Cargo Connect: Improving Airport Freight Flow

System Requirements Specification vs 2.0

SE 6387 Advanced Software Engineering Project
Prof. R. Z. Wenkstern -- UT Dallas

05/09/25

Group *1*
Brad E. Stover
Smit Patel
Anuja FNU
Chijioke Elisha-Wigwe

Revision History

Version	Date	Description	Authors
1.0	02/26/25	Completed initial draft	BS, PS, AF, CE
2.0	05/09/2025	Final draft updated	BS, PS, AF, CE

Contents

1. Introduction

1.1. Purpose

The purpose of this document is to establish the requirements for the Cargo Connect system.

1.2. Scope

The scope of this document is the functional and nonfunctional requirements as they relate to the system. The objective is to explain as clearly and concisely as possible how the system will meet the needs of stakeholders via fulfilling their specific needs. We aim to establish an understanding between our team and the stakeholders of exactly how our solution will work.

1.3. Overview

Our product will assist in improving the traffic flow of logistics carriers by targeting the bottleneck of airport chaos. By utilizing route optimization, real-time information, and schedule modification, we aim to bring logistics flow at airports to its maximal efficiency as much as possible. Integration with the DALI and Airport system will allow for this to transpire.

2. Overall Description

2.1. Product Perspective

The primary purpose of our product is to maximize traffic flow, reduce logjams, and expedite delivery times for logistics carriers. Through features such as parking optimization, dock assignment, and airport entry routing, we will accomplish this objective. Target users include major logistics carriers across the continental United States. Our product will interact with the DALI system, a traffic optimization solution, and the two will be integrated to form a fully cohesive software product.

2.2. Product Functions

The product should display information to the specific truck operator that includes: arrival time to airport, real-time information from DALI about the up-to-date route to the airport, specific airport entry, assigned dock, loading time, designated parking area, DALI-derived information regarding the route to the next destination, and specific airport exit.

Furthermore, the product should receive information from the logistics company such as cargo scheduling, dock assignment data, and parking assignment data. In terms of sending information, the product should relay to the existing airport system the arrival time of freight trucks, their live GPS location, and their target destination.

Lastly, the product should enforce interaction with the DALI system on the user's behalf on the way to the airport as well as the destination.

2.3. User Characteristics

The characteristics of the intended user base revolve around being a major logistics carrier within the continental United States. The intended user is concerned about efficiency and delivery times, and desires

to make sure everything in terms of traffic flows smoothly at airports. They are obligated to serve their customer base with the fastest delivery possible, and they are in need of a solution to provide enhancement in this regard. They are burdened with the inefficiencies of airports (such as delayed packages in transit) and are presumably willing to try something new in a novel attempt to ameliorate this bottleneck situation.

2.4. Constraints

Our team only has approximately four months for product development and launch, which is a relatively short window, and this is a major constraint. We are limited by privacy laws surrounding user data in terms of what we can do with objectives such as integration with package tracking, real-time oversight of drivers, and GPS location of trucks. Additionally, we cannot control the level of user engagement with the DALI system; drivers can override suggestions with their own free will, potentially limiting the effectiveness of our solution.

2.5. Assumptions and Dependencies

We are beholden to the DALI system for much of the underlying functionality of our product, which constitutes a major dependency. Furthermore, we are dependent on the logistics system to provide scheduling, parking assignment, and dock assignment information; if there are any outages or bugs, this will pose a problem for the ongoing functionality of our product. We are assuming that the end customer will adopt our entire solution and implement it fully, rather than simply taking certain parts and putting them into play. We are also assuming full compliance on the part of the drivers in following directions and suggestions given by the system.

3. Hardware Specification

3.1. Hardware Component 1

We have a physical server for DB and a server for computational tasks.

3.1.1. Functionality

Database Server Stores, retrieves, and manages structured cargo data such as tracking, inventory, schedules, and user information. Computation Server Performs route optimization, cargo allocation algorithms, data analytics, and handles API processing or backend logic.

3.1.2. Operational Requirements

Servers must operate 24/7 with minimal downtime. Must support concurrent access by logistics operators, transporters, and system agents. Database servers must support regular backups, replication (if needed), and transaction integrity. Computation servers must have sufficient compute resources (CPU, RAM) to handle route optimization, ETA calculations, and real-time data processing. Secure network communication between both servers.

3.1.3. QoS Requirements

The CPU will be powerful enough to run algorithms in order to provide the utmost accuracy while filtering out local interference. Correspondingly, the RAM capacity will be robust enough to achieve all of these tasks as well.

3.1.4. Parametric Requirements

Phone/Mobile battery capacity must extend to a lifespan of up to 10 years. RAM should be expandable to accommodate new features and algorithms that come with periodic operating system updates [1].

3.1.5. Design Requirements

Modular architecture separating data persistence and processing logic.

4. External Interface Requirements

4. 1. User Interfaces

- **Truck Driver Interface**: Android app for real-time updates, including arrival times, optimized routes suggestions via DALI, airport entry points, dock assignments, parking availability, and loading schedules.
- **Truck Manager Interface**: Web-based dashboard for monitoring truck movements, dock assignments, and real-time data from the airport and DALI systems.
- **Usability**: Simple, fast, and customizable for different airport operations to minimize distractions and improve efficiency.

4. 2. Hardware Interfaces

- **DALI System**: Real-time communication for optimized traffic routing suggestions.
- **Server Hardware**: Cloud-based backend with processing and storage capacity for real-time data.

4. 3. Software Interfaces

- **Airport Management System**: API integration for exchanging data on flight arrival/delays information.
- **DALI System**: Receives traffic and routing data suggestions for optimized paths.
- **API**: Exposes data for integration with external systems (e.g., freight tracking, logistics platforms) using formats like JSON and XML.

4. 4. Communication Protocols and Interfaces

- **HTTPS/REST APIs**: Secure communication between systems.
- **WebSocket/UDP**: Real-time, low-latency updates for truck locations and traffic data.
- **Data Encryption**: Secure, encrypted communication (TLS 1.2 or higher)

5. System Features

5.1. System Feature A

5.1.1. Description

The system will display heads-up information to the truck driver.

5.1.2. Action/result

The driver will be well-informed about decisions to make in terms of navigating the route.

5.1.3. Functional Requirements

1. The system shall show the arrival time to the airport.
2. The system shall show real-time route suggestions from DALI during navigation to the airport
3. The system shall show which entry to enter the airport through.
4. The system shall show the assigned dock and time of loading.
5. The system shall show the designated parking area.
6. The system shall show real-time information from DALI about the optimized route to the next destination.
7. The system shall show the ideal airport exit for the route.

5.1.4. NFR

The system shall continuously display accurate information via constant synchronization.

5.2. System Feature B

5.2.1. Description

The system will send information to and receive information from the existing airport system.

5.2.2. Action/result

The system will be constantly up-to-date in terms of routing information.

5.2.3. Functional Requirements

1. The system shall receive airplane and cargo scheduling information from the airport system.
2. The system shall receive dock and parking assignment data from the airport system.
3. The system shall send arrival time of freight trucks to the airport system.
4. The system shall send the live GPS location of active freight trucks to the airport system.
5. The system shall send the target destination of freight trucks to the airport system.

5.2.4. NFR

The system shall maintain accurate and timely information via integration with the airport system API.

5.3. System Feature C

5.3.1. Description

The system will be integrated with the DALI system.

5.3.2. Action/result

The driver will have real-time and up-to-date navigation information at his/her fingertips.

5.3.3. Functional Requirements

1. The system shall engage the user with the DALI system while traveling to the airport.
2. The system shall engage the user with the DALI system upon exit from the airport.

5.3.4. NFR

The system shall keep a consistent and reliable connection to DALI with absolutely minimal downtime.

6. Non-Functional Requirements

6.1. Product NFR

1. The system shall circumvent service interruptions using backup servers and load balancing.
2. The system shall use encryption to store sensitive user data.
3. The system shall have an intuitive user interface for freight truck drivers to use.
4. The system shall support integrations with third-party service including airport systems.
5. The system shall respond with traffic suggestions within 5 seconds of receiving current location
6. The system shall allow cargo carriers to enter cargo schedule details

6.2. Process NFR

1. The system shall support the JSON data format for proper communication with external systems.
2. The system shall accurately process data, without loss to ensure the reliability of decisions.
3. The system shall ensure a maximum of 100ms latency when interacting with real-time data

Appendix A: Glossary

Term	Definition
Airport system	The existing infrastructure at the airport that manages scheduling, dock assignments, parking assignments, and other logistics-related data.
API	A set of protocols and tools that allow different software systems to communicate with each other, such as Cargo Connect and airport management systems.
Cargo Scheduling	The process of managing and organizing the loading and unloading of cargo at an airport to ensure efficiency.
DALI System	A traffic optimization system that provides real-time route recommendations for freight trucks traveling to and from the airport.
Dock assignment	The process of allocating specific docks at the airport for freight trucks to load and unload cargo.
Freight trucks	A logistics vehicle responsible for transporting cargo to and from the airport.
NFR	Requirements that define system attributes such as performance, security, reliability, and scalability rather than specific functionalities.
Real time data	Live, continuously updated information regarding truck locations, scheduling, and routing to ensure operational efficiency.
Route optimization	The process of determining the most efficient path for freight trucks to take to minimize delays and maximize fuel economy.
Software interface	The means by which different software components interact, including user interfaces and API integrations.

System features	The specific capabilities of the Cargo Connect system, such as real-time navigation, airport system integration, and parking management.
System constraints	Limitations affecting the project, such as development time, legal restrictions, and driver compliance.
Scalability	The ability of the system to expand by adding more servers or features to accommodate increased demand.

Appendix B: References

[1] *What is Parametric Design in Requirements Engineering? – Valispace*. (n.d.). https://www.valispace.com/what-is-parametric-design-in-requirements-engineering/

[2] *PlacePod*. (2024, November 22). PlacePod. https://placepod.com/