Software Requirements Specification

For

Logistics Dispatch

Prepared by Keane Wesselius,
Conner Gordon, Michael Camarata,
Chris Payne, Lincoln Huber,
Galmo Said,
Joshua Ruymen, Shiva Shrestha

Prepared for Taban Cosmos

January 30th, 2023

Version 1.0

Table of Contents

Table of Figures

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed and unambiguous description of the Logistics-Dispatch software requested by Taban Cosmos. This document will explain the features, purpose, and functions of the software. This document will consider potential problems, feasibility, and priority of the proposed features. This document is meant for the client of the software and the developers of the software to ensure both parties agree and understand the functionality and purpose of the system.

1.2 Scope

The proposed software will help and assist in the transport of goods between merchants and suppliers. The software will allow three different kinds of users to login and perform tasks. The merchant user type will be able to request deliveries from suppliers and track saved deliveries. The supplier type user will confirm the merchant's delivery request. The driver user type will be able to accept deliveries confirmed by the supplier and see the optimized delivery route created by the software. Each delivery will only have one pickup location (point A) and one drop off location (point B). The different users will interact with a different version of the software. The merchant and supplier users will interact with a web-based desktop application while the driver user will interact with a mobile application.

1.3 References

"IPhone Models Compatible with IOS 16." Apple Support, Retrieved from Apple Support: support.apple.com/guide/iphone/supported-models-iphe3fa5df43/ios.

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

Razaaq, A. (2022, August 24). *Android 12: Release Date, Supported device – Everything We Know So Far*. Retrieved from GetDroidTips: https://www.getdroidtips.com/android-12/#Devices-that-will-receive-the-Android-12-update

1.4 Overview of Document

The following sections will go into detail of the proposed Logistics Dispatch software. Section 2 describes an overview of the software. Section 3 covers interfaces that the software will work with. Section 4 will cover specific features and functional requirements of the software. Section 5 explains the nonfunctional requirements of the software. The final sections are supplemental information such as a glossary.

2.0 Overall Description

2.1 Product Perspective

The software product, Logistic Dispatch, is a new standalone system and is not part of a larger system or any family of software products. The software consists of three interconnected subsystems that come together to form the main software system. The first subsystem is for merchant users, the second subsystem is for supplier users, and the third subsystem is for driver users. The software is designed so a merchant can purchase items from a supplier and the supplier can request a delivery of the purchased items to the merchant. A diagram has been provided to show the interaction between these systems and their users.

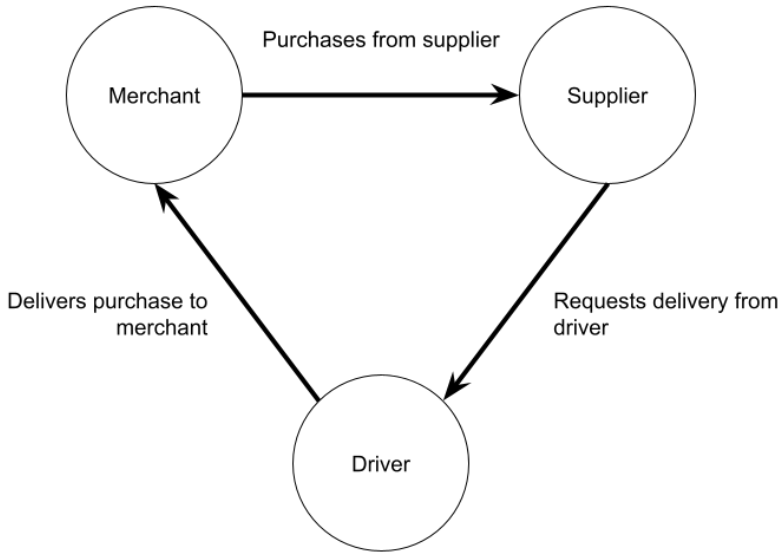

Figure 1: Overview of product roles

2.2 Product Functions

The overarching goal of the proposed software is to allow merchants to request products from a supplier and have a driver deliver the requested products. The system will be capable of requesting a delivery from point A to point B, where point A will be a supplier's warehouse and point B is a merchant's storefront.

The requested software will be capable of tracking pending, in progress, and completed deliveries. Once a delivery has been requested, the product can be tracked from its initial location to any intermediate locations until it reaches the customer. Once the product reaches the customer the application will confirm the delivery has been completed.

The system will have a user authentication system to ensure only authorized individuals are allowed to access the system. The authentication system will be capable of handling three different types of users, specifically merchants, suppliers, and drivers.

The Logistics-Dispatch application shall be able to create an optimized route for the driver to follow in order to accomplish the delivery efficiently. The system will be capable of

batching orders to optimize delivery routes further. The overall goal of this system is to schedule, monitor, and confirm the delivery of goods. The software will provide real-time alerts and notifications to ensure timely delivery of goods, such as delays and any other issues.

The following is a diagram showing the main actions the users will take and the order of operations from the start to finish of one delivery. It is important to note that for the processes that continue with a dashed line, the database must be updated before continuing onto the next process. In addition, once the merchant or supplier reaches the track order process the user can check the status of the order whenever they please, but it is not required for the delivery process to continue.

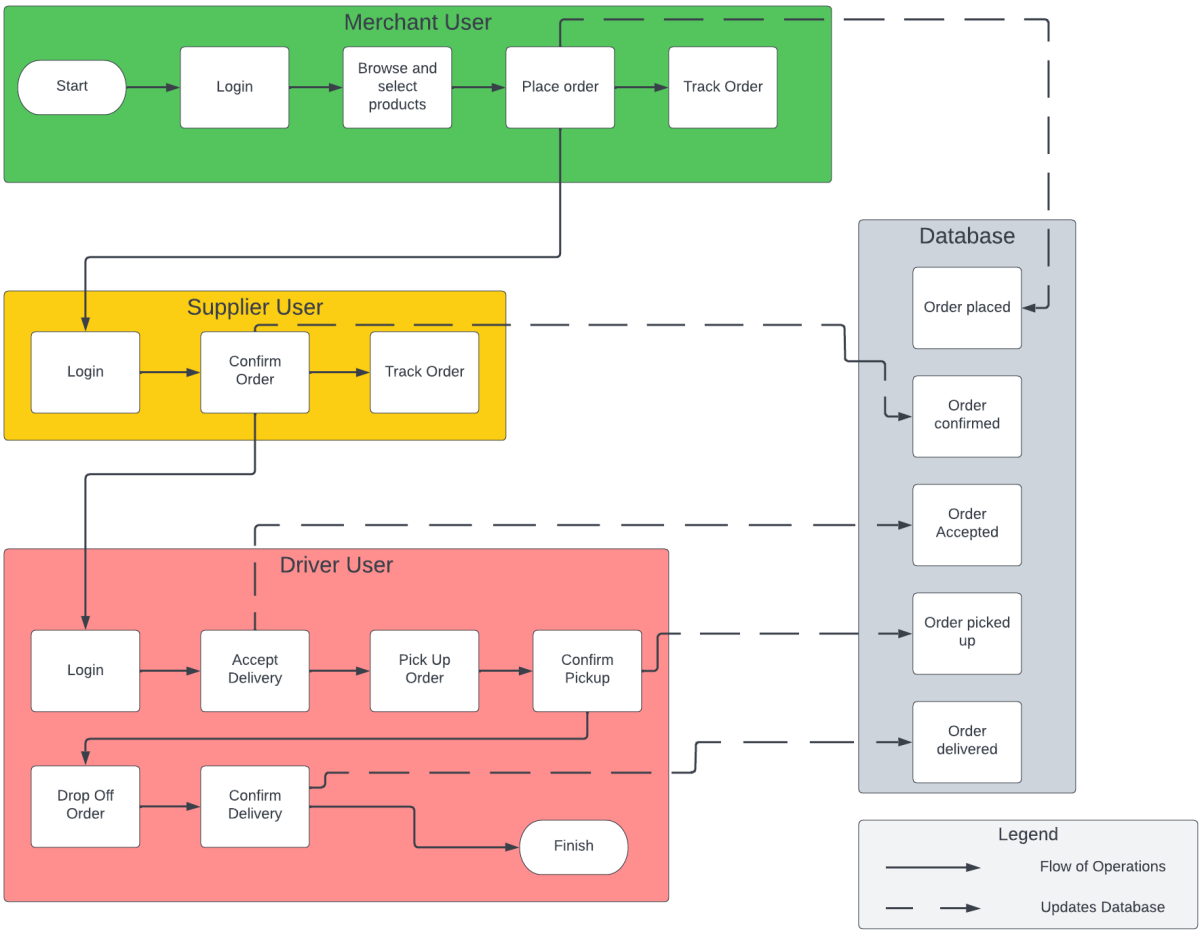

Figure 2: Overview of a delivery

2.3 User Characteristics

The system will support three distinct types of users. The first type of user is the merchant, whose use case of the system involves requesting a delivery from a supplier and checking on the status of requested deliveries. The second type of user is a supplier. The supplier has the goods the merchant wants, and the supplier's use case is accepting the deliveries

requested by the merchant. Both the merchant and supplier will interact with the desktop version of the application.

The last type of user is a driver, whose use case of the system involves accepting a delivery, picking up the goods, and then dropping them off at the required points as well as confirming the goods have been delivered. The driver is the only type of user who will interact with the mobile version of this software.

2.4 Operating Environment

There will be multiple operating environments for this application. Depending on if the user type is a driver, vendor, or merchant, a different operating environment will be required. The following subcategories briefly lay out the operating environments. Later in section 3.2 and 3.3 the specific hardware devices and software versions respectively will be explained in greater detail.

2.4.1 Desktop Environment

The merchant and supplier users will use a web-based desktop application. The desktop application will run on any desktop computer capable of running Mozilla Firefox, Apple Safari, or any Chromium-based browser with support of ECMAScript 5 and HTML5. The desktop environment must be able to store all of the database data for the system to remain operational, which will grow as the system is used. The database does not need to be run locally on the same system as the desktop server, as long as they are able to access each other via a standard network protocol such as TCP.

2.4.2 Mobile Environment

The driver user will use a mobile application. The mobile application will run on either Apple's iOS operating system or Google's Android operating system. The mobile application will support mobile devices with operating system versions outlined in section 3.3.2.1 and 3.3.2.2.

2.5 Design and Implementation Constraints

Our chosen map API, being the Google Maps API, allows for map and route information and optimization which will automatically determine the fastest route possible. Due to setting the starting and end point, the API does not allow for the route to be optimized for other factors. However, since all deliveries are from point A to point B, the fastest route would be the optimized route. Optimization of multiple drivers routes is to be included, which would be ensuring that the delivery order minimizes the amount of travel a driver needs to do. Due to not needing to do the computational expensive route optimization, the backend and database will be stored and run on a developer machine for the initial prototype phase instead of a server, although this might be untenable depending on the implemented functionality of the software.

2.6 User documentation

The documentation delivered with this software will be inline comments in the code as well as documentation on how to install and set up the software. Further documentation detailing the environment variables for the software will also be available.

3.0 External Interface Requirements

3.1 User Interfaces

The merchant user will interact with a login screen upon first opening the application. After logging in the user will see two options: request delivery and track deliveries. Requesting delivery will show the user a list of products available from suppliers to purchase which is the bottom right window in the figure below. After picking the products they want the user will click a button labeled submit order. If the user selects track deliveries, they will see a screen displaying pending, in progress, and completed deliveries specific to that merchant which is the top left window in the figure below.

The supplier user will interact with a login screen upon first opening the application. After logging in the supplier will see two options: accept deliveries and track deliveries. The "Accept Deliveries" button will take the supplier to the bottom left screen showing all the orders that a merchant has requested from the supplier. Information shown will be an accept and deny button and an itemized list of the requested items should the user click the "view details" button. Track deliveries will show all pending, in progress, and completed deliveries specific to that supplier.

The following figure shows a preliminary mockup of the desktop user interface and is subject to change over the course of development.

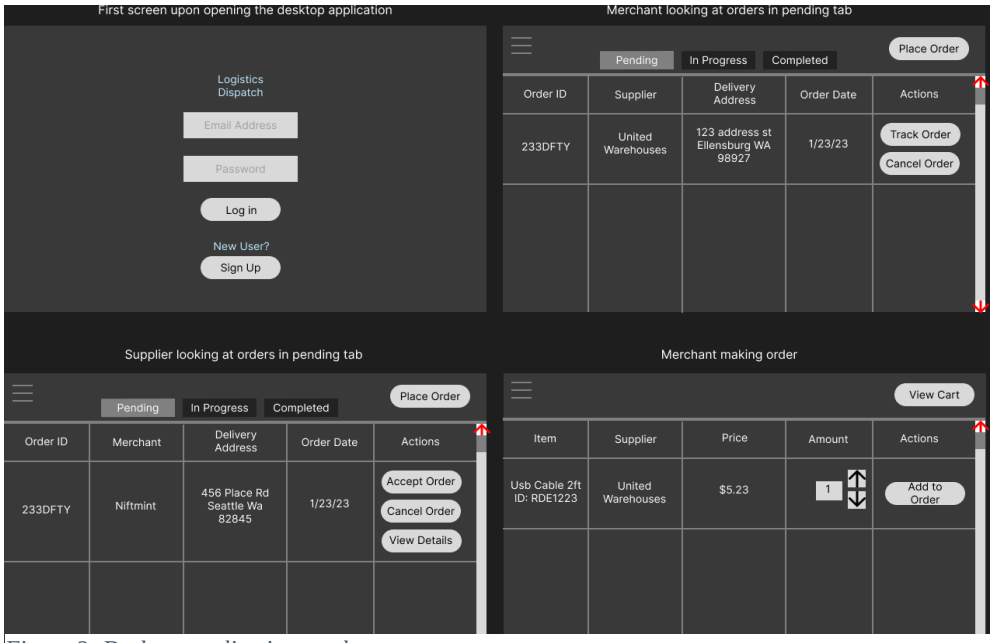

Figure 3: Desktop application mockup

The driver user will also interact with a login screen when first opening the application. After logging in the user will be shown a list of vehicles and the user will choose the vehicle they are using to deliver. Then the user will be shown a list of valid and available deliveries for that vehicle. After selecting and accepting a delivery, the application will show a map centered on the user's GPS location with a route pointing them to the pickup location and drop off locations.

The following figure is a preliminary mockup of the mobile user's desktop interface and is subject to change over the course of development.

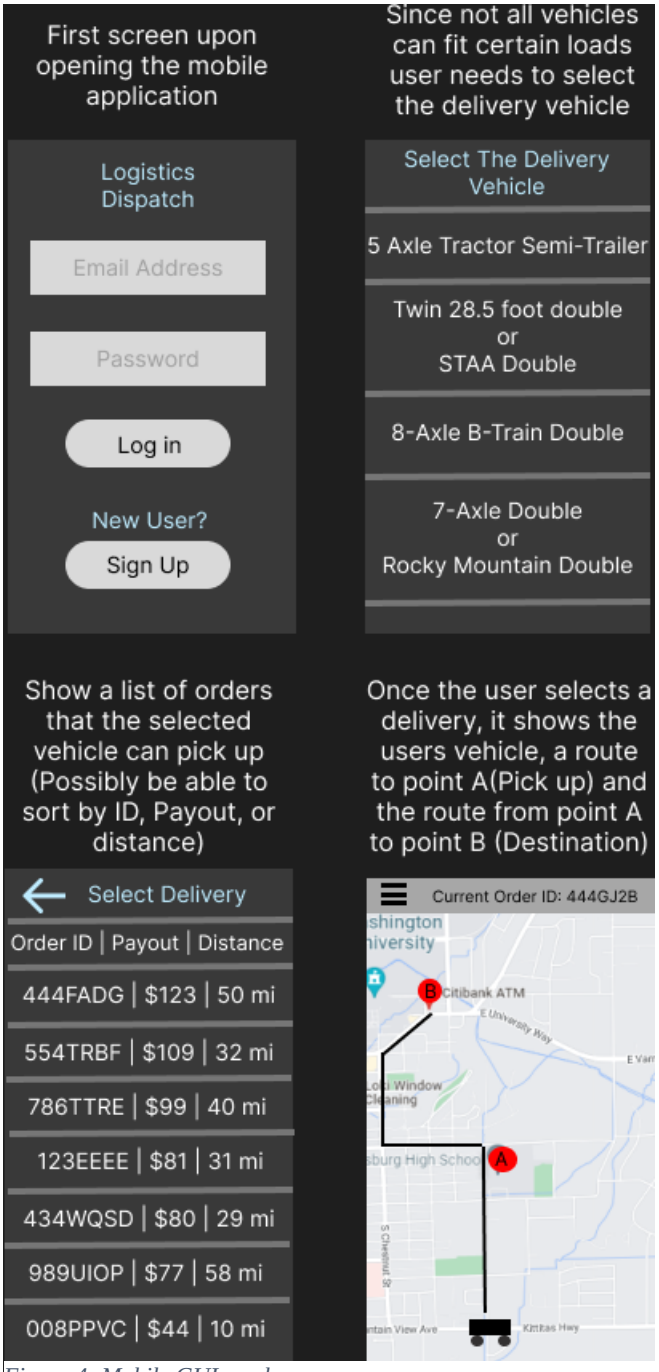

Figure 4: Mobile GUI mockup

3.2 Hardware Interfaces

This software will come in two different versions. The merchant and supplier users will use a desktop web-based application and the driver user will use a mobile application. This section is broken up into hardware sections but not every user will interact with all these sections.

3.2.1 Desktop

For the merchant and supplier users the software will operate on CISC computers with access to the internet. To interact with the application, desktop users are required to have a mouse and keyboard or any input device capable of pointer events and alphanumeric character input.

3.2.2 Mobile

For the driver users the software will operate on mobile devices, specifically iOS and Android devices. It is expected that the user's device will have a touch screen, a camera, access to the internet, and location services to successfully run the application. The following subcategories list out the supported devices depending on the operating system.

3.2.2.1 iOS

The operating systems for iOS devices supported by this product are iOS 16, 15, and 14. The device must be an iPhone 6s or later, or an iPhone SE. The following list is all the devices that support these iOS versions and therefore will be supported by this software.

3.2.2.2 Android

The supported Android versions by this product are Android 13, 12, and 11. Android is supported by different manufacturers with their own "skin" of the operating system, although functionality of the underlying operating system is essentially the same. The ability to interface with Google Maps is a requirement for the devices.

3.3 Software Interfaces

The proposed software will have two different versions with different functions and different software Interfaces. Not every user will interact with all these software interfaces.

3.3.1 Desktop/Laptop

The computers used must be capable of running Chromium-based web browsers, Apple Safari, or Mozilla Firefox as these will run the web-based desktop software. The

backend will use Node.js, which allows for the backend to utilize the same programming language as the mobile app. Node.js has official compatibility with our database, being MongoDB and is capable of handling many, asynchronous connections which is a requirement for the project. Stability of the project is also paramount, especially for the backend, so choosing a potentially more performant but potentially unsafe language such as C++ was disregarded.

3.3.2 Mobile

The mobile application will run on iOS 16, 15, and 14. In addition the mobile application will run on Android 13, 12, and 11. The mobile application with utilize the React Native Expo development framework, which utilizes JavaScript, React, HTML, and CSS for an easy cross-platform development environment which can be used to test without the need to go through the Google or Apple development pipelines or devices, which will greatly benefit initial prototyping. The mobile version of the product should be able to send system notifications and noises to notify a driver of any delivery updates on other information relevant to their delivery.

3.3.3 APIs

The driver application will interface with the Google Maps API to get information about available routes for deliveries. Both applications will interface with security/authentication API for secure sign into the applications. The backend Node.js server will interface with the MongoDB database via an official API.

4.0 System Features

4.1 General User Features

4.2.1 Create Account

4.2.1.1 Description

The software shall provide a way to create an account for the application.

4.2.1.2 Use Sequence

If the user has not created an account within the software yet the following steps will be required:

1. The user clicks “Create Account”.
2. The user clicks a dropdown menu labeled “User Type” and selects either merchant, supplier, or driver option.
3. The user enters a valid email address.
4. The user enters a password for their account.
5. The user reenters the password to confirm it.
6. The user clicks the create account button.

7. A prompt displaying the options the user input when creating their account is shown which includes user name, email, and account type, asking for confirmation.

4.2.1.3 Possible errors and handling

The possible errors that can happen is the user does not fill out a field and submits the application. To handle this the application will check if all the fields have been filled out and display a message to the user telling them what one needs to be filled in to continue. Another error that can happen is the user overlooks the "User Type" drop down and ends up creating the wrong type of account. To deal with this issue the drop-down menu will start on a generic field that must be changed before creating the account.

4.2.1.4 Priority

The priority of this requirement is high as users must have accounts for this software to view sensitive information about orders and order status.

4.2.1.5 Feasibility

This requirement is feasible.

4.2.1.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.2.1.7 Validity

This requirement has been deemed valid as creating an account is an essential feature of a logistics dispatch software.

4.2.2 Secure Login

4.2.2.1 Description

The software will allow users that have created an account to securely sign into the software.

4.2.2.2 Use Sequence

The user will have to have created an account before this using this feature.

1. The user enters the email address they used when creating the account.
2. The user enters the password they used when creating the account.
3. The user clicks "Log In".

4.2.2.3 Possible errors and handling

A possible error that can occur is the user enters an email and password that is not in the system. If this happens the software will output a message to the user saying that the email password combination is not valid.

4.2.2.4 Priority

The priority of this requirement is medium as the user will have to log into their account to access the information related to them. Due to the first version of the software being a prototype, ensuring proper security for logins is not a primary objective. Implementing a two-factor authentication system is of low priority, as having a secure login system is already not of the highest priority.

4.2.2.5 Feasibility

This requirement is feasible.

4.2.2.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.2.2.7 Validity

This requirement has been deemed valid.

4.2 Merchant User Features

4.2.1 Browse Products and Place Orders

4.2.1.1 Description

The software will display the products available for the merchant to purchase and allow them to select multiple products and place an order.

4.2.1.2 Use Sequence

The user will have to be logged in to the software before performing these steps:

1. The user selects “Make Order”.
2. The software will then display a list of suppliers and the user will select a supplier.
3. The software will then display a list of products from that supplier and the user will select the products they want.
4. The user will press “Place Order” and input delivery address.
5. The system will show the order id of the order that was just placed, and the order will go into the track orders category in the software under pending. When the order is confirmed, the system will show that order was confirmed.
6. The user will be able to see all previous orders and check the delivery status of that order.

4.2.1.3 Possible errors and handling

A possible error that can happen is that from the time the user starts selecting products to when they place the order, another user has bought up the stock of a product and there is not enough available for the current user. To

combat this, the system will check and update the available number of products when the user places the order and will give an error stating that the product is no longer available.

4.2.1.4 Priority

The priority of this requirement is high because without it the merchant would not be able to make purchases and receive products.

4.2.1.5 Feasibility

This requirement is feasible.

4.2.1.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.2.1.7 Validity

This requirement has been deemed valid.

4.2.2 Order Tracking

4.2.2.1 Description

The user will be able to track the status of the pending, confirmed, in progress, and completed orders. The user will be able to see if an order has been delivered or the current location of the order such as out for delivery or arrived at (midway location) or departed from (midway location).

4.2.2.2 Use Sequence

1. The user signs into their account
2. The user either selects the order from a list or enters the order ID and will be able to see QR code for tracking the status of delivery.
3. The status of the order is displayed

4.2.2.3 Possible errors and handling

A possible error is that the order does not get updated and therefore will be shown in the wrong place when looking at its status, this is a problem that still happens today with other delivery services and a workaround is yet to be determined.

4.2.2.4 Priority

The priority of this requirement is high as it is important to know where products are and to confirm they have been delivered at the desired location.

4.2.2.5 Feasibility

This requirement is feasible to achieve as orders have to be confirmed that they have been delivered to the correct location and such updating a value in the database of orders will achieve this.

4.2.2.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.2.2.7 Validity

This requirement is valid as tracking orders is an essential feature of logistics dispatch software.

4.2.3 Delivery Optimization

4.2.3.1 Description

The software will optimize routes depending on the pickup and drop off locations of the orders.

4.2.3.2 Use Sequence

This part of the software will be automated meaning that the user will not have to interact with the system to accomplish this requirement. The system will pick an optimized route when the merchant requests a delivery.

4.2.3.3 Possible errors and handling

A possible error that can occur is the route created by the software is using a road that has been closed. As of right now Taban has suggested that the development team not worry about these types of issues.

4.2.3.4 Priority

The priority of the requirement is high as it is required for a cost-effective solution for shipping products.

4.2.3.5 Feasibility

This requirement is feasible as the software will be using an API that has features built in for optimizing routes.

4.2.3.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.2.3.7 Validity

This requirement is valid as making deliveries more efficient is an important part of logistics.

4.2.4 Delivery Batching

4.2.4.1 Description

The software combines multiple orders that have the same pickup and drop off locations.

4.2.4.2 Use Sequence

This part of the software will be automated meaning that the user will not have to interact with the system to accomplish this requirement. The system will

recognize when multiple orders are going to and from the same locations and combine them into one delivery.

4.2.4.3 Possible errors and handling

A possible error that can occur is that the combined orders will be too large to fit on one delivery vehicle. To deal with this issue the software will calculate the combined volume of the order and display the order only to vehicles that have the space to accept them. Another way to work around this issue would be to batch orders that take up the space of the smallest delivery vehicle that could be used to deliver, however this would be an inefficient solution.

4.2.4.4 Priority

The priority of the requirement is low as it would be a nice feature to have but not required for the software to function.

4.2.4.5 Feasibility

This requirement is feasible.

4.2.4.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.2.4.7 Validity

This requirement is valid as making deliveries more efficient is an important part of logistics.

4.3 Driver User Features

4.3.1 Accept Delivery

4.3.1.1 Description

Once an order has been placed a driver will accept the delivery saying they will pick up the products and drop them off to their destination.

4.3.1.2 Use Sequence

1. The user logs into the application.
2. The user selects the vehicle they are driving.
3. The app displays the available deliveries to them.
4. The user selects the delivery they want.
5. The user clicks “Accept Delivery”.
6. Sign off package on delivery, if necessary based on priority.

4.3.1.3 Possible errors and handling

One possible error is that the user accidentally accepts a delivery, to fix this the user can cancel the delivery if they have yet to pick up the load.

4.3.1.4 Priority

This requirement's priority is high as without it the orders will not be shipped.

4.3.1.5 Feasibility

This requirement is feasible.

4.3.1.6 Consistency

This requirement is consistent with the rest of the requirements listed in this SRS.

4.3.1.7 Validity

This requirement is valid as orders need drivers to be able to accept deliveries for goods to be shipped.

4.3.2 Show Delivery Route

4.3.2.1 Description

The software will show the user their current location on a map with a line(route) showing them where the next stop they need to visit is.

4.3.2.2 Use Sequence

1. The user accepts a delivery (see the above requirement).
2. The app loads the map and shows them the delivery route.

4.3.2.3 Possible errors and handling

One possible error is that the user loses GPS signal whilst in the middle of a delivery, to correct this error the app will also give a list of instructions on where to turn to complete the delivery without the need of a GPS signal.

4.3.2.4 Priority

The priority of this requirement is high as the drivers will need to know where to go to complete their deliveries. This requirement is also a high priority because the driver may be able to get to the locations themselves, but the route would not be optimized, costing the company money.

4.3.2.5 Feasibility

This requirement is feasible as the software will be using the Google maps API which handles most of these features.

4.3.2.6 Consistency

This requirement is consistent with the rest of the requirements listed in the SRS.

4.3.2.7 Validity

This requirement is valid as the drivers will need to know what route to take to complete their deliveries.

4.3.3 Confirm Delivery of Goods

4.3.3.1 Description

The application will allow the user to confirm that a delivery has been made.

4.3.3.2 Use Sequence

1. The user accepts an order.
2. The user drives to the location the application directs them to.
3. Once a user has made the delivery, the application will allow them to confirm that the delivery was made with a button that says, “Confirm Delivery”.

4.3.3.3 Possible errors and handling

An error that can occur is that the user accidentally clicks “Confirm Delivery” when the delivery was not made. To fix this issue the user will be able to unclick the button.

4.3.3.4 Priority

The priority is this requirement is high as it will update the information the stakeholders see when tracking deliveries and to confirm or deny shipping issues should they arise with a customer.

4.3.3.5 Feasibility

This requirement is feasible.

4.3.3.6 Consistency

This requirement does not conflict with any of the other requirements listed in this SRS.

4.3.3.7 Validity

This requirement has been deemed valid.

4.3.4 Vehicle Specific Dispatch

4.3.4.1 Description

The software will assign deliveries to specific vehicles depending on the amount and type of products being delivered. Other factors that would affect the specific vehicle being selected is the distance and type of delivery (delivering to another warehouse or delivering to customers).

4.3.4.2 Use Sequence

The driver will select the vehicle they are using to deliver the goods, depending on the vehicle they choose, the options for deliveries to accept will be specific to that vehicle.

4.3.4.3 Possible errors and handling

If no drivers of a specific vehicle are available, then certain orders will not be delivered. If this happens and a delivery tailored to a specific vehicle has been unattended for too long, it will become available to other unoptimized vehicles to get the goods delivered.

4.3.4.4 Priority

The priority of this requirement is medium-high as the software can function without specific vehicles being assigned to specific deliveries.

4.3.4.5 Feasibility

Given that all the information needed to implement this feature will be available if the rest of the software is finished this is an optimization problem and as such is feasible.

4.3.4.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.3.4.7 Validity

This requirement is valid as it helps maximize the profits of transporting goods across the United States.

4.4 Supplier User Features

4.4.1 Confirm Order

4.4.1.1 Description

The software will allow the supplier to confirm an order that was placed by a merchant user.

4.4.1.2 Use Sequence

1. The user selects the “Confirm Deliveries” option.
2. The software will display a list of orders that have been requested from merchants.
3. The supplier will have the option to confirm or deny each delivery request.
4. If an order is confirmed it will update on the merchant's side showing the order has been confirmed and the order will appear on the driver’s side of the application to accept.

4.4.1.3 Possible errors and handling

A possible error that can occur is the user accidentally accepts a request they meant to deny or vice versa. To deal with this issue the software will give a grace period of 5 minutes to undo the decision before it's completely confirmed.

4.4.1.4 Priority

The priority of this requirement is high as the supplier will have to accept the order before the driver can see the delivery.

4.4.1.5 Feasibility

This requirement is feasible.

4.4.1.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.4.1.7 Validity

This requirement has been deemed valid.

4.4.2 Order Tracking

4.4.2.1 Description

The user will be able to track the status of the pending, in progress, and completed orders. The user will be able to see if an order has been delivered or the current location of the order such as out for delivery or arrived at (midway location) or departed from (midway location).

4.4.2.2 Use Sequence

1. The user signs into their account.
2. The user either selects the order from a list or enters the order ID.
3. The status of the order is displayed.

4.4.3.3 Possible errors and handling

A possible error is that the order does not get updated and therefore will be shown in the wrong place when looking at its status, this is a problem that still happens today with other delivery services and a workaround is yet to be determined.

4.4.2.4 Priority

The priority of this requirement is high as it is important to know where products are and to confirm they have been delivered at the desired location.

4.4.2.5 Feasibility

This requirement is feasible to achieve as orders must be confirmed that they have been delivered to the correct location and such updating a value in the database of orders will achieve this.

4.4.2.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.4.2.7 Validity

This requirement is valid as tracking orders is an essential feature of logistics dispatch software.

4.5 Backend Features

4.5.1 Data Storage and Management

4.5.1.1 Description

The software will store currently in-progress deliveries and previously completed deliveries and allow the backend to access and modify data in the database.

4.5.1.2 Use Sequence

1. A driver confirms their selection of a delivery. The system creates a new entry in the database tracking this delivery.
2. A driver may update the status of the delivery, which will be mirrored on the backend database.

4.5.1.3 Possible errors and handling

One possible error is in the case of the system not having enough space to store new data into the database. In this case, the system should gracefully shutdown or fail to allow deliveries to be taken in the case of low disk storage. A better solution is a mechanism which will migrate to a backup, temporary off-site storage solution to ensure the system is able to maintain operability.

4.5.1.4 Priority

The priority of this requirement is high, as it is vital to the backend to function, which if non-functioning, doesn't allow the mobile app to function.

4.5.1.5 Feasibility

This requirement is feasible.

4.5.1.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.5.1.7 Validity

This requirement has been deemed valid.

4.5.2 Route Optimization

4.5.2.1 Description

The software will attempt to provide optimized routes for multiple deliveries.

4.5.2.2 Use Sequence

1. A driver confirms their selection of a series of deliveries for a day.

2. The backend will attempt to advise the driver on which routes to do first which will utilize the least amount of fuel / time. This optimization may be implemented using a Bell-Man Ford Algorithm or similar algorithm.

4.5.2.3 Possible errors and handling

One possible error is in the case of the system advising a route which isn't optimal either due to a miscalculation or other external factors, such as a road closure.

4.5.2.4 Priority

The priority of this requirement is low, as drivers have the option to reject the suggestion in the first place.

4.5.2.5 Feasibility

This requirement is feasible.

4.5.2.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.5.2.7 Validity

This requirement has been deemed valid.

4.5.3 Prioritization

4.5.3.1 Description

The software will attempt to assign a priority on certain deliveries based on expected delivery time and merchant choice.

4.5.3.2 Use Sequence

1. A merchant is able to request a delivery from a supplier with a set priority of low, medium, or high which will pay different rates to drivers.

2. Drivers are able to see the set priority of a job and pick jobs without strict requirements (low-priority jobs) or jobs which need to be delivered in a timely manner (high priority jobs).

3. For high-priority jobs, drivers will need to sign-off on packages delivered due to the increased value of the delivered goods.

4.5.3.3 Possible errors and handling

One possible error which may occur is a invalid priority assignment by a merchant. Care will be taken to ensure that a delivery is correct before it is put onto the jobs drivers can select.

4.5.3.4 Priority

The priority of this requirement is medium-high, as a priority system will greatly increase the usability of the product.

4.5.3.5 Feasibility

This requirement is feasible.

4.5.3.6 Consistency

This requirement does not conflict with the other requirements listed in this SRS.

4.5.3.7 Validity

This requirement has been deemed valid.

5.0 Non-Functional Requirements

5.1 Performance Requirements

The software will update the location of an order viewable by a merchant or supplier user within 3 minutes of the driver updating the location of the order. To ensure the software meets these performance requirements, the development team will conduct proper performance testing.

5.2 Security Requirements

Since there will be sensitive data on this application only the individuals that know their email and password will be able to access their account and data. User data will be stored in the database and all orders pertinent to a user will be linked to them in the database by their user ID. Only users linked to an order will be able to access all information about that order. If secure user logins are implemented, proper care to encrypt the users usernames and passwords should be taken to ensure no unauthorized access to user data is allowed.

5.3 Software Quality Attributes

The software must be robust and performant. The software will be able to adapt to evolving business needs. The software will also need to be robust enough that it will not crash during daily use.

Appendix A: Glossary

API:	Application Programming Interface, used to interact with other software
Driver:	A type of user who will use the mobile application to deliver goods and packages. The driver will deliver goods from the supplier to the merchant.
Goods:	Term to refer to the products being shipped
GPS:	Global Positioning System, used to track the location of devices
User:	The person operating this specific software being described
Supplier:	A type of user for this software, who uses the desktop application to accept deliveries requested by the merchant. The supplier has the goods that the merchant will sell in their storefront.
Merchant:	A type of user for this software, who uses the desktop application to request deliveries to their place of business. An example of a merchant type user would be a Wal-Mart storefront.