

Construction Equipment Delivery Program



An Application of the Traveling Salesman and Knapsack Problems

**Group 14
CS 440
at the
University of Illinois Chicago**

September 2019

Table of Contents

I. Project Description	7
1. Project Overview	7
2. The Purpose of the Project	7
2.a. The User Business or Background of the Project Effort	7
2.b. Goals of the Project	7
2.c. Measurement	8
3. The Scope of the Work	8
3.a. The Current Situation	8
3.b. The Context of the Work	8
3.c. Work Partitioning	9
3.d. Competing Products	10
4. The Scope of the Product	10
4.a. Scenario Diagram(s)	11
4.b. Product Scenario List	11
4.c. Individual Product Scenarios	11
5. Stakeholders	12
5.a. The Client	12
5.b. The Customer	12
5.c. Hands-On Users of the Product	12
5.d. Maintenance Users and Service Technicians	12
5.e. Other Stakeholders	12
5.f. User Participation	13
5.g. Priorities Assigned to Users	13
6. Mandated Constraints	13
6.a. Solution Constraints	13
6.b. Implementation Environment of the Current System	14
6.c. Partner or Collaborative Applications	14
6.d. Off-the-Shelf Software	14
6.e. Anticipated Workplace Environment	14
6.f. Schedule Constraints	14
6.g. Budget Constraints	15
7. Naming Conventions and Definitions	15
7.a. Definitions of Key Terms	15
7.b. UML and Other Notation Used in This Document	15

7.c.Data Dictionary for Any Included Models	17
8. Relevant Facts and Assumptions	18
8.a.Facts	18
8.b.Assumptions	18
II. Requirements	18
9. Product Use Cases	19
9.a.Use Case Diagrams	19
9.b.Product Use Case List	19
9.c.Individual Product Use Cases	19
10.Functional Requirements	26
11.Data Requirements	27
12.Performance Requirements	28
12.a. Speed and Latency Requirements	28
12.b. Precision or Accuracy Requirements	29
12.c. Capacity Requirements	30
13.Dependability Requirements	30
13.a. Reliability Requirements	30
13.b. Availability Requirements	31
13.c. Robustness or Fault-Tolerance Requirements	31
13.d. Safety-Critical Requirements	31
14.Maintainability and Supportability Requirements	32
14.a. Maintenance Requirements	32
14.b. Supportability Requirements	32
14.c. Adaptability Requirements	33
14.d. Scalability or Extensibility Requirements	33
14.e. Longevity Requirements	34
15.Security Requirements	34
15.a. Access Requirements	34
15.b. Integrity Requirements	35
15.c. Privacy Requirements	36
15.d. Audit Requirements	36
15.e. Immunity Requirements	37
16.Usability and Humanity Requirements	38
16.a. Ease of Use Requirements	38

16.b.	Personalization and Internationalization Requirements	38
16.c.	Learning Requirements	39
16.d.	Understandability and Politeness Requirements	40
16.e.	Accessibility Requirements	40
16.f.	User Documentation Requirements	41
16.g.	Training Requirements	41
17.	Look and Feel Requirements	42
17.a.	Appearance Requirements	42
17.b.	Style Requirements	42
18.	Operational and Environmental Requirements	43
18.a.	Expected Physical Environment	43
18.b.	Requirements for Interfacing with Adjacent Systems	44
18.c.	Productization Requirements	44
18.d.	Release Requirements	44
19.	Cultural and Political Requirements	45
19.a.	Cultural Requirements	45
19.b.	Political Requirements	45
20.	Legal Requirements	46
20.a.	Compliance Requirements	46
20.b.	Standards Requirements	56
21.	Requirements Acceptance Tests	46
21.a.	Requirements – Test Correspondence Summary	46
21.b.	Acceptance Test Descriptions	49
II.	Design	53
22.	Design Goals	53
23.	Current System Design	53
24.	Proposed System Design	54
24.a.	Initial System Analysis and Class Identification	54
24.b.	Dynamic Modelling of Use-Cases	55
24.c.	Proposed System Architecture	57
24.d.	Initial Subsystem Decomposition	58
25.	Additional Design Considerations	59
25.a.	Hardware / Software Mapping	59
25.b.	Persistent Data Management	59

25.c	Access Control and Security	59
25.d	Global Software Control	60
25.e	Boundary Conditions	60
25.f	User Interface	60
25.g	Application of Design Patterns	61
26.	Final System Design	62
27.	Object Design	62
27.a	Packages	62
27.b	Subsystem I	62
27.c	Subsystem II	63
27.d	Subsystem III	63
III.	Project Issues	64
28.	Open Issues	64
29.	Off-the-Shelf Solutions	64
29.a	Ready-Made Products	64
29.b	Reusable Components	65
29.c	Products That Can Be Copied	65
30	New Problems	65
30.a	Effects on the Current Environment	65
30.b	Effects on the Installed Systems	65
30.c	Potential User Problems	66
30.d	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	66
30.c	Follow-Up Problems	66
31	Migration to the New Product	67
31.a	Requirements for Migration to the New Product	67
31.b	Data That Has to Be Modified or Translated for the New System	67
32	Risks	67
33	Costs	68
34	Waiting Room	69
35	Ideas for Solutions	69
36	Project Retrospective	70

IV. Glossary	70
V. References / Bibliography	71
Index	71

I.

Project Description

1. Project Overview

The Construction Equipment Delivery Program is a mobile application that helps construction companies know which trucks are accountable for each delivery as it will be responsible for planning the route of each truck delivering equipment and which warehouse it goes to gather more equipment. It can also be used so that the companies can spot any bottlenecks in their service like a lack of tools or trucks so that they can fine-tune these deficiencies. The route of each truck is determined by an algorithm in the application before the trucks mobilize towards sites. As each truck can only carry a certain amount of equipment, the algorithm must plan the route to effectively utilize each trucks carrying capacity. Items that the truck can carry have different priorities because the priority is based on how important it is to a project at the moment. They also have different weights because not everything weighs the same. Once the truck has dropped off the materials at all the sites it plans on visiting it must gather more materials to drop off at other construction sites based on the plan. It gathers materials at various warehouses this company that are designated by the app to maximize efficiency. At the end of the day it also tells the truck drivers the best path to collect all the equipment they dropped off

2. The Purpose of the Project

The application is being done to help construction companies and contractors who deal with the burden of transporting equipment and materials from warehouses to several construction sites by providing the best route and efficient cargo capacity possible.

2.a. The User Business or Background of the Project Effort

The construction business requires to build houses, buildings, fix or maintain bridges, routes or even perform construction underwater. The entire industry is responsible not only for construction development but also for internal logistics, project management, resource utilization, including those involving transportation, delivery time, cost optimization.

2.b. Goals of the Project

The project aims to improve the revenue and support cost savings by determining the best possible route to deliver equipment and tools to the various construction sites. The application must consider certain restrictions, such as the limited number of trucks, their

capacity, the time when the equipment is needed, etc. in order to manage resources effectively and saving money in the process.

2.c. Measurement

These are the main metrics we are taking into account to achieve our goals and at the same to contribute to the business to some extent.

- Minimize the delivery time of equipment to start with the construction projects.
- Number of trips made by the transportation trucks
- Minimize fuel consumption (achieved by the above)
- Reduce the waiting time of trucks which picks up materials
- The number of positive feedback from the project managers and truck drivers.

These will help us to measure the success of the software and whether we could positively impact business revenue.

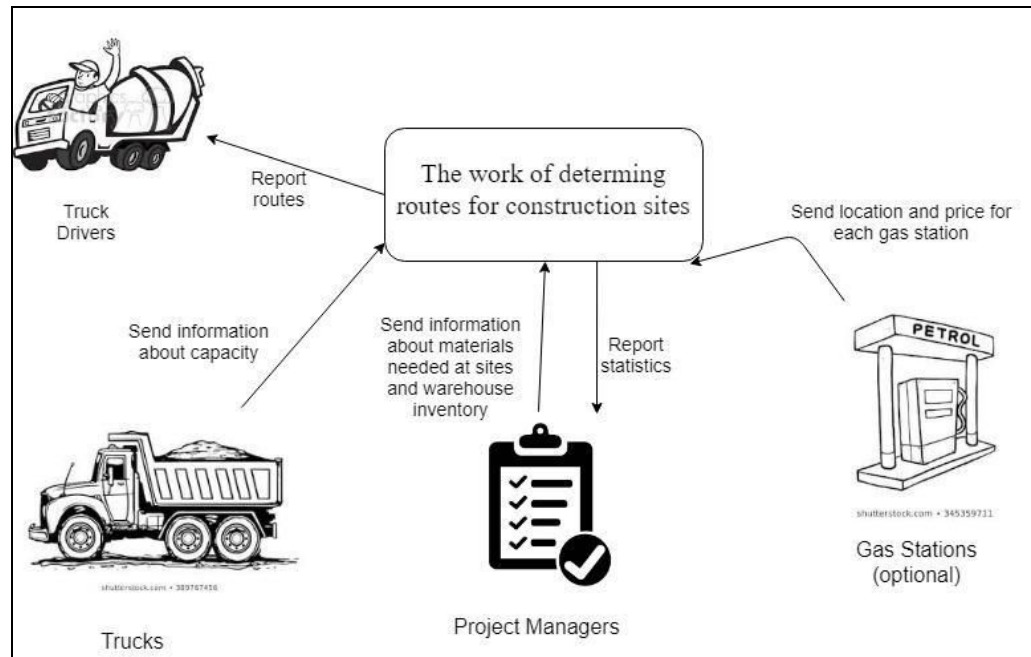
3. The Scope of the Work

The work consists of determining the best possible routes to drop off materials needed at and pick up leftover materials from various construction sites, optimizing the cost to drive to each site and the amount of materials and equipment carried in the truck(s).

3.a. The Current Situation

In order to ensure that the correct materials are at the correct construction site at the proper time, currently each construction project has a project manager. This individual takes into account the cost of materials from various sources, the availability of each of these materials needed for the project, inventory already accumulated by the company, what deadlines need to be met at which time, and many other factors affecting the progress of the construction work. The project manager then analyzes and assembles all this information to determine how the construction should proceed, including when materials need to be ordered by which day/time and to which site.

3.b. The Context of the Work



3.c. Work Partitioning

Event Name	Input and Output	Summary
Material and warehouse information added	Project Manager sends information about materials needed and warehouse inventory (in)	Record the information sent in and store it
Capacity information added	Truck (in)	Record the information about the capacity of each truck (i.e. its volume and maximum weight)
Gas station information added	Gas Station (in)	Record the information about the location and price at each of the gas stations near the construction sites or warehouses

Time to determine route	Truck Drivers (out)	Determine the best route for drop-off and pick-up and send this to the truck driver's smartphone or tablet attached to the truck's dashboard
Compile Statistics	Project Manager (out)	After recording information about the number of stops made, the inventory at each warehouse, and the distances between warehouses and construction sites, recommendations are made and sent to the project manager.

1.d. Competing Products

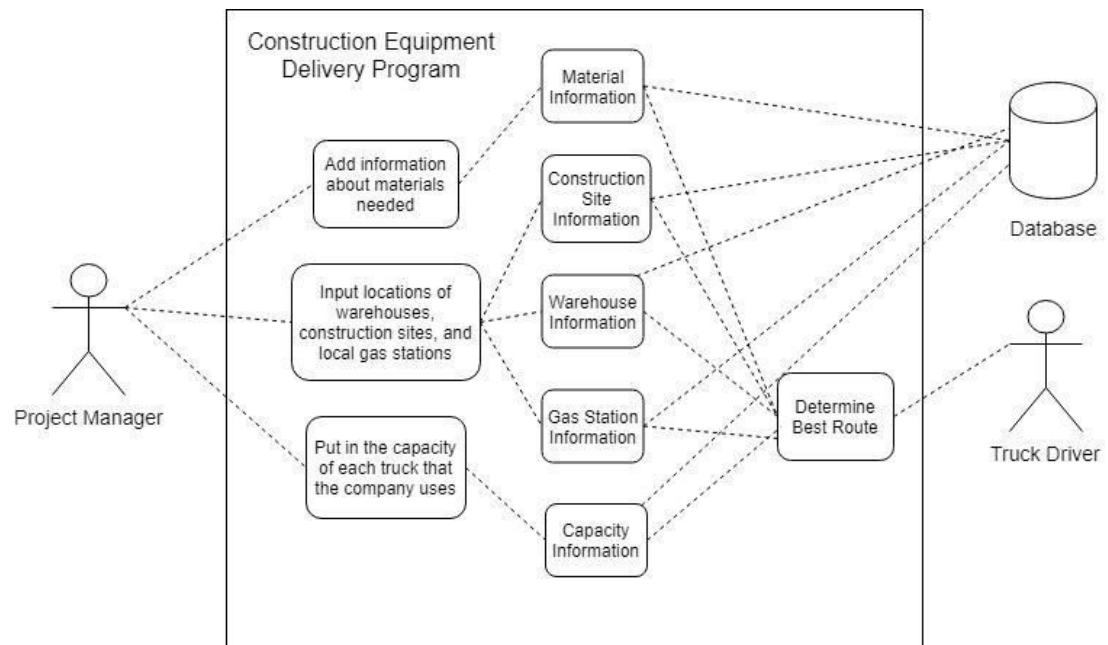
There are currently no competing products which work over the same exact scenario, but there are others in the market which address similar routing problem.

- Badger Maps - <https://www.badgermapping.com>
- Route XL - <https://www.routexl.com/>
- Route4me - <https://www.route4me.com/>
- Road Warrior - <https://www.roadwarrior.app/>
- Copilot GPS - <https://copilotgps.com/en-us/>

4. The Scope of the Product

The product will be used by construction companies and contractors to help determine best possible routes for dropping off and picking up materials. These routes will be based off of optimizing the capacity of the truck(s), the information given about the locations of warehouses and construction sites, the materials needed at each site, and minimizing the total distance travelled. It could also be used to determine if more trucks, materials, or warehouses at better locations need to be acquired.

4.a. Scenario Diagram(s)



4.b. Product Scenario List

- (1) Input Information About Construction Sites, Warehouses, and Gas Stations
- (2) Input Information About Materials Needed

4.c. Individual Product Scenarios

(1) Input Information About Construction Sites, Warehouses, and Gas Stations:

When first using the program, the project manager must put in the location of each construction site, each warehouse, and all local gas stations. After this first use, these may be edited or removed (for example, when they have finished working on a construction site) but initially, there needs to be some information that needs to be added.

- ##### (2) Input Information About Materials Needed:
- Before a route may be determined, the project manager will put into the program what materials are needed at each construction site, along with when these materials need to be delivered by, the quantity needed, and their dimensions and weight.

5. Stakeholders

5.a. The Client

Construction companies that way they can improve efficiency in their business. With the app all truck drivers will have their route so that there will be no overlap like there can be without a plan. Not only that but they can spot weaknesses in their route because they will see that trucks are either waiting for equipment to be finished or are all visiting the same. This will improve efficiency of the business because it will eliminate confusion about what employees in the field need to do.

5.b. The Customer

Construction companies are the customers as they are the only people who would benefit from the utility this app provides. Efficient route planning is possible with Google maps so I don't see why anyone else would need the other utilities provided by our service.

5.c. Hands-On Users of the Product

- **Username/Category:** Truck Drivers/ Construction Equipment Deliverers
- **User role:** Open the app to get the route they should follow for the day. It includes the equipment they need to pick up and where to pick it up.
- **Subject matter experience:** They know how to drive a truck. They may also know how to use the equipment.
- **Technological Experience:** Novice
- **Other user characteristics:** They work for a construction company. Not blind.

5.d. Maintenance Users and Service Technicians

The app's database would be maintained by clerical workers or support staff within the company. They are responsible for filling the database with information such as which locations need what equipment and where they can find that equipment.

5.e. Other Stakeholders

N/A

5.f. User Participation

We will expect user participation during the creation of the user interface in order to ensure this application has a way to navigate it efficiently. We need to verify they can understand the directions the application tells them to take.

5.g. Priorities Assigned to Users

All users are equally as important as they are all truck drivers in the eyes of the application. The only thing that is different about them is the route they have been assigned and the equipment they carry on the trucks.

6. Mandated Constraints

6.a. Solution Constraints

Criterion 1

Description: The Construction Delivery Program is meant to run on mobile, tablet, and desktop.

Rationale: Construction delivery companies need a computer to send directions to the driver. The driver needs a mobile device or tablet to receive directions to get to the construction site. The construction delivery program should be able to both send and receive directions from the origin to the destination.

Fit Criterion: The software should be an API backend along with multiple frontends on each platform.

Criterion 2

Description: The Construction Delivery Program will be developed as a web app.

Rationale: The program needs a server to communicate between the driver and the origin. Rather than creating multiple applications, the creator should make a single API backend along with multiple frontends for each platform.

Fit Criterion: The software should be developed in a web framework like Ruby on Rails, Django, Spring Boot, etc.

Criterion 3

Description: The code should be easily modifiable and expandable.

Rationale: Different construction delivery companies have different delivery methods and different algorithms for getting the equipment to the construction site.

Fit Criterion: The software should be developed using Model View Controller principles and design patterns so that the code could be easily modified and expanded by addition or replacing delivery algorithms.

6.b. Implementation Environment of the Current System

The company shall receive a company login to the website to create Base users and Delivery users. The company can deactivate a user if they are no longer with the company. A Base user can dispatch multiple Delivery users to deliver the construction equipment to the construction site.

6.c. Partner or Collaborative Applications

The software team should be in contact at least one Construction Delivery company. To create a stable algorithm, they will need to ask the company what they look for in for delivery efficiency and fuel efficiency

6.d. Off-the-Shelf Software

RabbitMQ: A messaging service that should be used to assign jobs from the Base to the Delivery users. It reduces the need to

Mailer Service: A service like Rails Mailer, Spring Email, etc. can be used to notify the Delivery user for new jobs and should be used to send activation links to new users

6.e. Anticipated Workplace Environment

Every driver should have a phone at the very least and could have a tablet or computer available. Dispatchers are generally located in an office and should have a full computer available. The Construction Delivery software should be able to run on all of those devices.

6.f. Schedule Constraints

The original product will take time to develop completely, but once the delivery algorithm is finished, it will be easy to create users and agencies around the Construction Delivery companies and their needs and have a prototype to work for a single company only.

6.g. Budget Constraints

The budget should satisfy the salaries of employees for the firm. The software should stick to the open source community to save on additional third-party purchases. The budget is relatively low for this project

7. Naming Conventions and Definitions

7.a. Definitions of Key Terms

Construction Site: Location where a construction project is planned or has started

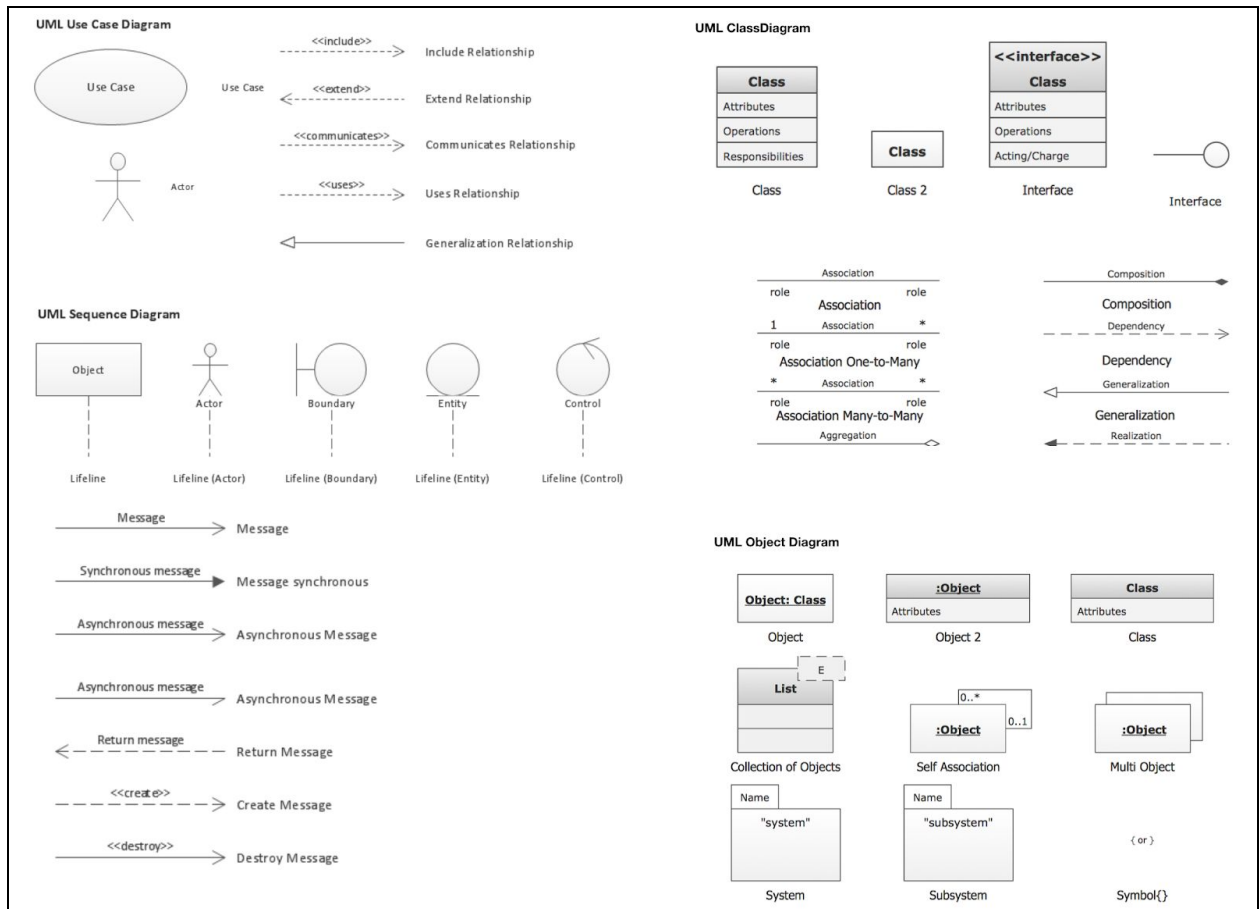
Equipment: Consider not only heavy equipment but also tools and materials needed to start with work on construction sites. Their information must include: weight, time is needed

Truck capacity: Metadata of the vehicle namely maximum weight, and volume

Route: Path the driver has to follow to get to the different construction sites and warehouses

Warehouse System: Some companies already warehouse management systems that provide information about the stock of equipment and offer interfaces to sync up with our Application.

7.b. UML and Other Notation Used in This Document



Ref. ConceptDraw : UML Diagrams.

UML Diagrams terms from "UML Distilled" by Martin Fowler.

Use cases: Capture the functional requirements of a system. Use cases work by describing the typical interactions between the users of a system and the system itself, providing a narrative of how a system is used.

Use Case Diagrams: Context diagram which shows the system boundary and the interactions with the outside world. The use case diagram shows the actors, the use cases, and the relationships between them.

Object Diagram: Snapshot of the objects in a system at a point in time. It can be used to show an example configuration of objects. It is useful when the possible connections between objects are complicated.

Class Diagram: Describes the types of objects in the system and the various kinds of static relationships that exist among them. Class diagrams also show the properties and operations of a class and the constraints that apply to the way objects are connected

Sequence Diagram: Describe how groups of objects collaborate in some behavior. It shows a number of example objects and the messages that are passed between these objects within the use case.

«include» dependency: Indicates that the path of one use case is included in another. Typically, this occurs when a few use cases share common steps. The included use case can factor out the common behavior.

«extend» dependency: Provides a more controlled form of extension than the generalization relationship. Here, the base use case declares a number of extension points. The extending use case can alter behavior only at those extension points

7.c. Data Dictionary for Any Included Models

Data Structures

Truck Capacity: Volume, maximum weight, available fuel

Construction site Object: Address, Project code, Equipment needed, working hours.

Route Object: List of construction site Objects, time schedule,

Equipment Object: fabric, weight, volume,

Data Property

Trucks Capacity: Regular Cargo capacity is between 1000-2000 lbs and volumes varies and maximum speed allowed is 60 mph.

Routes: Are not considering Traffic, nor time, just the distance of the journey

Itinerary: Hours in construction sites when the equipment is needed, if there are too many in the same hour, more trucks are required.

Warehouse: Some Routes involve a trip to the Warehouse. this also provides stock information, address and time

8. Relevant Facts and Assumptions

8.a. Facts

Some project managers need to train their drivers to use the platform. Some drivers use something as simple as Google Maps to get to their destination.

8.b. Assumptions

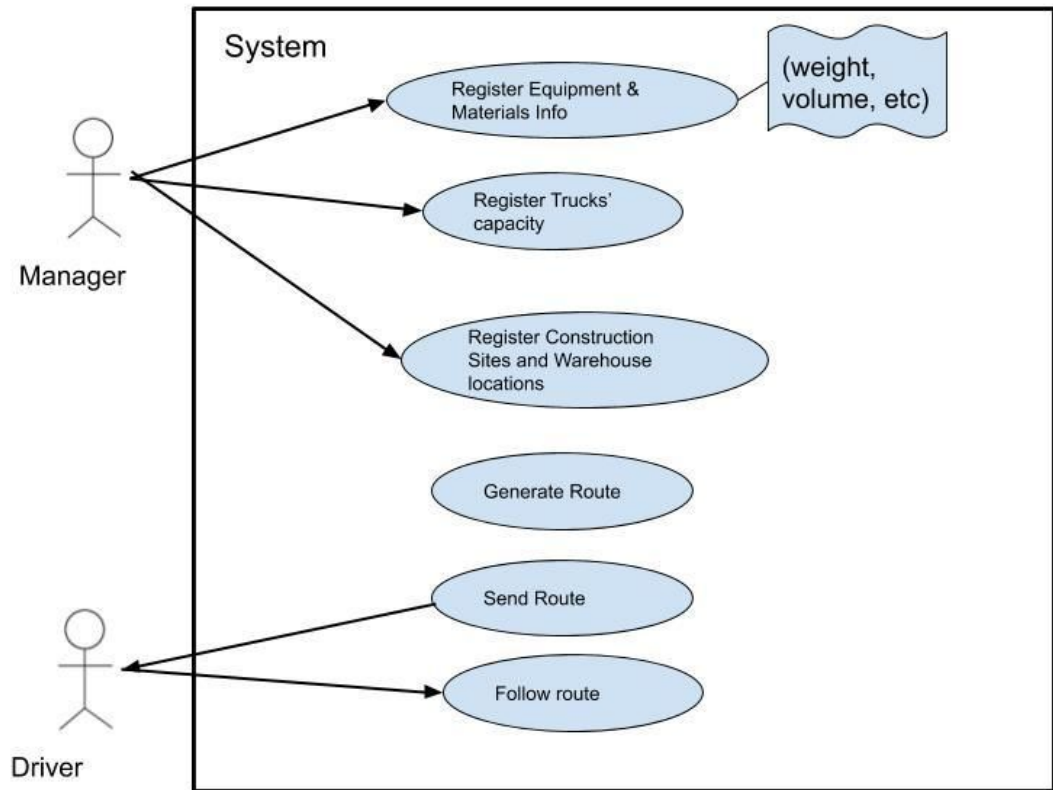
The project managers will need to receive training to use the platform. But they may or may not have prior knowledge in dispatching their drivers to the delivery site. Training of the software should occur after receiving training of dispatching drivers.

During the development, we will be in contact with a Project Manager from a Construction Delivery company to make sure the software is intuitive and can provide accurate and relevant information to both the dispatcher and driver.

II. Requirements

9. Product Use Cases

9.a. Use Case Diagrams



9.b. Product Use Case List

1. Register Construction Sites and Warehouse Locations
2. Register Equipment and Materials Info
3. Register Trucks' Capacity
4. Generate Route
5. Send Route
6. Follow route

9.c. Individual Product Use Cases

Use case ID: 0001
and Warehouse Locations

Name: Register Construction Sites

Pre-conditions: Manager trained how to operate the application and metadata ready to use in the registration process

Post-conditions: New construction site location registered as part of the new project.

Initiated by: Manager

Triggering Event: New construction project is initiated in the company and the delivery of equipment needs to be completed.

Additional Actors: N/A

Sequence of Events:

1. The Manager opens the application, and choose option to register a new location based on the construction project.
2. A Map is loaded in the screen centered in the city that was set by default previously.
3. The Manager selects a point in the map and assigns this to the new construction project.
4. The manager can then register the rest of the metadata for the construction project.
5. This process can be repeated sequentially for next construction site location that the company needs to register.

Alternatives: Optionally the Manager can enter the address as a reference, since normally new construction are in open space in a remote location where probably don have street/avenue or number

Exceptions: N/A

Use case ID: 0002
Materials Information

Name: Register Equipment and

Pre-conditions: Manager trained how to operate the application and metadata ready to use in the registration process

Post-conditions: New Equipment is registered to be used in any project.

Initiated by: Manager

Triggering Event: Company acquired new equipment

Additional Actors: N/A

Sequence of Events:

1. The Manager opens the application, and choose the option to register new equipment or materials based on the construction needs.
2. The form is displayed allowing the manager to register the type of equipment, the volume, the weight, dimensions, quantity and current warehouse where it is located.
3. The manager can select from a list of Warehouses previously registered.
4. Then, the manager can choose the assign which one is assigned to an existing project or can do this assignment later. In the Update Equipment Use Case
5. This process can be repeated sequentially for the next Equipment that the company acquires.

Alternatives: In the event that the warehouse location is not listed, it needs to be created, this step is in the Use Case for Create Warehouse.

Exceptions: N/A

Use case ID: 0003

Name: Register Trucks' Capacity

Pre-conditions: User is driving

Post-conditions: Move on to next action

Initiated by: Application

Triggering Event: Application opened

Additional Actors: N/A

Sequence of Events:

1. The application is opened and tells truck driver how to get to their destination
2. The truck driver follows the directions until they arrive at the location
3. From there the truck driver drops off the equipment that the application tells them to drop off and goes on their merry way.
4. The truck driver confirms that they have done the job on the application.
5. Then they move on to the next action that the application determines

Alternatives:

Exceptions:

Use case ID: 0004

Name: Generate Route

Pre-conditions: User is driving

Post-conditions: Move on to next action

Initiated by: Application

Triggering Event: Application opened

Additional Actors: N/A

Sequence of Events:

1. The application is opened and tells truck driver how to get to their destination
2. The truck driver follows the directions until they arrive at the location
3. From there the truck driver drops off the equipment that the application tells them to drop off and goes on their merry way.
4. The truck driver confirms that they have done the job on the application.
5. Then they move on to the next action that the application determines

Alternatives: Instead of dropping off equipment they may be picking up equipment. In this case instead of removing equipment from truck they add equipment to their truck.

Exceptions: The user made a mistake and did not drop off or gather the necessary equipment. In this case the user needs to know that they must go back to correct their error

Use case ID: 0005

Name: Send Route

Pre-conditions: Manager trained how to operate the application and metadata ready to use in the registration process

Post-conditions: New Equipment is registered to be used in any project.

Initiated by: Manager

Triggering Event: Company acquired new equipment

Additional Actors: N/A

Sequence of Events:

1. The Manager opens the application, and choose the option to register new equipment or materials based on the construction needs.
2. The form is displayed allowing the manager to register the type of equipment, the volume, the weight, dimensions, quantity and current warehouse where it is located.
3. The manager can select from a list of Warehouses previously registered.
4. Then, the manager can choose the assign which one is assigned to an existing project or can do this assignment later. In the Update Equipment Use Case
5. This process can be repeated sequentially for the next Equipment that the company acquires.

Alternatives: In the event that the warehouse location is not listed, it needs to be created, this step is in the Use Case for Create Warehouse.

Exceptions: N/A

Use case ID: 0006

Name: Follow Route

Pre-conditions: User is driving

Post-conditions: Move on to next action

Initiated by: Application

Triggering Event: Application opened

Additional Actors: N/A

Sequence of Events:

1. The application is opened and tells truck driver how to get to their destination
2. The truck driver follows the directions until they arrive at the location
3. From there the truck driver drops off the equipment that the application tells them to drop off and goes on their merry way.
4. The truck driver confirms that they have done the job on the application.
5. Then they move on to the next action that the application determines

Alternatives: Instead of dropping off equipment they may be picking up equipment. In this case instead of removing equipment from truck they add equipment to their truck.

Exceptions: The user made a mistake and did not drop off or gather the necessary equipment. In this case the user needs to know that they must go back to correct their error

10. Functional Requirements

F-ID#1 – Give directions

Description: The system must show the route a truck driver will take throughout the day to drop off and pick up equipment.

Rationale: The whole application revolves around delivering equipment so the application must give the directions to follow. So, if we do not give them any directions to follow then our whole application falls apart.

Fit Criterion: While the application is running, the app should give information about the route the truck driver is taking, and progress to following instructions based on their progress.

Acceptance Tests: T-ID#1

F-ID#2 – Access information

Description: The system must contain information about the location of equipment, where construction sites are, what equipment each construction site needs, along with other information related to the location and utility of equipment. It will access this information to determine the best route for the drivers.

Rationale: For the application to give directions about where each piece of equipment is and where it needs to be, we must have a database that stores this information. Without it, there is no way for the application to determine a route to give the user.

Fit Criterion: The application must be able to access the information while determining the best route each truck driver will have.

Acceptance Tests: T-ID#2

F-ID#3 – Admin information

Description: The system must provide a way for management to look at how each day is progressing. It must also provide a way to see how previous days went and look at the route of each driver and the number of visits per location and the like.

Rationale: This is to identify any bottlenecks the truck drivers have when doing their route. If management sees that they visit a single warehouse 100 times in a day then they can use this information to improve their business.

Fit Criterion: Management must have the ability to view information about route information as it happens, and view information about previous days.

Acceptance Tests: T-ID#3

11. Data Requirements

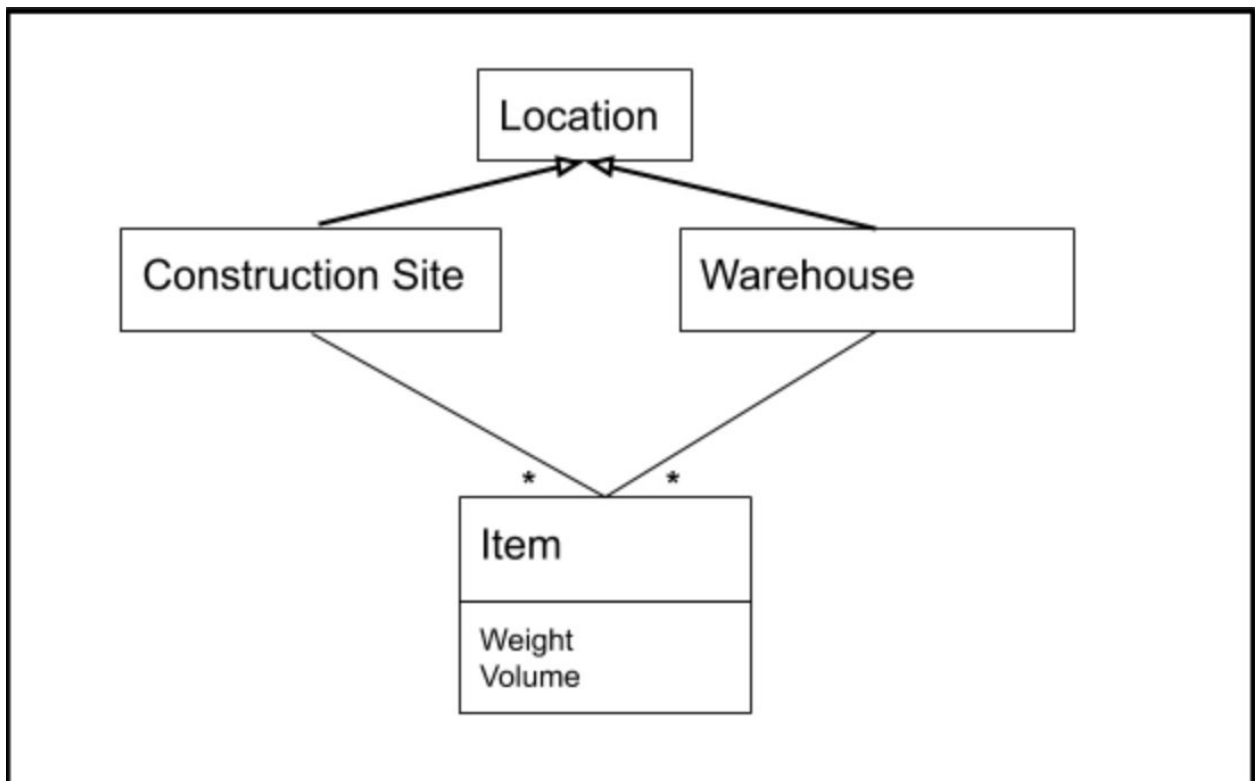


Diagram #001 Description: This is an example of what the class structure of the application may look like and how the data may be stored, modeled in a UML class diagram.

D-ID#1 - Data Security

Description: The application should maintain a moderate level of security to protect the data of the construction company.

Rationale: The privacy of information held within the software is not an extremely high priority since the amount of materials at a construction site is not sensitive information. Nonetheless, to prevent complaints from users and unforeseen consequences of not securing data, these measures must be in place.

Fit Criterion: No one other than admin-level users of the system should be able to access the information of managers, drivers, or materials currently at a construction site.

Acceptance Test: T-ID#15

D-ID#2 - Site and Item Information

Description: The application should keep track of the locations of construction sites and warehouses, along with the weight and volume of various materials needed.

Rationale: Without this information, the software would not be able to perform with the expected functionality.

Fit Criterion: The system must be able to store and, during the determination of the route, access information regarding where the construction site(s) and warehouse(s) are located and how much space each item takes.

Acceptance Test: T-ID#26

12. Performance Requirements

12.a. Speed and Latency Requirements

P-ID#1 - Manager Input (Register / Update)

Description: The Application shall load the input forms with needed meta-data in no more than 1.5 seconds, so the manager can register or update any of the entities required to feed the database. i.e

- Locations (Constructions sites, warehouses)
- Truck Capacity
- Equipment information (weight, volume)
- Project needs (required equipment, priority)

Rationale: The registration / Update process is a critical step but does not demand high-response, since it is going to take place the day before the actual delivery of equipment, though a limit time is set to allow quick operation.

Fit Criterion: The application should allow entering several of this entity in a smooth way, so response time greater than 2 seconds is unacceptable

Acceptance Tests: T-ID#5

P-ID#2 - Generate Routes

Description: The Routes are going to be generated following the Traveling salesman approach. Although this is a queue process that is going to be processed later after the data is entered, it should be completed no later than 30 min after the request of generation is placed.

Rationale: The Application does need to generate the routes right away since the process is going to be taken by a queue manager later, and finally send the routes to drivers later or at night.

Fit Criterion: Even though the route generation is handled in a queue process, the expected time to take the task and generate the route is 2-3 min.

Acceptance Tests: T-ID#1

12.b. Precision or Accuracy Requirements

P-ID#3 - Equipment dimensions and weight

Description: The weight, registered in Kilograms (equivalent in Lbs will be displayed), will be rounded to the upper whole number. The dimensions are going to be registered in meters and centimeters, equivalent in feet and inches will be displayed.

Rationale: The systems need weight and volume (calculated from the dimensions) to accommodate the equipment in an efficient cargo.

Fit Criterion: The application needs to consider the capacity of the available trucks plus the priority of equipment in the assigned construction project.

Acceptance Tests: T-ID#6

P-ID#4 - Location

Description: To register the construction sites, the application uses a Global Positioning location using a map.

Addresses are optional as Reference.

Rationale: In order to generate the routes for a map, the application needs the difference locations (GPS) and based on its distance and priority, it will generate the best optimal route.

Fit Criterion: Global positioning locations consider latitude and longitude. and will be saved as an attribute of the Construction site entity.

Acceptance Tests: T-ID#6

12.c. Capacity Requirements

P-ID#5 - Disk space

Description: A desktop-version application will not require more than 500 MB space in the hard disk. The routes generation is going to be handled by a queue manager in a separate server, which requires 1G space. And 200 MB space is required in Mobile devices + 10 MB for Route.

Rationale: The application can support the limited number for managers per company due to is not expected that more than 50 managers are registering construction project info.

Fit Criterion: The installation on mobile devices can be higher up than 210 Mb if the user does not wipe old data.

Acceptance Tests: T-ID#7

13. Dependability Requirements

13.a. Reliability Requirements

DR-ID#1 - In the Event of Failure

Description: In case a failure occurs, no information should be lost.

Rationale: Without information being saved, the software would not be able to perform with the expected functionality the next time it is run.

Fit Criterion: The system must lose 0% of the information in the event of a shut-down (either a normal exit of the application or a crash).

Acceptance Test: T-ID#27

13.b. Availability Requirements

DR-ID#2 - Time Available

Description: The system should be available for use 24 hours a day, 365 days a year (though shutdowns for maintenance/updates may occur at most once per week and between the hours of 10:00 PM and 4:00 AM).

Rationale: Managers or owners of construction sites should be able to plan routes for delivery at any time they would like. The window for maintenance or updates is set at that time frame because those hours are not likely to be during the workday for construction sites and so, not many users will be on the system (advance notice will be given to all users of the system in the case that it must be shutdown for these reasons, so ideally no users will be on the system).

Fit Criterion: The system must be available at all hours.

Acceptance Test: T-ID#28

13.c. Robustness or Fault-Tolerance Requirements

DR-ID#3 - Reload Application

Description: If the system is taking too long to perform an action, it will reload and try again.

Rationale: With the software frozen, the user is unable to interact with the application or view any of its information.

Fit Criterion: The system must allow the user to refresh it, or do so automatically after a certain period of time has passed since the last instruction began executing.

Acceptance Test: T-ID#14

13.d. Safety-Critical Requirements

DR-ID#4 - Overheating

Description: The system should not result in overheating the computer on which it is running.

Rationale: Overheating the computer will cause serious damage and could present potential risks to the health of individuals near the computer.

Fit Criterion: The system must be able to work with many other applications running in the background and should not require a lot of work from the processors in the CPU.

Acceptance Test: T-ID#7, T-ID#29

14. Maintainability and Supportability Requirements

14.a. Maintenance Requirements

M-ID#1 - Change Request

Description: Every new request or update to an existing one, is going to be accomplished within a sprint(2weeks) considering a possible change of priorities.

Rationale: Following the agile practices and keeping a good relationship with the Customer, we can accommodate a new requirement in the current sprint.

Fit Criterion: The product owner will meet weekly with the Customer to gather any possible new requirements.

Acceptance Tests: T-ID#8

14.b. Supportability Requirements

M-ID#2 - Guarantee and post-maintenance

Description: After the release of the product, there is a window time of 6 months where the customer can report issues and the development team should address them in 1 or 2 sprints depending on the complexity of the issue.

Rationale: The team will train help desk employees which can address the most common problems that can arise.

Fit Criterion: The product should maintain the standards and not overpass the threshold of reported incidents, which allow the team to offer a high-quality product.

Acceptance Tests: T-ID#8

14.c. Adaptability Requirements

M-ID3# - Platforms

Description: The application will have a Desktop-version, which can run either in Windows, Linux or Mac Operating system. The Mobile version is agreed to be developed in Java for Android supporting mobile devices. Specifications for the future can include Apple iPhone mobile support.

Rationale: The desktop version is developed in java which can run in any of these platforms.

Fit Criterion: The Registration can be done in the Web version or in the Desktop.

Acceptance Tests: T-ID#9

M-ID4# - Languages

Description: The mobile product will initially offer 2 languages: The default is set to English, but optionally the user can choose Spanish. New other languages will be a new requirement.

Rationale: The second option is for the Driver who can be more comfortable reading indications on their mother tongue.

Fit Criterion: The change of languages happens in real-time when the drivers use the mobile application.

Acceptance Tests: T-ID#9

14.d. Scalability or Extensibility Requirements

M-ID5# - Scaling

Description: The application is intended to serve construction companies in the state of Illinois, starting with the first customer in Chicago. The users (managers and drivers) per company are set to 100. Extending to the rest of the states will take 3 years where we are expected to support 200 companies with 100 users each.

Rationale: Setting first city-level companies will allow us to make adjustments and fix critical errors, and then scale State area level.

Fit Criterion: The application is developed following the design patterns that let us avoid common problems already known.

Acceptance Tests: T-ID#10

14.e. Longevity Requirements

M-ID6# - Life span

Description: Once released for its regular use, the application should operate for a minimum of 7 years - what the guarantee was established. Then is recommendable an upgrade of software and hardware

Rationale: Software has not life span, but the mentioned time frame is to limit a guarantee if no changes have occurred in the hosting hardware.

Fit Criterion: Team will maintain the practice of updating the latest tools to build the application to guarantee the correct operation once deployed.

Acceptance Tests: T-ID#11

15. Security Requirements

15.a. Access Requirements

S-ID1# - Authentication

Description: The admin will grant access through providing user and password, so either Manager and driver can log in and use the application. Users are able to change the password.

Rationale: To maintain the security requirements and let only authenticated users operate the application.

Fit Criterion: The system will log any failed log-ins and see if not-allowed access was attempted.

Acceptance Tests: T-ID#12

S-ID2# - Roles

Description: User with role "Manager" can log in and Register entities information (sites, truck, equipment) and Request a Route Generation to be sent to the assigned driver.

"Drivers" are solely able to log-in to the app in the mobile device and load a map with the generated route that was sent to its user.

"Admin" role can create these users and assign roles to them

Rationale: The drivers don't need to see any of the other entity information. On the other hand, the manager can see the map that was generated just for auditing purposes.

Fit Criterion: The System will log and report if any of the roles attempt to do an action that is not allowed.

Acceptance Tests: T-ID#12

15.b. Integrity Requirements

S-ID3# - Transactions

Description: Every transaction must follow the ACID properties, thus ensuring no corrupted data. The application must prevent from registering duplicated entities.

Rationale: If a construction site is registered by a manager using only the address, another manager should be able to register the same address for a new construction site.

Fit Criterion: During regular operations of the application less than 1% of the data should be inconsistent.

Acceptance Tests: T-ID#13

S-ID4# - Database Restore

Description: The database is going to schedule an incremental cumulative backup every day (by the end of the business day), encrypted and safely store in a cloud service. i.e. AWS. So, in the event of any catastrophe, DBA can restore it timely.

Rationale: With the aim of not losing the business and credibility of the customer, we must implement policies to backup and restore data.

Fit Criterion: Backups should generate daily and once a month DBA schedule a restore activity to ensure the consistency of the data.

Acceptance Tests: T-ID#14

15.c. Privacy Requirements

S-ID5# - Report of performance

Description: Either the Manager doing its tasks in the application or the driver following the route, the compliance of its duty is also registered, however, the performance of each user is displayed in a report for the administrator.

Rationale: Nobody else other than the administrator should be able to monitor the performance of each employee. Thus, their work is also considered private from other employees.

Fit Criterion: The Report options should be available only for user with Administrator role.

Acceptance Tests: T-ID#15

S-ID6# - Notify Admin user of sensitive information

Description: Whenever the Administrator or Manager is about to register sensitive information of a certain entity, the application will notify them of the data collection practices the company holds.

Rationale: The company needs to fulfill the law and compliance related to storing sensitive information, such as Names, Address, SSN, etc.

Fit Criterion: The user must agree on the confidential terms that are presented on the application before registering sensitive information for the first time, this response must be recorded for future reference.

Acceptance Tests: T-ID#15

15.d. Audit Requirements

S-ID7# - Confidential Information of staff

Description: The application will notify users that every action on the system is logged with its username and with the sole purpose of auditing.

Rationale: Following security procedures, in the event of any wrong transaction is performed the auditor can find out which user made the transaction, times, etc.

Fit Criterion: Metrics should be kept in log files for 2 weeks, then they should be stored in an archive media.

Acceptance Tests: T-ID#15

S-ID8# - Audit Report

Description: The system not only need to provide reports of the performance of each employee but also the interaction with the system such as: how many updates to the records of entities, when was the request of route generation triggered, at what times do the drivers fulfill the routes, etc.

Rationale: The auditor should be able to pull these reports and if more detail is needed the admin can pull up the logs with metadata of those transactions.

Fit Criterion: Auditor is an ad-hoc role that can impersonate an admin and has all the options to generate reports above mentioned.

Acceptance Tests: T-ID#12

15.e. Immunity Requirements

S-ID9# - VPN Usage

Description: If the user has not configured so, the Application must require to start a VPN with the company network.

Rationale: Using the secured network will prevent the attacker from intercept the transactions with sensitive information and inject some malicious code to impact the business.

Fit Criterion: The Application should allow the user to start the application with the proper VPN tool installed and running.

Acceptance Tests: T-ID#16

S-ID10# - Denial of service

Description: To prevent any Denial of Service the application should block any user who tries to log in unsuccessfully three times in a row. The only option to unlock it is to request it to the administrator through an email, thus it provides proof of the event.

Rationale: Possible attackers can try inserting malicious code with minimal information and try to get access with bot attempting logins many times.

Fit Criterion: The user who fails to log in gets blocked and a record of their IP is logged for future reference,

Acceptance Tests: T-ID#17

S-ID11# - Auto-update security patches

Description: The application should constantly monitor the repository of security patches and pull them up to keep the application up to date. This option is provided through the market of the mobile version.

Rationale: After the first release of the application, there might be possible vulnerabilities that need to require a code fix as soon as possible, so the team can mitigate any impact of the exploit of a bug

Fit Criterion: The application should create a log where states that check for updates was performed either an update was found or not.

Acceptance Tests: T-ID#11

16. Usability and Humanity Requirements

16.a. Ease of Use Requirements

H-ID#1 - Mistake Rate

Description: We should see very little mistakes come from new users and experienced users. We want this because of the nature of the job they are doing where mistakes can be costly to the whole route of delivery and can make what they do very inefficient.

Rationale: To make the most out of each employee's time, minimizing mistakes maximizes the time that each employee does productive work rather than fixing errors.

Fit Criterion: We should see routes delivered to completion at a rate of at least 99%

Acceptance Tests: T-ID#4

16.b. Personalization and Internationalization Requirements

HI-D#2 – Language Selection

Description: The product will allow for user to select a language.

Rationale: We want to give the users the best chance to follow their route without mistakes. To do this we should let them select the language that they are most comfortable with.

Fit Criterion: 100% of users should report that the language configuration allows users to select a language that they are most comfortable with.

Acceptance Test: T-ID#18

HI-D#3 – Sound Configuration

Description: The product will allow a user to select whether they want to receive their directions with sound and text and to select the volume of the sound.

Rationale: We want to give users an option to listen to the directions that way when they are driving, they can keep their eyes on the road. Also, some people have poor vision and allowing sound will assure these people that they know what they must do. There are also people with poor literacy and giving these people an option to be more productive leads to more deliveries occurring.

Fit Criterion: 90% of users should report that they are aware of the sound option and have knowingly toggled it on or off.

Acceptance Test: T-ID#19

16.c. Learning Requirements

HI-D#4 – Simple Hands-On

Description: The functionality available to hands on users should be easy to use for someone if they can read and understand their best language well.

Rationale: The most our truck drivers need to do is pick up equipment, drive, drop off equipment, and communicate with someone what equipment they need. Because of the simple nature of the work the people who apply for this job must not have outstanding qualifications.

Fit Criterion: 90% of users should understand what the application does and how to use it effectively after a single tutorial.

Acceptance Test: T-ID#4 , T-ID#20

HI-D#5 – Simple Admin

Description: The functionality available to management should be easy for someone who has previous experience in this field or a related one. Or someone who has undergone training or has a college degree in this field.

Rationale: Our users on the management side need to understand what the information is and how to use it effectively. If our users do not have proper training or experience, then we should not expect them to manage this product well.

Fit Criterion: 99% of management users should understand how to access and understand the management portion of the application after undergoing training.

Acceptance Test: T-ID#4 , T-ID#20

16.d. Understandability and Politeness Requirements

HI-D#6 – Clear

Description: The product shall use symbols and words that are naturally understandable by the user community.

Rationale: We do not want to bog down the employee with overly technical details that has no impact on their work, so we should communicate with them using terminology that is understandable by both parties while being clear. Keeping the communication channels clean will minimize the difficulty of the product for the user to our hands-on users.

Fit Criterion: 95% of users should not need explaining about the terminology of the application and what each message is trying to communicate with them.

Acceptance Test: T-ID#20 , T-ID#21

16.e. Accessibility Requirements

HI-D#7 – Color-Blind

Description: The product will have an option for colorblind users to select that way they can see what is going on in the application.

Rationale: Being colorblind is not too uncommon so if we encounter a color-blind user, we should accommodate them. This will also remove obstacles from them doing their work rather than needing to struggle to get their work done.

Fit Criterion: 100% of users should know about the colorblind option and have it toggled on or off on purpose.

Acceptance Test: T-ID#22

HI-D#3 – Sound Configuration

Sound configuration mentioned in section 16-b this is also an accessibility requirement. Helps users who are hard of hearing.

16.f. User Documentation Requirements

HI-D#8 – Tutorial

Description: The product will ship with a tutorial that outlines the functions of the app and gives assistance on how to access certain features and will allow each user to view it when necessary.

Rationale: If our users do not know how to use the application then it will have no purpose, or they will misuse the product. Giving a tutorial allows each user whenever they are confused to access it so that they know what they are doing.

Fit Criterion: 100% should know about the tutorial.

Acceptance Test: T-ID#23

16.g. Training Requirements

HI-D#9 – Training

Description: All users of the application will receive training.

Rationale: Users must receive training to acclimate themselves with a new part of their daily routine, interacting with the application. If they do not receive training, then they will not know how to use it.

Fit Criterion: 100% of users will undergo training.

Acceptance Test: T-ID#23, T-ID#24

17. Look and Feel Requirements

17.a. Appearance Requirements

L/FI-D#1 – Not Distracting

Description: The application's visual appearance should not distract users from driving.

Rationale: Hands-on users will drive all day and if this application is a visual distraction for some reason then it will make driving more dangerous.

Fit Criterion: 99% of users will say this application is not visually distracting while driving.

Acceptance Test: T-ID#25

L/FI-D#2 – Minimalist

Description: The application's visual appearance should only contain the bare minimum required to function.

Rationale: Construction companies will use this application to improve efficiency of employees, so we do not need an attractive design, we need a functional one.

Fit Criterion: 99% of users will say they can access features necessary for app to work without hassle.

Acceptance Test: T-ID#25

17.b. Style Requirements

L/FI-D#3 – Professional

Description: The application's visual appearance will be professional.

Rationale: This is an application meant for work not fun so we should make the style of the application reflect that.

Fit Criterion: 80% of users will say this application looks professional.

Acceptance Test: T-ID#25

L/FI-D#4 – Trustworthy

Description: The application will look trustworthy.

Rationale: We want users to have confidence that the application is giving trustworthy information. If it looks trustworthy then users will be more confident taking directions from the app.

Fit Criterion: 90% of users will say this application looks trustworthy.

Acceptance Test: T-ID#25

18. Operational and Environmental Requirements

18.a. Expected Physical Environment

OE-ID#1 - Environment

Description: The application may be used in an office setting, outside and standing up in the middle of a construction site, or on the road.

Rationale: Project managers and owners of the construction company need to be able to access the application both in their office and on various construction sites. Furthermore, truck drivers will need to be able to use the application on the road to view the route as they travel from destination to destination.

Fit Criterion: The system must be available with limited to no Internet access, and/or provide a means for the user to download the route or other information for easy access on a construction site or on the road.

Acceptance Test: T-ID#30

18.b. Requirements for Interfacing with Adjacent Systems

OE-ID#2 - Interaction with Database

Description: The application must be able to interact with the technology used for the database that stores the information.

Rationale: Without this information, the software would not be able to perform with the expected functionality.

Fit Criterion: The system must be able to access the database 98% of the time when determining a route.

Acceptance Test: T-ID#2

18.c. Productization Requirements

OE-ID#3 - Distribution

Description: The application will be distributed as a ZIP file that is only available after purchase of the software. It must have easy-to-follow steps for installation.

Rationale: This format does not require the purchase of other products and will allow for faster and easier installation, since the user may purchase online and immediately receive the software.

Fit Criterion: The system must be easily installed (less than one hour) for 95% of users.

Acceptance Test: T-ID#9, T-ID#20

18.d. Release Requirements

OE-ID#4 - Release Cycles

Description: Maintenance releases and updates to the software that improve functionality and/or efficiency should be made available as often as once a month to once every six months and must be backwards compatible.

Rationale: The software must be constantly improving and bug-free to guarantee customer satisfaction.

Fit Criterion: The system updates must not cause previous versions to fail and should take 15% of the effort required for the first release of the product.

Acceptance Test: T-ID#31

19. Cultural and Political Requirements

19.a. Cultural Requirements

C/P-ID#1

Description: The program needs to be usable in any location with an internet connection, so it must take cultural and socioeconomic differences into account. That way, the software can achieve high customer satisfaction.

Rationale: The software needs to be used anywhere so that the driver knows where to deliver the construction equipment. A driver shouldn't have to get lost when getting to their destination. For example, a delivery company in a third world country should still have access to the same product without discrimination.

Fit Criterion: 99% of new users of the software should not notice performance issues based on their current location,

Acceptance Tests: T-ID#32

19.b. Political Requirements

C/P-ID#2

Description: The software can be used in many parts of the world in the country's native language. However, the software must be developed and maintained in the United States of America.

Rationale: The software firm is based in the United States of America and it was developed for profit

Fit Criterion: The software must become profitable over time that exceed millions of dollars annually.

Acceptance Tests: T-ID#33, T-ID#36

20. Legal Requirements

20.a. Compliance Requirements

L-ID#1 Confidentiality

Description: All delivery information must be kept private.

Rationale: The database has access to company information and all the information must be kept private and confidential between different delivery companies.

Fit Criterion: While the software is running, every company should have access to only their delivery records only

Acceptance Tests: T-ID#34

20.b. Standards Requirements

L-ID#2 Delivery Standards

Description: The software must follow the rules of any construction delivery program along with the local laws.

Rationale: This software is set up for many different construction delivery companies who each have their own different rules of operating.

Fit Criterion: When the software is running, the software must follow the company rules in route planning and dispatching.

Acceptance Tests: T-ID#35, T-ID#37

21. Requirements Acceptance Tests

21.a. Requirements – Test Correspondence Summary

Table 1 - Requirements - Acceptance Tests Correspondence

[illegible]

[illegible]

21.b. Acceptance Test Descriptions

T-ID#1- Optimal Route

Description: This test will make sure that the user receives the optimal route for some information.

T-ID#2- Information Gathering

Description: This test will make sure that the application can gather information from the database.

T-ID#3- Admin test

Description: This test will make sure that the management users can access information that is only for them.

T-ID#4- Error Test

Description: This test will measure the rate of error that users will have in standard use.

T-ID#5 - Register entities information

Description: This takes will make sure the manager can register or update the entities (sites, truck, equipment).

T-ID#6 - Measure Unit

Description: This test will make sure that every entry about measure must follow the assigned unit as well as the GP location.

T-ID#7 - Disk space

Description: This test will measure the space required for desktop and mobile application and it shall not pass the assigned limit.

T-ID#8 - Change Request

Description: This test will make sure that the new requirement must be deployed on the production environment within one sprint from the date it was approved.

T-ID#9 - Adaptability

Description: This test will confirm that The desktop application can run in the 3 most common OS (Windows, Linux, Mac) and can provide at least the 2 designated languages.

T-ID#10 - Scaling

Description: This test will measure how many users are being registered and actively using the application during each period and confirm that it is growing according to the plan.

T-ID#11 - Longevity

Description: This test will make sure that the application provides the option to upgrade itself by checking the repository of updates.

T-ID#12 - Security

Description: This test will make sure that every transaction including the failed ones is recorder properly in log files and after 2 weeks stored in an archive media.

T-ID#13 - Integrity

Description: This test will make sure that the application won't fail with concurrent user operation and registering at the same time.

T-ID#14 - Restore

Description: This test will make sure that the data is secure following the Backup and Restore practices by executing the scripts in a small version of the DB.

T-ID#15 - Privacy

Description: This test will make sure that no other than the admin can access the Manager or Driver's information. The application is following the law requirements and notify the user of the actions being taken.

T-ID#16 - Immunity

Description: This test will make sure that the application cannot run without the proper configuration of the VPN in both the workstation for the manager and device for the driver.

T-ID#17 - User locked

Description: This test will confirm that the user gets blocked once it tries to log in unsuccessfully for more than 3 times in a row.

T-ID#18 - Languages

Description: This test will confirm that there is support for multiple languages.

T-ID#19 - Sound

Description: This test will confirm that there is support to receive information from the application with audio supplemented with the text.

T-ID#20 - Simplicity

Description: This test will assure that users will find this application easy to use and causes little confusion for their tasks.

T-ID#21 - Common Terminology

Description: This test will assure that users of the application will not be confused about the terminology within the application.

T-ID#22 - Color Blind

Description: This test will ensure that the option for a colorblind mode is effective for helping colorblind users.

T-ID#23 - Tutorial Test

Description: This test will ensure that there is an option for a tutorial of the application with the application.

T-ID#24 - Training Test

Description: This test will ensure that each user has undergone training for the application.

T-ID#25 - Distraction Test

Description: This test will ensure that users of the application do not find that it disturbs their ability to perform their job.

T-ID#26 - Site and Item Information Test

Description: This test will ensure that the application correctly stores and is able to access information regarding where the construction site(s) and warehouse(s) are located and how much space each item takes.

T-ID#27 - Shutdown Test

Description: This test will ensure that the application correctly saves all information in the case of a shutdown - either a normal exit of the system by the user or a crash in the program.

T-ID#28 - Availability Test

Description: This test will ensure that the application is available 24/7 by having developers test accessing the system at random times and dates and confirming that functionality is as expected.

T-ID#29 - Overheating Test

Description: This test will ensure that the computer does not overheat when it is running with many applications running simultaneously in the background. It will also measure the space needed and how much work is required of the processors in the CPU to ensure that these are appropriate.

T-ID#30 - Environment Test

Description: This test will ensure that the application can run in various locations, even ones with limited to no Internet access, and that there exists a functioning option to download information that is needed.

T-ID#31 - Test of Updates

Description: This test will ensure that the application updates periodically, with backwards compatibility and with the update being sent to all users of the software.

T-ID#32 - Worldwide Support

Description: This will test that the software can be used worldwide in any location with internet and GPS available.

T-ID#33 - Multi Language Support

Description: This will test the software is accurate in multiple different languages.

T-ID#34 - Confidentiality

Description: This will test that the Company can only access their own information and not access information of another company.

T-ID#35 - Company Guidelines

Description: This test will check that the software is within the boundaries of the company's guidelines and route planning methods.

T-ID#36 - Language Switching

Description: This will test that the software can be seamlessly switched between different languages without interrupting the current activity.

T-ID#37 - Federal Law

Description: This will test that the route planning algorithm follows the country's and state's (if applicable) laws.

III. Design

22. Design Goals

The design goals that we must optimize relate to the primary function of the application. Make the algorithm as efficient as possible so that our truck drivers spend more time working. We want to minimize the amount of time it takes to generate an algorithm without impacting the efficiency of the algorithm. We want to minimize the errors a user makes in delivering the routes.

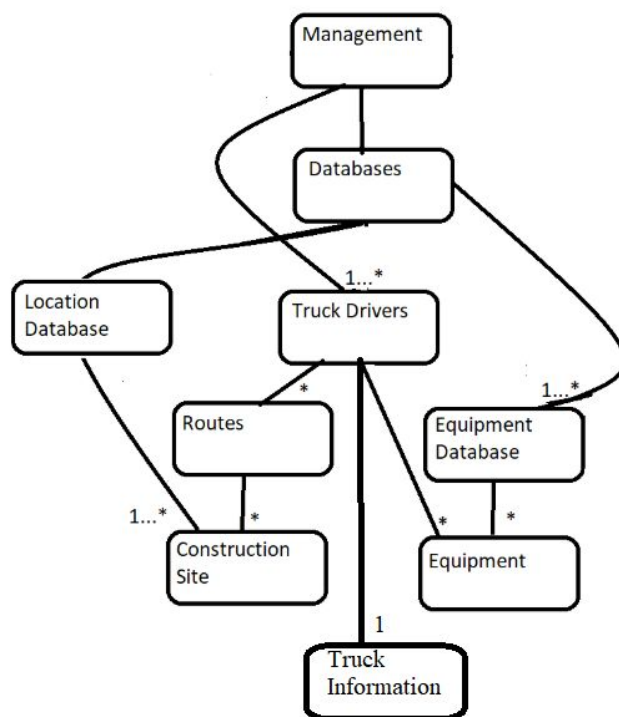
23. Current System Design

There is no pre-existing system.

24 Proposed System Design

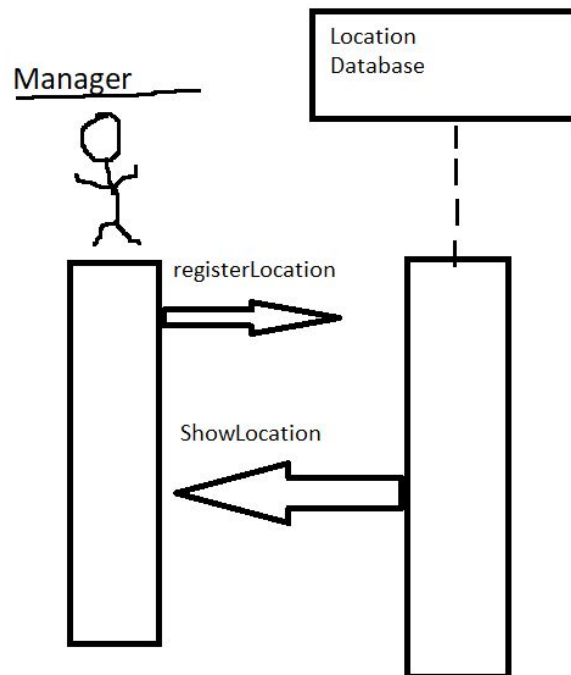
24a Initial System Analysis and Class Identification

The management class has access to all of the databases so that they can upload new information and view information that corresponds to each database. They also have access to the truck drivers so that they can view truck driver information such as the routes and what equipment they are delivering. The truck driver is not privy to the knowledge of the database it only has access to the routes which lead to construction sites and the equipment it will deliver.

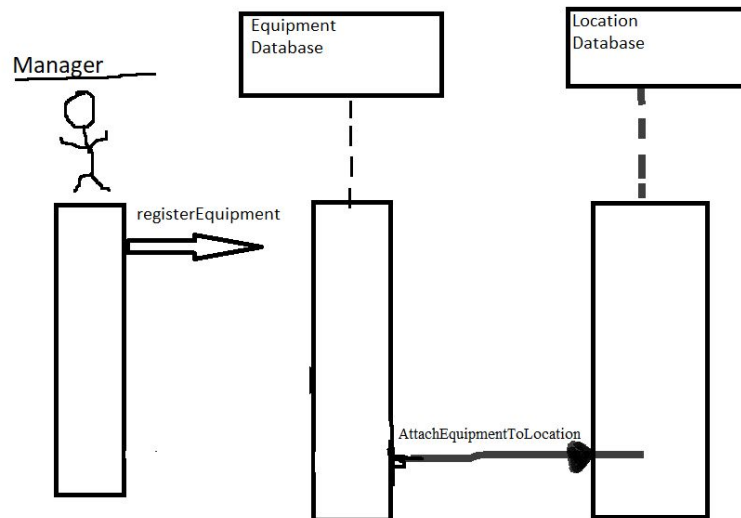


24b Dynamic Modelling of Use-Cases

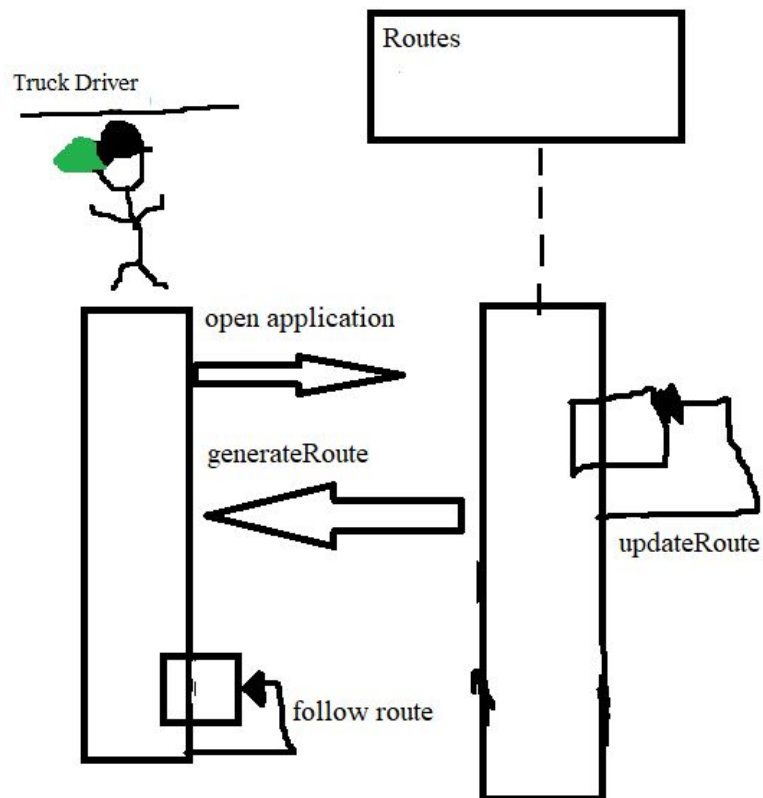
This is a use case of the manager adding a location to the database that contains all the locations that need construction equipment. To generate routes we need the location of where the equipment will be dropped off.



This is a use case of the manager adding information to the equipment database. Our application deals with dropping off equipment at locations so in this use case we are registering equipment and setting where it will be dropped off.



This diagram deals with the truck driver following the route generate. Our application is for construction companies and the hands on users of the application are the truck drivers who follow the routes that the app generates. The route updates as the truck driver follows directions.



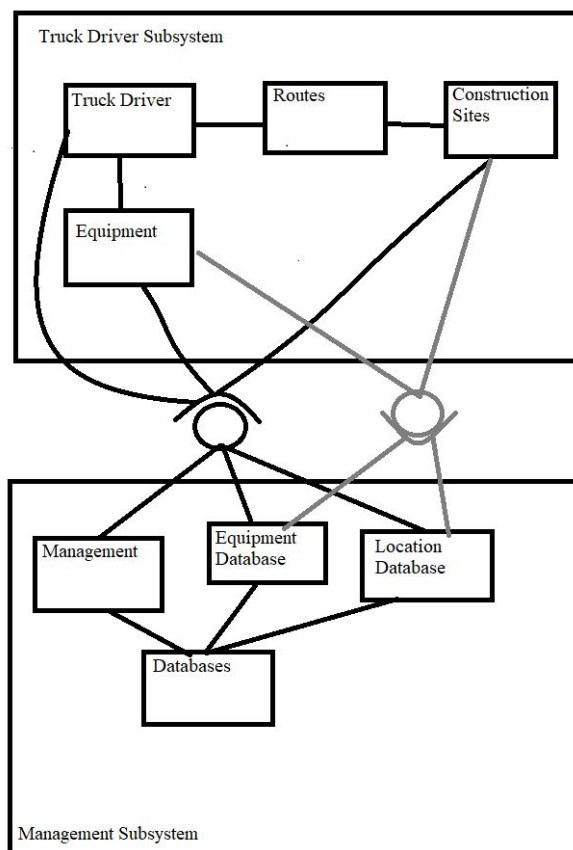
24c Proposed System Architecture

The system architecture that satisfies our project design goals and requirements is the Model-View-Controller (MVC) system architecture. The model will represent the various databases that our program requires such as storing information for equipment and storing the location of the construction sites or warehouses. The view will represent what the truck drivers see such as the routes that they will follow and the equipment that must be picked up, or dropped off, at each location. The controller will deal with updating the routes and databases as the truck driver goes about their

day. When a truck driver drops off equipment, the controller will update the database about the location of the equipment and that the equipment that location required has been satisfied.

24d Initial Subsystem Decomposition

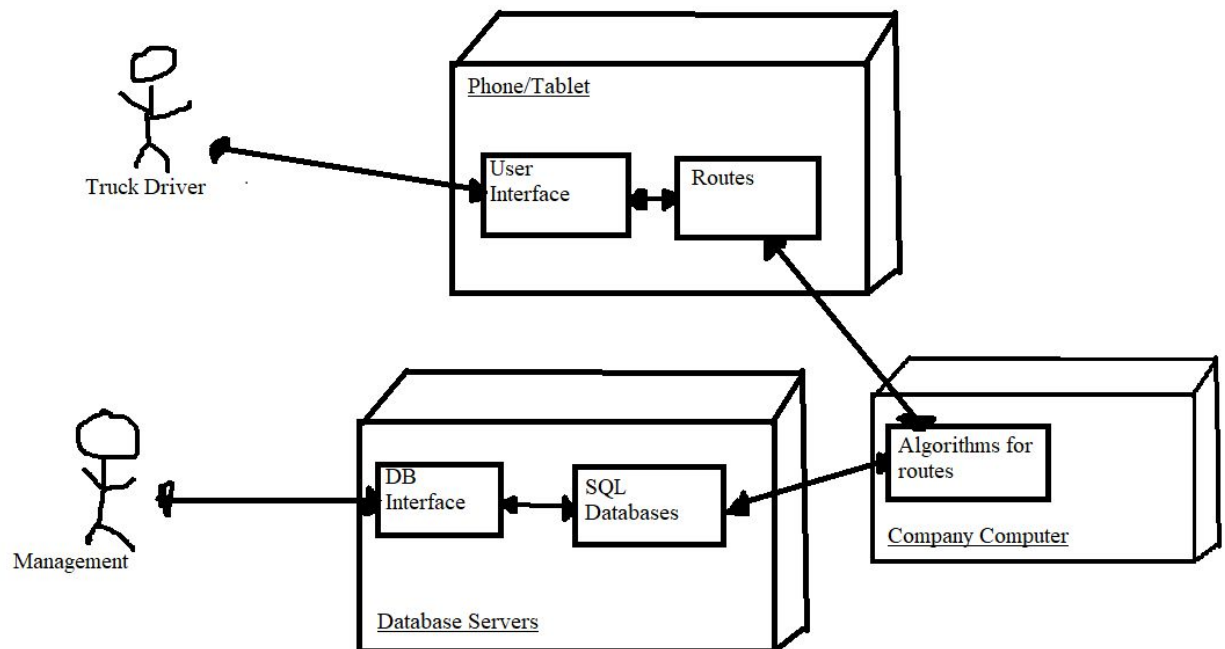
The decomposition deals with separating the two main users, those being the management side users and the truck drivers, who are hands on users. The truck driver deals with specific items in the database because they do not need access to the whole database, they only need access to items that their work deals with. The management deals with databases as they do not have to deal with specific items in databases because they are not doing hands on work. They work with the application as a whole without focus on the finer details, they must ensure that information is up to date. Information flows from the management side to the truck driver side as routes and equipment information is being pulled from the database. It is then updated when routes are completed from the truck drivers side.



25. Additional Design Considerations

25a Hardware / Software Mapping

The truck driver will open the application and use the user interface to view the routes generated by the computer. The routes are generated on some remote computer with enough computing power to generate these routes for each truck driver with considerations such as truck capacity and equipment location. Management deals with the database which in turn will feed the information it contains to the company computer so that it can use the information to generate excellent routes.



25b Persistent Data Management

Persistent data is saved on databases. These databases stores information about equipment the company handles and the location of construction sites and what is needed at each location. We rely on safety features that come with the database management system (DBMS) that we select. We can also create proprietary software to improve safety.

25c Access Control and Security

There are two different types of users so there are different permissions that come with each type of user. The main type of security is the fact that management and truck drivers would not be using their machines near each other. The truck driver

would be using their cell phone or tablet to access their information which management should have no concern over because they only need them to do their job. The management will be operating on a computer with a password that is hopefully effective. They will also not be working near the other user type so it will be unlikely that they are under security attack from them.

25d Global Software Control

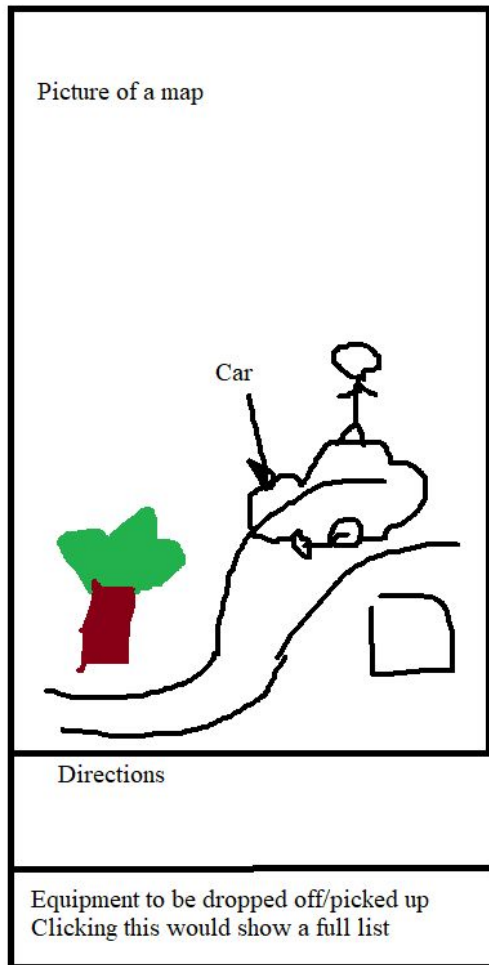
N/A

25e Boundary Conditions

When the system fails the database information must be retained with an attempt to contain the most recent data. This will mostly rely on the safety features included in the DBMS of choice. The database information is the only information that should be saved as besides that there is no other important information. Failure for the truck drivers we give them the route that has already been loaded on their phone and from their give directions via a radio until the system goes back online. Then the information must be entered into the database by management or clerical workers to maintain an up to date database.

25f User Interface

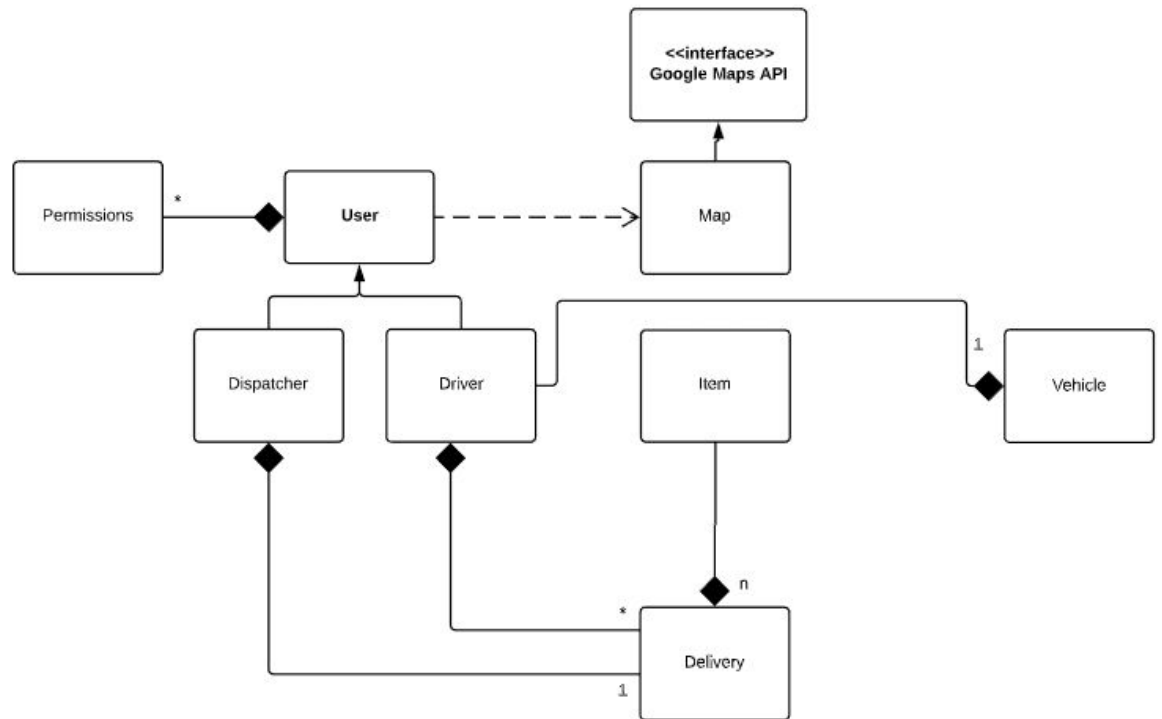
This is the user interface which contains the information to be displayed to the hands on users. The main focus is on the map but it has the direction for the map at the bottom. There would be sound for the direction. At the bottom of the screen there is a list of what equipment that must be dropped off. You could swipe to the right and you will be presented with an options menu which includes items like a color blind mode, options to mute the sound, customize the size of the user interface etc.



25g Application of Design Patterns

We will use the observer pattern for our software. Have the truck drivers be observers who are notified of routes they must take by the system and the information that is sent to them by the system. When they are done with work they can unsubscribe themselves that way they are no longer notified with new information and when they start work they can subscribe again. This will also notify management when workers have begun and ended their shift.

26 Final System Design



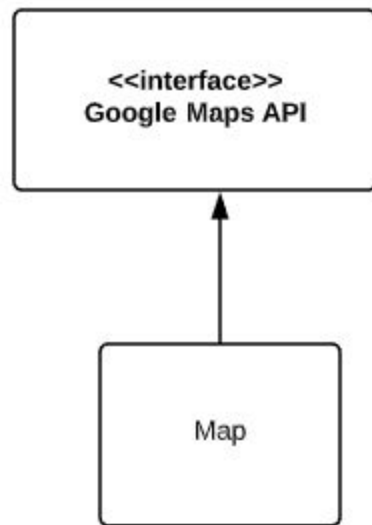
27 Object Design

27a Packages

Not applicable

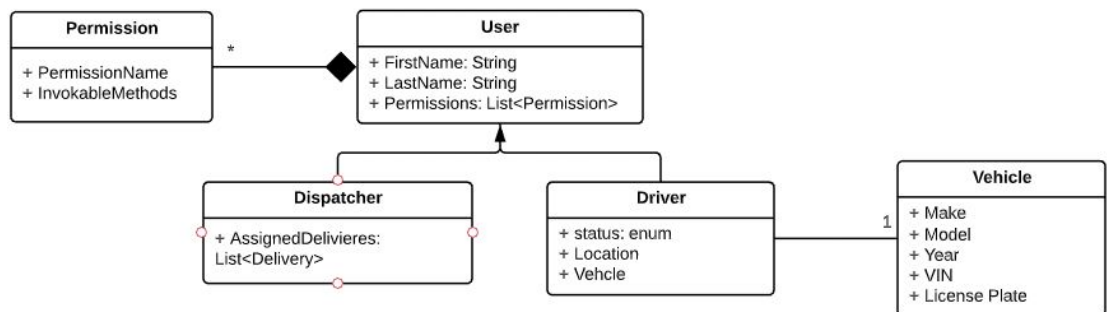
27b Subsystem I

This subsystem consists of the Map system



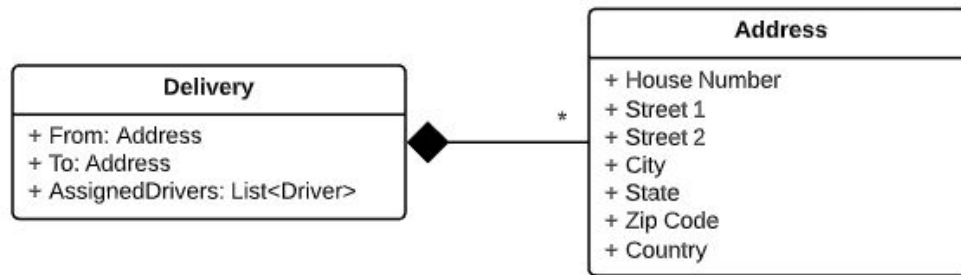
27c Subsystem II

This subsystem consists of the users of the program



27d Subsystem III

This Subsystem consists of the Delivery Records



IV. Project Issues

28 Open Issues

Will this require internet access for all tablets/phone that routes are distributed with or will they be pre-loaded into each phone/tablet?

How can we guarantee the software will make the shortest path every time with the considerations we need it to have?

How will we verify that users have not made mistakes when they have picked up/dropped off equipment at locations?

How will updates impact existing users of the product?

Will our project be a distraction on the road for the drivers causing more accidents/mistakes?

How will management receive feedback from the truck drivers performance or any bottlenecks in the system of delivering equipment?

How will work continue for people when system failure occurs for an extended period of time?

29 Off-the-Shelf Solutions

29a Ready-Made Products

There are similar solutions in the market such as:

- Routific
- Route4me

However, the application covers just one area of the business: Construction Equipment. And for that, we are not only considering the Traveling Salesman Problem, but also the Knapsack problem.

29b Reusable Components

For the software development the tech stack includes:

Back-end: Java, Spring, Microservices

Front-end: ReactJS

Mobile: IOS and Android Libraries

Need to evaluate Cloud Support in AWS for:

Database: DynamoDB

OnDemand running code: Lambda

29c Products That Can Be Copied

We may not build the Maps and its API from scratch. Instead, we will acquire a Google license for its Maps library. Thus, we can use in both Mobile Operating systems (IO/Android).

Also, the maintenance for Trucks, Projects, Construction sites, Equipment is easily developed by CRUD libraries offered in the market for the programming language we chose to operate.

Nevertheless, the algorithms of Traveling Salesman and Knapsack problems are going to be built from our end. And that is basically why we intend to offer this product to more than one company.

30 New Problems

30a Effects on the Current Environment

A potential problem arises when assigning the suggested path, but the map does not consider traffic, accidents, or blocked streets, or any other not foreseeable event which makes the driver change the route or look for better options on its own.

30b Effects on the Installed Systems

The software itself may not have any effect on the other hardware or software of the existing systems portfolio.

There should not be more demands than the ones specified in the installation requirements section. Mobile client applications will be mainly installed on tablets or

smartphones owned by the company. and intended to be used solely for business purposes.

30c Potential User Problems

In the environment, some drivers may be impacted in the way they work, since some do not like to be tracked by the GPS, although this is becoming a common practice in this type of jobs.

Also, they could know much better the streets and traffic times and would prefer not to rely on the directions of a mobile application.

There is no impact identified on the duties of the Manager role since it is also required to register the data that he or she already manages.

30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

When drivers get the proposed path to follow and are guided to their next destination, they are subject to some limitation proper of the Driving context in the streets such as Weather, traffic, which could change the route.

We are discarding problems

- Gas, because either the tank is full of gas or the route can lead you to a near Gas Station.

- The energy of the mobile devices are charged before starting the journey and those are even connected to the car outlet.

30e Follow-Up Problems

Potential problems can occur when scaling the service to hundreds of companies and the Cloud Infrastructure is not designed to support the on-demand processing.

Also, required customization of the Application is needed for specific customers, we need to monitor any possible error caused for the latest changes.

31 Migration to the New Product

31a Requirements for Migration to the New Product

Not Applicable - This is a brand new product. A new customer just need to download the application from the respective Market stores.

31b Data That Has to Be Modified or Translated for the New System

For Manager who might already have information on Projects and their needs of equipment, the address, etc, as well as the database of the equipment owned by the company, there is going to be the option to upload this information in CSV, JSON format.

It is expected that wherever the Manager uses to manage these other documents, they can export this information into the above-mentioned formats, so the import process runs smoothly.

Otherwise, the input of projects, trucks, construction sites and equipment has been manually entered into the application.

32 Risks

The application may not generate best possible routes with all the inputs.

The truck drivers may not follow the directions given by the route.

If information is input incorrectly then routes generated will be made with faulty information making them worthless or at least not as optimal.

We may cause more traffic accidents involving trucks if our drivers interact with the application too often or if they must view it too often. This will lead to death as I do not think most civilians will be able to handle the impact of a truck crashing into them. All this will ultimately lead to the company handling more lawsuits and losing money.

If software crashes the truck drivers will not know how to continue with their day and may lead to lost time.

If the application is confusing then truck drivers may spend time figuring it out during work when they should be working.

Our application may function so well that we end up firing truck drivers from the much improved efficiency. This will put people out of work and lead to higher unemployment in the country.

Application may not track the amount of gas a truck has properly which will lead to stranded trucks.

It may force truck drivers to work far too much as they will constantly have work given to them by our application. Overworking the employees may have negative impacts on their mental health.

33 Costs

The cost to complete the development of the first release of this application will include the developers' salaries, which will range from \$1,500 to \$2,000 per week for each developer. It is estimated that this application will take about three to six months to be fully developed, depending on the size of the team chosen and the skill level of the team members. The time to build the product is greatly reduced due to the incorporation of the travelling salesman and knapsack problems, for which algorithms have already been developed. Furthermore, these problems are constantly being researched, with scientists always looking for a more efficient way to solve these problems, so reuse of the code available to solve the travelling salesman and knapsack problems is efficient. This means that the majority of the development process will consist of developing the application that are specific to construction work. This includes functional requirements such as providing a means for input of information about truck capacities, warehouse and construction site locations, and inventory of what equipment is available, then enabling the routes to be sent to truck drivers' tablets or mobile devices, and notifying project managers when equipment is running low.

The initial release will also include testing of the product with real construction companies that will provide feedback regarding what features need to be improved or implemented for more efficiency or ease of use. This will also cost time and resources so as to incentivize companies to test the application, perhaps providing financial compensation of about \$1,000 to \$2,000 for each company that uses the application

for a week and provides feedback. The support team, that will be put together after the initial release, should be compensated at \$700 to \$1,000 per week and consist of a small team of about five to ten people at the beginning, perhaps expanding as the scope of the application increases. Further releases, which will incorporate new features and provide updates to the application, will require continued payment to the developers and their managers. However, these releases will not require testing in the field, as information about bugs or ways to improve the product may be collected by the application's users.

34 Waiting Room

Future releases of the application will include the following waiting room tasks (in order from highest to lowest rank):

- 1) Syncing the application with the software that the construction company uses to keep track of inventory. The application will provide the option to be connected to other programs that the company uses to keep track of inventory at warehouses, and sync with these programs to aid in determining which warehouses should be visited (since it could be the case that not all warehouses have enough tools or equipment for the closest construction sites). This will begin with making the product compatible with the most popular software used by the majority of construction companies, and then slowly including other less popular applications. In this way, information about what equipment is available may be more easily accessed and the project manager no longer needs to update both applications with this information - the product will simply retrieve the information it needs about what is in the inventory at each warehouse from the inventory software used.
- 2) Incorporating gas stations along the route that is determined. This feature will improve the routes by eliminating the need for the truck drivers to consider where and when to fill up the gas tank along their delivery route. Should the company choose, there will also be an option to include the capacity of the gas tank for the truck, gas stations within the vicinity of the construction sites and warehouses, and the prices at these gas stations. This will include stops for more gas at the most cost-effective price in the routes determined by the application.

35 Ideas for Solutions

The application should be developed in a language such as Java or C++, which has large amounts of documentation available and various libraries that may be included

within the product. The Google Maps API would be good to incorporate into this application, so as to make the process of determining distances and best routes easier. However, should the API become commercialized in the future, a more cost-effective API should replace the functionality of the Google Maps API. In order to sync with inventory software, as suggested in the Waiting Room section, other applications may be examined such as the ability for Google Maps and Spotify to work together to play music and give voiced directions simultaneously.

36 Project Retrospective

Looking back on the project, a method that proved to be effective was splitting up the work fairly among the group members. This allowed for efficient completion of the reports and kept the group members satisfied since not too much nor too little work was assigned. This is something that could work in the future as well, though to improve upon this, it would be good to look ahead at the sections and assign dependent sections to the same person. In that way, one group member does not have to wait for another group member to finish one section of the report before beginning on their own section. Retrospectively, the group worked together quite well, with each member taking appropriate responsibility for performing their share of the work. In regards to the application design, the functional and nonfunctional requirements take into consideration many of the concerns that may arise in the use of this application, and the design appropriately addresses these issues. Further analysis of problems that may arise should be done based on feedback from those who work on construction sites, and implementation details that address these concerns should be properly examined and executed in order to ensure that the product is the best one on the market.

V. Glossary

Construction Site - location that needs equipment to continue work or it has equipment that will be picked up after work at the site is complete..

Equipment - materials or hardware that a construction site may need to continue doing work.

Manager - admin user that will oversee various aspects of the application such as truck driver routes and database information.

Route - set of directions that a truck driver will follow to arrive at their destination.

Smartphone - apparatus that truck driver will get their routes from.

Tablet - apparatus that truck driver will get their routes from.

Truck driver - a person that drives a truck which will either drop off or pick up equipment.

Truck - a vehicle that stores equipment that can travel to different locations. Trucks have a given volume and weight limit that they can carry.

Warehouse - location that houses equipment. Truck drivers will visit this location to pick up or drop off equipment.

VI. References / Bibliography

- [1] Robertson and Robertson, Mastering the Requirements Process.
- [2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.
- [3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.
- [4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

VII. Index

Design	53, 62
Requirements	19, 46
Test	46, 49