: The system is considered successful in this requirement if there are no documented system failures that occur during the first six months of implementation.
- Acceptance: #16

Patches

- Description: The system's patches and bug fixes will be performed when transfers are not in operation.
- Rationale: System patches should not interfere with Divvy drivers' tasks.
- Fit Criterion: This requirement is considered successful when bug fixes are made and no documented evidence of interruption is reported.
- Acceptance: #17

13c Fault Tolerance Requirements

Internet Connection Failure:

- Description: The system will connect to a mobile hotspot authorized by a phone provided by Divvy.
- Rationale: In the event of an onboard internet failure, Divvy delivery workers should still be able to make transfers.

- Fit Criterion: This requirement can be considered when the system can maintain full functionality given a hotspot connection from a mobile phone.
- Acceptance: #18

13d Safety Requirements

Addiction:

- Description: The system UI must be developed in a way that does not addict drivers to the screen.
- Rationale: Use of electronics while driving is the number one cause of road accidents every year.
- Fit Criterion: This requirement is met when there are no documented cases of "using while driving" among delivery drivers.
- Acceptance: #19

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

Report Bugs

- Description: The system must report all bugs to the third party.
- Rationale: The third party needs to document and fix these bugs in the code.
- Fit Criterion: This requirement is met when all discovered bugs have been fixed in the code. Note that this does not guarantee all bugs in the program.
- Acceptance: #20

14b Supportability Requirements

Customer Service

- Description: The developers of this system will be responsible for all support related inquiries.
- Rationale: Since the developers understand the inner workings of the system, they will be most effective in any support related details.
- Fit Criterion: This requirement is met when all patches, bugs, and vulnerabilities are fixed by the developers at the third party.
- Acceptance: #21

14c Adaptability Requirements

Multiple Operating Systems

- Description: The system must operate on Windows XP or later version and MacOS.
- Rationale: Different workers will use different systems based on the nature of the work. While Divvy delivery workers may use PCs, Operations Analysts may use Mac computers. Therefore, the system should work on these two operating systems.
- Fit Criterion: The system should be easily installed on Windows and Mac without any in-built security checkpoints preventing installation process.
- Acceptance: #22

14d Scalability Requirements

This requirement is ambiguous. One of the scalable requirements is for this system to handle other cities provided Divvy executives expand operations to other cities. Therefore, it is not inherently clear whether the system should be designed for other cities because Divvy does not conduct operations in other cities.

Since this algorithm only handles the city of Chicago, there is no need for any additional scalability for this system.

14e Longevity Requirements

Life Expectations:

- Description: The system must be able to maintain updates for as long as the Divvy corporation is in business.
- Rationale: In the event Divvy goes out of business, all data used by the software must be destroyed.
- Fit Criterion: This requirement is met by developing a contract with data storage cloud providers to ensure all data is destroyed if software is no longer in use. Cloud providers will take care of data destruction.
- Acceptance: #23

15 Security Requirements

15a Access Requirements

Database Access

- Description: All information on Divvy databases is considered confidential. Only authorized individuals may modify the data in the database.
- Rationale: Operations Analysts will use the data to determine the success of the system. Incorrect data could lead to improper business decisions.
- Fit Criterion: This criteria is met when all database integrity constraints are created by Database Administrators. When these integrity constraints are all created, only authorized access will be allowed.
- Acceptance: #24

15b Integrity Requirements

Database Constraints

- Description: All databases must be protected with integrity constraints.
- Rationale: This is to avoid the manipulation of confidential data.
- Fit Criterion: This requirement is met when a test of unauthorized access to the database is proven unsuccessful.
- Acceptance: #24

15c Privacy Requirements

Encryption of Accounts:

- Description: All Divvy workers will have their own accounts. All account information must be encrypted so third party maintenance teams do not access the unauthorized data.
- Rationale: This is designed to protect the sensitive information of account holders.
- Fit Criterion: The system must utilize standard encryption for all data pertaining to user accounts.
- Acceptance: #25

Data Brokerages:

- Description: The third party company must not provide access to the user data by a data brokerage.
- Rationale: This is designed to maintain user privacy.
- Fit Criterion: The third party company must not maintain a contract with any data brokerage.
- Acceptance: #26

15d Audit Requirements

This requirement does not apply. The system is not dealing with confidential medical, financial, or educational data.

15e Immunity Requirements

NIST Standards

- Description: The program must be use security standards defined by the National Institute of Standards of Technology.
- Rationale: This is to ensure the security of the data.
- Fit Criterion: This requirement is met when Windows Threat Defender and Mac OS Threat Defender do not detect any security loopholes in the system.
- Acceptance: #27

Database Constraints

- Description: The databases that communicate with the system must have integrity constraints imposed.
- Rationale: This is to protect the integrity of the database.
- Fit Criterion: This requirement is met unauthorized individuals who try to access the database are unable to change the data and are cited for database violations.
- Acceptance: #24, 28

16 Usability and Humanity Requirements

16a Ease of Use Requirements

Instructions

- Description: The instructions of the system must be clear enough where a person with at least a HS diploma can understand.
- Rationale: This requirement is to accommodate drivers.
- Fit Criterion: This requirement will be tested during the training phase. All drivers should understand the written instructions associated with this system based on hiring practices.
- Acceptance: #29

GUI Scenes

- Description: All buttons, texts, and text fields must be placed in identifiable locations on the UI Screen.
- Rationale: This requirement is to accommodate drivers.

- Fit Criterion: All drivers should be able to identify the buttons needed to do their tasks.
- Acceptance: #30

16b Personalization and Internationalization Requirements

Font Sizes

- Description: The user should be able to modify the size of the font to fit his or her needs.
- Rationale: This is to ensure that users do not receive any complaints regarding fonts.
- Fit Criterion: This requirement is met when there is a setting that users can use to modify the font and font size of the text.
- Acceptance: #31

16c Learning Requirements

Divvy is responsible for ensuring all drivers are able to read. Since the hiring process is managed by Divvy, this is a relevant assumption where no requirements are needed.

16d Understandability and Politeness Requirements

Instructions:

- Description: The wording of the instructions must not be offensive to users in any way.
- Rationale: This to ensure that the employee code of conduct is not violated.
- Fit Criterion: The ethics and HR officers at Divvy will review the wording of the instructions before the release of the system.
- Acceptance: #29

16e Accessibility Requirements

Not Applicable. Divvy is responsible for ensuring that drivers are flexible with computer technology. This is managed through the hiring process for drivers.

16f User Documentation Requirements

All Divvy drivers and frontline supervisors will be required to attend trainings led by developers after the first release of this system. Drivers and supervisors will receive separate trainings for their given interfaces of the system. New driver hires will be given one-on-one training by supervisors.

16e Training Requirements

All training for this system will be provided by the lead developers of this project. Lead developers will oversee a classroom format training for drivers and supervisors. Operations Analysts and Executives will not receive training for this system. Instead, supervisors will train Operations Analysts and Executives on how to receive reports from the system. Supervisors will attend a different training session than drivers as their use of the system will be different from that of drivers.

17 Look and Feel Requirements

17a Appearance Requirements

Divvy Appearance

- Description: The system must utilize a graphics that correspond with the Divvy website.
- Rationale: The system must be consistent with all Divvy software systems.
- Fit Criterion: This will be determined by a team of graphic designers. The success of this requirement will be judged based on how relatable it is to the Divvy website.
- Acceptance: #32

17b Style Requirements

Appropriate for All Backgrounds

- Description: The system must be friendly and accessible to all Divvy delivery drivers.
- Rationale: Divvy delivery drivers should be able to use the system without any interpretation issues.
- Fit Criterion: A high number of complaints from Divvy delivery workers means that the system is not user friendly. Zero or few complaints will determine that workers are able to adapt to the look and feel of the system.
- Acceptance: #33

18 Operational and Environmental Requirements

18a Expected Physical Environment

Onboard WiFi

- Description: The system must be able to connect to onboard WiFi routers installed in each of the Divvy vehicles.

- Rationale: The Divvy drivers need to have network connection when determining efficient routes between stations.
- Fit Criterion: This requirement is met when the system is able to connect to a router planted on the vehicle.
- Acceptance: #34

18b Requirements for Interfacing with Adjacent Systems

Databases:

- Description: The system must be able to input and output data from existing Divvy databases.
- Rationale: A lot of the data being used to analyze the system will be placed onto databases.
- Fit Criterion: This requirement is met when the system does not enter an erroneous state when communicating with the databases.
- Acceptance: #24

Maps:

- Description: The system will integrate a Map API for determining the routes.
- Rationale: The system will develop the most efficient route for Divvy drivers. Therefore, a map is key in providing assistance for routing.
- Fit Criterion: This requirement is met when the system is able to fully integrate a map API without any errors or documented issues.
- Acceptance: #7

18c Productization Requirements

Distribution:

- Description: The product shall be installed on Divvy's system via the System Administrator.
- Rationale: The Systems Administrators has control to all of Divvy's networks. Therefore, the product will be installed on all Divvy machines administered by the network.
- Fit Criterion: This requirement is successful when the system is installed on all necessary Divvy machines.
- Acceptance: #35

18d Release Requirements

Instructions:

- Description: The system must have an instruction manual for Divvy delivery workers and frontline supervisors.
- Rationale: Drivers and Supervisors will have slightly different interfaces. Therefore, drivers and supervisors need to understand how to implement slightly different systems.
- Fit Criterion: This requirement is met when all Divvy delivery workers and frontline supervisors are able to comprehend the instructions.
- Acceptance: #29

19 Cultural and Political Requirements

19a Cultural Requirements

English Release:

- Description: All language and words used in this product will be in the English language only.
- Rationale: The official language in the city of Chicago is English.
- Fit Criterion: There must be no words of other languages used in the wording of this program. We can assume that all drivers employed with Divvy know how to speak English based on Divvy's hiring procedures.
- Acceptance: #36

Wording of Instructions:

- Description: The wording of the instructions must not offend any race, religion, ethnic group, gender, or gender identity.
- Rationale: This is to avoid major lawsuits.
- Fit Criterion: All wording is to be checked by an attorney or qualified employee to ensure the language is not offensive in any way. Since this is system is used only in Chicago, the norms and cultures of other countries do not need to be taken into account.
- Acceptance: #29

19b Political Requirements

Location Bias:

- Description: The system must ensure that all Divvy stations are accounted for when transfers are conducted.
- Rationale: This is to ensure fairness and accuracy to all communities.

- Fit Criterion: This is to be measured when all stations have experienced at least one transfer since the implementation of this system.
- Acceptance: #37

20 Legal Requirements

20a Compliance Requirements

Product Distribution:

- Description: The system is to be used for the Divvy corporation only.
- Rationale: The contract specifies that no other form of distribution is allowed.
- Fit Criterion: The system development team must install the system on all machines used by the Divvy corporation. A team of systems administrators will be responsible for this task.
- Acceptance: #35

Traffic Laws:

- Description: The system must take into account all traffic laws, road blockages, and construction when determining routes.
- Rationale: This is to ensure the safety of all drivers and to avoid lawsuits from the city of Chicago.
- Fit Criterion: While there is no way to measure the success of this criteria, a failure will happen a Divvy worker is caught by a security camera or pulled over by a police officer. A team will investigate and will determine whether the incident was the fault of the system or the driver.
- Acceptance: #38

20b Standards Requirements

Laptop Security:

- Description: The system must be able to pass all security standards imposed by the National Institute of Standards and Technology.
- Rationale: The primary rationale is to avoid system vulnerabilities where hackers can gain access to Divvy's network.
- Fit Criterion: Systems administrators from the third party will be responsible for ensuring compliance based on NIST standards.
- Acceptance: #39

21 Requirements and Acceptance Tests

21a Requirements - Test Correspondence Summary

Requirement	Test
Determine Efficient Route	1
Determine Number of Bikes to Transfer	2
Report Faulty Bikes	3
Provide Reports	4
Receive Bikes	5
Station Count	6
Map Data	7
Delivery Times	8
Loading the Application	9
Data to Server	10
Route Correctness	11
Save Work	12
Requests	13
Bandwidths	14
Backups	15
Shifts	16
Patches	17
Internet Connection Failure	18
Addiction	19
Report Bugs	20
Customer Service	21

Multiple OS	22
Life Expectations	23
Database Access	24
Database Constraints	24
Encryption of Accounts	25
Data Brokerages	26
NIST Standards	27
Database Constraints	24
Instructions	29
GUI Scenes	30
Font Sizes	31
Divvy appearance	32
Age Appropriate	33
Onboard WiFi	34
Distribution	35
English Release	36
Wording of Instructions	29
Location Bias	37
Traffic Laws	38
Laptop Security	39

21b Acceptance Test Description

1. **Determine Efficient Route** = test whether the system is able to determine the most effective route from one station to another while trying to avoid typical traffic challenges
2. **Determine Number of Bikes to Transfer** = test whether the system is able to generate an appropriate number of bikes to be transferred from one station to another
3. **Report Faulty Bikes** = test whether the system is able to detect any of the inventory as malfunctioning
4. **Provide Reports** = test whether the system is able to generate a report with the given data requested
5. **Receive New Bikes** = test whether the system is able to increase its bike count
6. **Station Count** = test whether the system can access the number of bikes each station has in real time
7. **Map Data** = test whether the system can accurately display and correlate the map data to the appropriate scenario
8. **Delivery Times** = test whether the systems can accurately display and correlate the delivery times to the user in real time
9. **Loading the Application** = test whether the system can properly load the GUI and/or data
10. **Data to Server** = test whether the system can properly send the data back to the main server
11. **Route Correctness** = test whether the system is able to generate a correct route without violating any traffic laws (Ex. not going the opposite direction on a one way street)
12. **Save Work** = test whether the system is able to save all the transfers that drivers have made during their shifts
13. **Requests** = test whether the system can handle a certain number of requests in a set amount of time
14. **Bandwidth** = test whether the system can respect its bandwidth limitations per month
15. **Backups** = test whether the system is able to provide proper backups to prevent data loss
16. **Shifts** = test whether the systems can operate within work hours and even during probable overtime shifts
17. **Patches** = test whether the system can provide functionality whilst the system is updating/patching itself

18. **Internet Failure** = test whether the system can support offline functionality in case of internet failure or interruptions
19. **Addictions** = test whether the system prevents the user from any on-screen distraction, which may cause serious accidents, or even possibly injuries
20. **Report Bugs** = test whether the system can detect bugs within itself and be reported to third-party support
21. **Customer Service** = test whether the system can handle any and all customer service related inquiries
22. **Multiple OS** = test whether the system can work on Windows and MacOS
23. **Life Expectations** = test whether the system can be maintained during the life cycle of the system and/or the company behind the Divvy system
24. **Database Access** = test whether the system provides access the user, with varying levels of security based on the user's position within Divvy
25. **Encryption** = test whether the system can encrypt sensitive data to prevent unauthorized visibility
26. **Data Brokerages** = test whether the system protects user and customer data from third-party brokerage companies who seek to sell such personal data
27. **NIST Standards** = test whether the source code and the overall security of the software meets security standards imposed by NIST
28. **Triggers** = test whether triggers set on the databases are effective in preventing unauthorized access to database
29. **Instructions** = test whether instructions are clear and concise and not offensive in any way
30. **GUI Scenes** = test whether the GUI scenes are appropriate for users
31. **Font Sizes** =test whether the system can display text-based data in a font size comfortable to the user, and to adjust it if not
32. **Divvy Appearance** = test whether the Divvy logo can be used on the interface of the desktop app
33. **All Backgrounds** = test whether the system is easy to use for drivers of all backgrounds
34. **Onboard WiFi** = test whether the system's WiFi is consistent while traveling on-vehicle
35. **Distribution** = test whether the system can be distributed by means of installing the system through Systems Administrator
36. **English** = test whether the system can be understood by the employee with experience in basic high school-level English (the minimum qualifications to be an employee)

37. **Location Bias** = test whether the system will take all stations into consideration when determining where to distribute bikes
38. **Traffic Laws** = test whether the system's functionality respects and obeys the traffic laws imposed by the city government
39. **Laptop Security** = test whether the laptop has proper security software to prevent any malware attacks on the desktop application

III Design

22 Design Goals

The main system will be integrating a map based API onto the system. Therefore, it is essential that the system integrate the map as efficiently as possible without creating any lag. Since drivers must make transfers within certain hours, speed of the map and the speed of the algorithm processing time is essential.

The system must also not occupy more than 2 GB of memory. Given that the system is data intensive, it is essential that the system not occupy too much memory where it slows down other operations of the computer.

The system must have an easy User Interface for the user. Given that the minimum education attainment for a delivery driver is a high school diploma, the system must ensure that a driver with the relevant education is able to understand the UI of the application.

23 Current System Design

At this time, there is no current system design.

24 Proposed System Design

24a Initial System Analysis and Class Classification

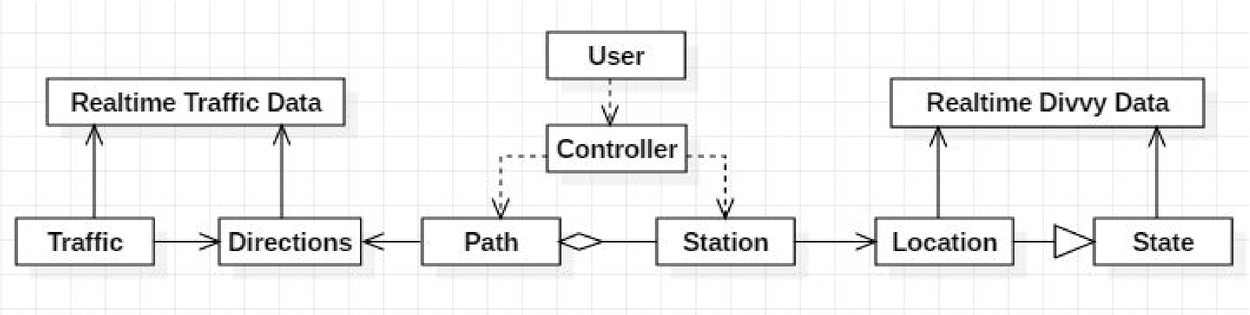


Figure 1.0: Class Diagram

24b Dynamic Modelling of Use Cases

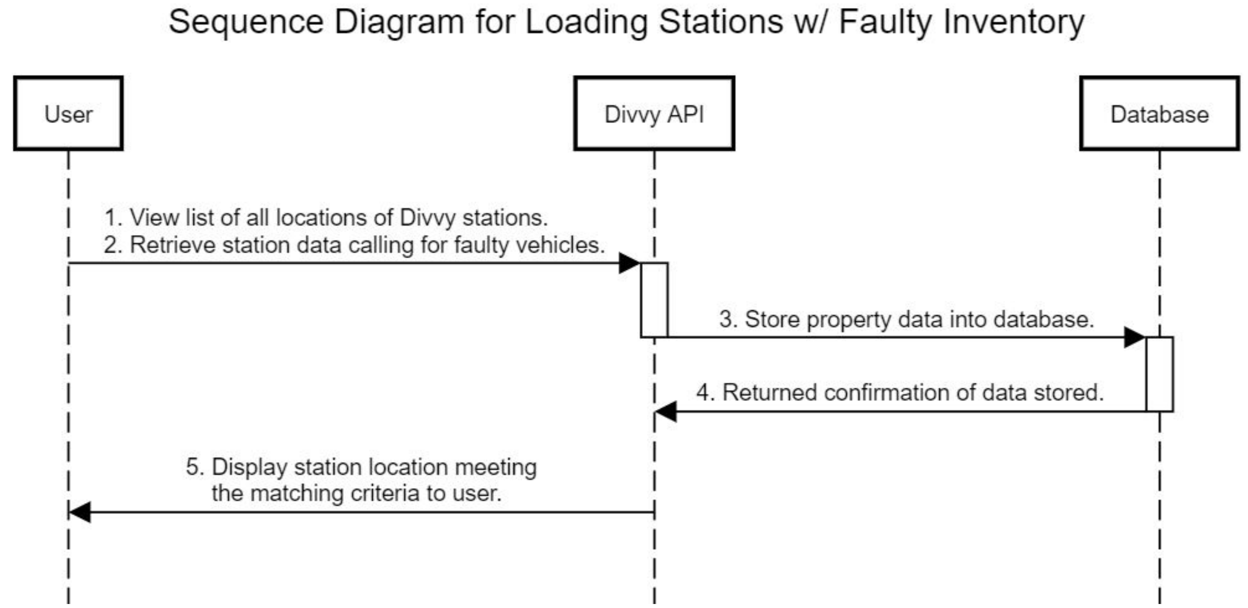

Sequence Diagram for Loading Stations w/ Faulty Inventory
User
Divvy API
Database
1. View list of all locations of Divvy stations.
2. Retrieve station data calling for faulty vehicles.
3. Store property data into database.
4. Returned confirmation of data stored.
5. Display station location meeting the matching criteria to user.

Figure 1.1: Sequence Diagram for Loading Stations w/ Faulty Inventory

24c Proposed System Architecture

The proposed system architecture is a Model-View-Controller system. The view for the user will consist of an interface where drivers will receive real time data on where to transfer bikes and the most efficient route for transfer. The Model will be the data of how many bikes are stored at each station in real time. The Controller will utilize the algorithm for determining the appropriate number of bikes to transfer. The Controller will be responsible for updating the view.

24d Initial Subsystem Decomposition

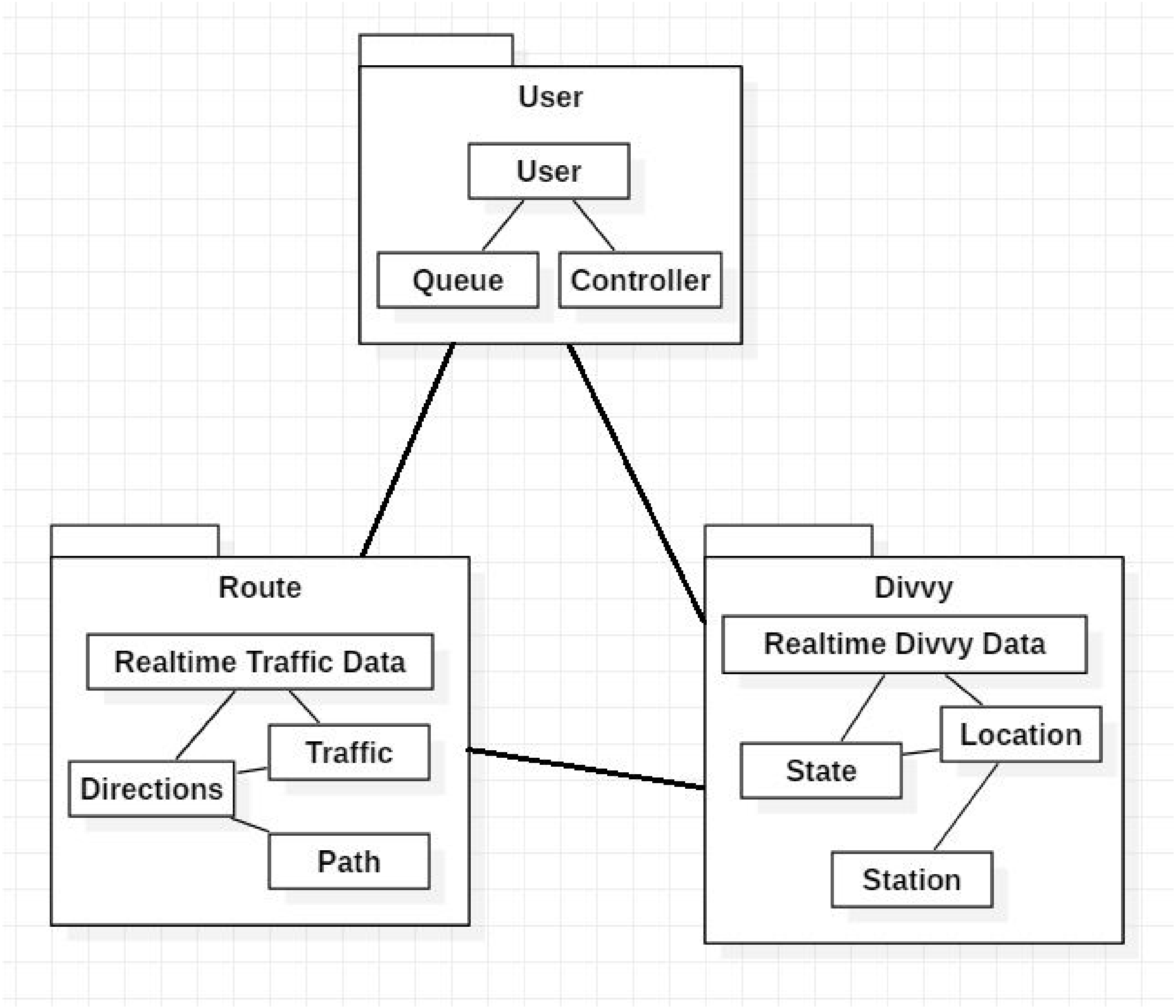


Figure 1.2: Subsystems Diagram

- The User subsystem will provide the main controller user interface and display the work queue of the employee, if their work position requires such (i.e. users that are in managerial positions do not receive their own queue). The data displayed within the UI will update as its interconnected subsystems do.

- The Route subsystem will provide the implementation of the algorithm to generate the most optimal route for Divvy drivers. The algorithm will collate data of real time traffic patterns and directional outputs to supply the best result.

- The Divvy subsystem tracks current status of all stationed vehicle inventory, especially those which are defective, with the dependency of real time data provided by the Divvy database.

25 Additional Design Considerations

25a Hardware/Software Mapping

Since the system is reliant on map data, a physical server will be needed for the map. Physical databases are also needed to store data related to commute times. Therefore, the system must ensure proper connection between databases and servers.

25b Persistent Data Management

All data relevant to the operation of the system will be stored on databases or on portable document files maintained by the system. The data stored on databases will be related to commute times between stations as well as the number of bikes at each station. The data in PDF format will be related to productivity.

25c Access Control and Security

All access points for Divvy delivery drivers must be accessible to both delivery drivers and frontline supervisors. However, the system operations for frontline supervisors with regards to analyzing productivity should not be made accessible to drivers. The same goes for operations analysts and executive management.

25d Global Software Control

Like stated, any information that is not being sent to the user. Like in Figure 1.0, the Real Time Map Data class should throw exception to the other classes using this class. This way the user should be notified.

25e Boundary Conditions

A system must be in place to maintain databases and servers that hold the information relevant to the system. Since the system is using databases already in place by Divvy, a system that maintains these databases should already be in place.

25f User Interface

The user interface will resemble that of the Divvy website. The UI should contain logos of Divvy throughout the screen while the main feature should be a map along with text fields and buttons for inputting the bikes and station location.

25g Application of Design Patterns

The main design patterns used in this system will be Observer, Decorator, Facade, and Template. Some of the other design patterns will be utilized. The Observer design pattern will be used heavily given that the system architecture is an MVC pattern. The Decorator pattern will be used to extend the functionality of the system. The Facade will be used to maintain the collect of interfaces that the user will see while Template will hold the skeleton for the algorithm.

26 Final System Design

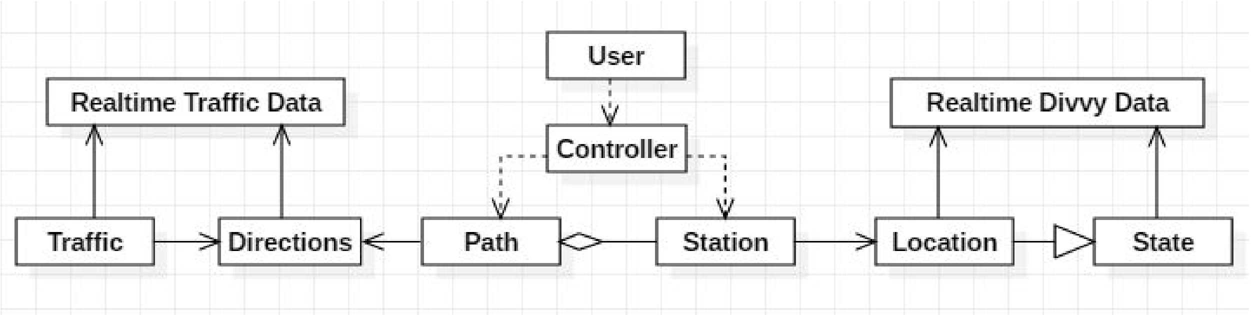


Figure 1.0: Class Diagram

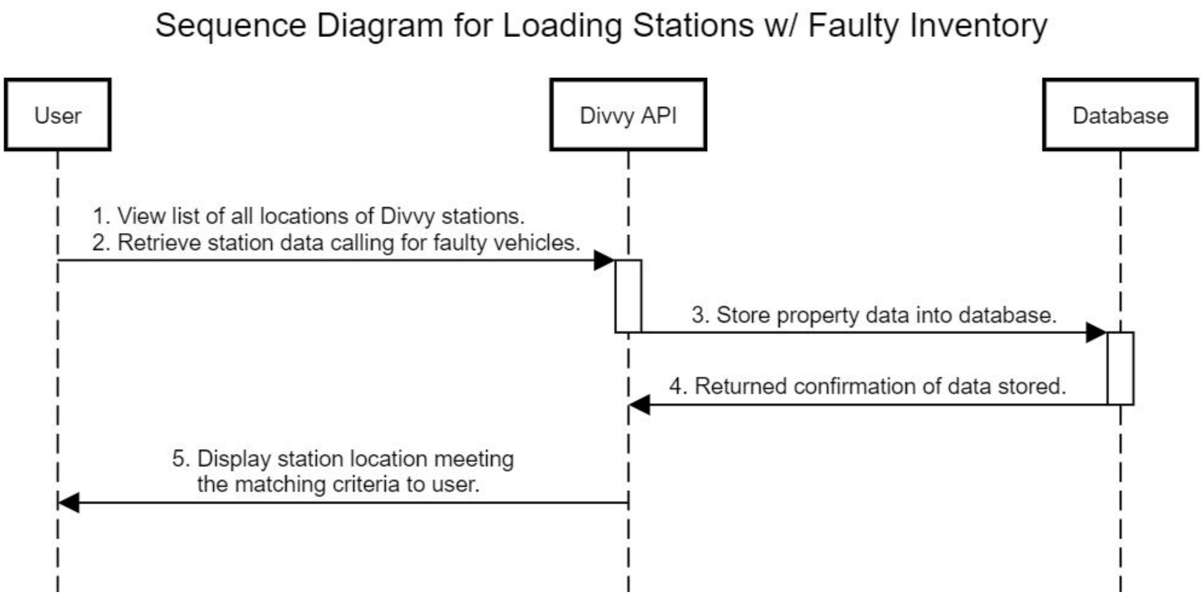

Sequence Diagram for Loading Stations w/ Faulty Inventory
User
Divvy API
Database
1. View list of all locations of Divvy stations.
2. Retrieve station data calling for faulty vehicles.
3. Store property data into database.
4. Returned confirmation of data stored.
5. Display station location meeting the matching criteria to user.

Figure 1.1: Sequence Diagram for Loading Stations w/ Faulty Inventory

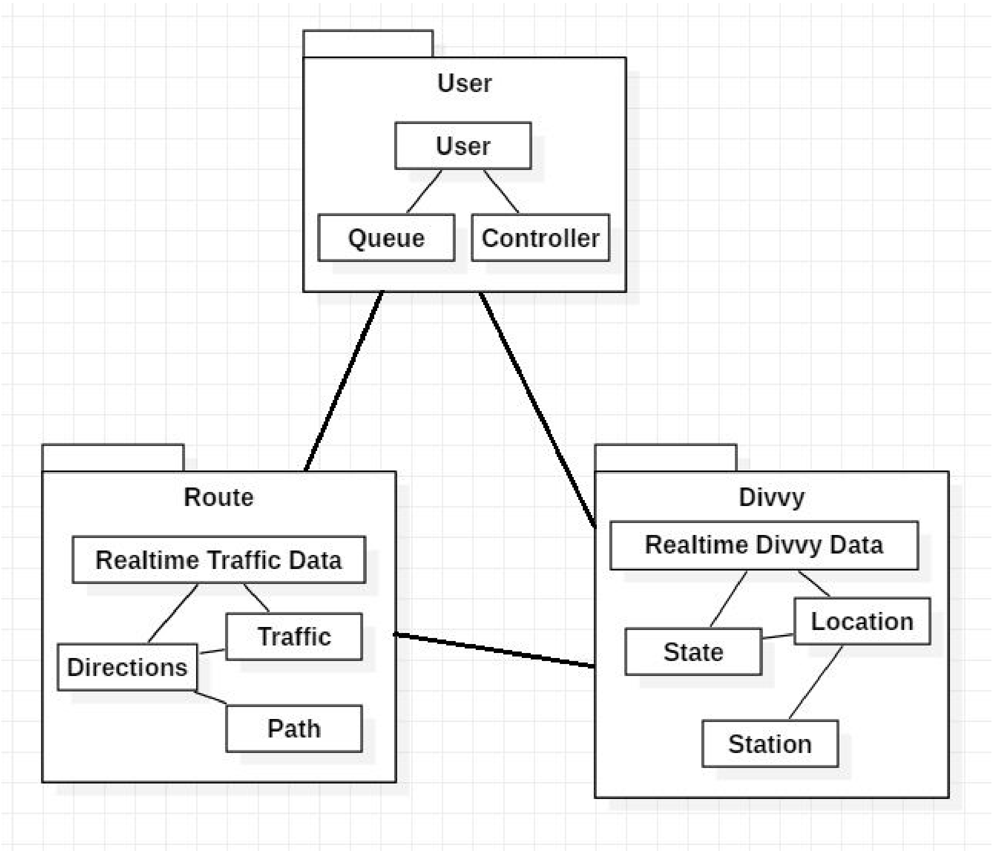


Figure 1.2: Subsystem Diagram

27 Object Design

27a Packages

This system will require the use of packages, including but not limited to realtime data of current traffic patterns and current status of stationed Divvy inventory, state of defective inventory, directions, locations of stations, specific station data, work queues of drivers, etc.

27b Subsystem I

The User subsystem is for user interaction.

27c Subsystem II

The Route subsystem is for optimal route generation for Divvy Van drivers.

27d Subsystem III

The Divvy subsystem is for keeping realtime records of currently stationed vehicles.

27e Subsystem ETC

Some other subsystem may include a management subsystem in keeping records of employees in cases of promotion, reallocation, or termination.

IV Project Issues

28 Open Issues

- How does the system account for road closures? If a Divvy station is located on the street of a closed construction site, how will the algorithm know to ignore that station for the time being?
- How does the system reconfigure to account for bikes that are not returned properly?
- How does the data produced by the system properly integrate and account for the revenue, expenses, and profit of the system?
- How does the system account for weather delays, traffic disruptions, and other issues related to routing?

29 Off-the-Shelf Solutions

While there are some off the shelf solutions available, there are no systems that can be integrated with Divvy. The map based API is the only commercial system that can be used within the system.

29a Ready Made Products

The main ready made product that can be integrated into this system is Google Maps real time data indicating the number of Divvy Bikes at each station. In July 2019, Google integrated a new feature in Maps which allows a user to see how many bikes are at each station in real time. Since the system is reliant on this data, the Google Maps API can be integrated with the system.

29b Reusable Components

The system will be developed in a programming language that can easily integrate a map based API. A license will be needed for the use of the programming language.

29c Products That Can Be Copied

While there are products that have a similar function to that of the system to be developed, there are no other products that can be copied due to trademark laws. These companies have placed stringent patents on their products; therefore, all code will have to be written from scratch.

30 New Problems

30a Effects on the Current Environment

Implementing the system will not have serious implications for the users of this product. Divvy delivery drivers already use a laptop during their shifts. Operations analysts, frontline supervisors, and executive managers will not need to dramatically implement changes to accomodate the system.

30b Effects on the Installed System

There will be little to no change on effects on the installed system. Since this a product built from scratch, the hardware and software of the system should not see any significant change.

30c Potential User Problems

While users will definitely benefit from the system, a problem that can cause trouble for users is the issue of slow updates. Since the system is data intensive because of the utilization of maps, many of the updates that users receive could slow the overall system.

30d Limitations

At this time, there are no limitations of this product. The one limitation that could arise is the fact that the data is limited to just the city of Chicago. However, the system only needs to worry about expansion provided a business decision is made to expand to other cities.

30e Follow-Up Problems

A problem that could potentially arise is the opening and closing of Divvy stations. Divvy chooses to close certain stations due to low checkout rates while opening other stations in more popular areas. This could potentially be a problem for the system if it cannot account for stations openings and closures.

31 Migration to New Product

31a Requirements for Migration

There is no need for migration.

31b Data That Has to be Modified for Transferred for New System

There is no data that has to be transferred. The new system will connect to Divvy's current databases and servers.

32 Risks

- Lack of proper development team.
- Project goes over budget.
- WiFi does not properly work on all Divvy vehicles.
- Driver issues; (alcoholism, fatigue)

33 Costs

The costs for the solutions will be roughly fixed. Development of the system should take at most one full year. It is estimated that most of the time should be spent on developers building algorithms for efficient transit and determining the number of bikes to be transferred. Most of the data and data models used in this system have already been established and therefore do not need any further improvement.

Divvy will not need to purchase additional hardware for the devices. These devices will be operation on laptop computers already carried by Divvy delivery workers. The main costs in development will be to hire developers focused on writing and perfecting the algorithm that meets the desired goals and requirements. In addition, there will also be ongoing maintenance costs for the back end maintenance of the system by the third party.

34 Waiting Room

- Have an algorithm account for specific disruptions in traffic patterns: ie, marathon, accident, natural disaster
- Account for station openings and closures
- Receive station demographics of the customers checking out from each station

35 Ideas for Solution

A potential solution to the project will be to allow the user to manually input the stations. This could pose some risks if the user were to make a mistake with the inputting of the station. However, this solution could account for the opening and closing of stations.

36 Project Retrospective

Overall, writing the requirements for this system went fairly well. It was known exactly what requirements were needed for the system as well as the type of environment needed. Developing proper design goals was difficult as there wasn't much idea of how to implement proposed goals. Certain design goals in the beginning were discarded due to lack of ideas on implementation.

V References/Bibliography

[1] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

VI Index