

# *Cargo Connect: Improving Airport Freight Flow*

## **Requirements Analysis** Vs 3.0

SE 6387 Advanced Software Engineering Project  
R.Z. Wenkstern

*Date 05/09/2025*

Group 1
FNU ANUJA
SMIT SONESHBHAI PATEL
BRADLEY EVAN STOVER
CHIJOKE ELISHA-WIGWE

**Revision History**

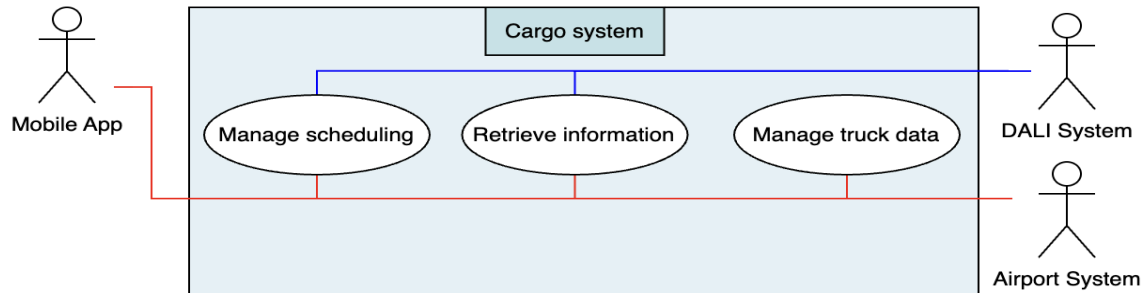
Version	Date	Decryption	Authors
1.0	03/10/25	Completed initial draft	AF, SP, BS, CE
2.0	03/25/25	Modified the draft	AF, SP, BS, CE
3.0	05/09/25	Updated the final document	

## Contents

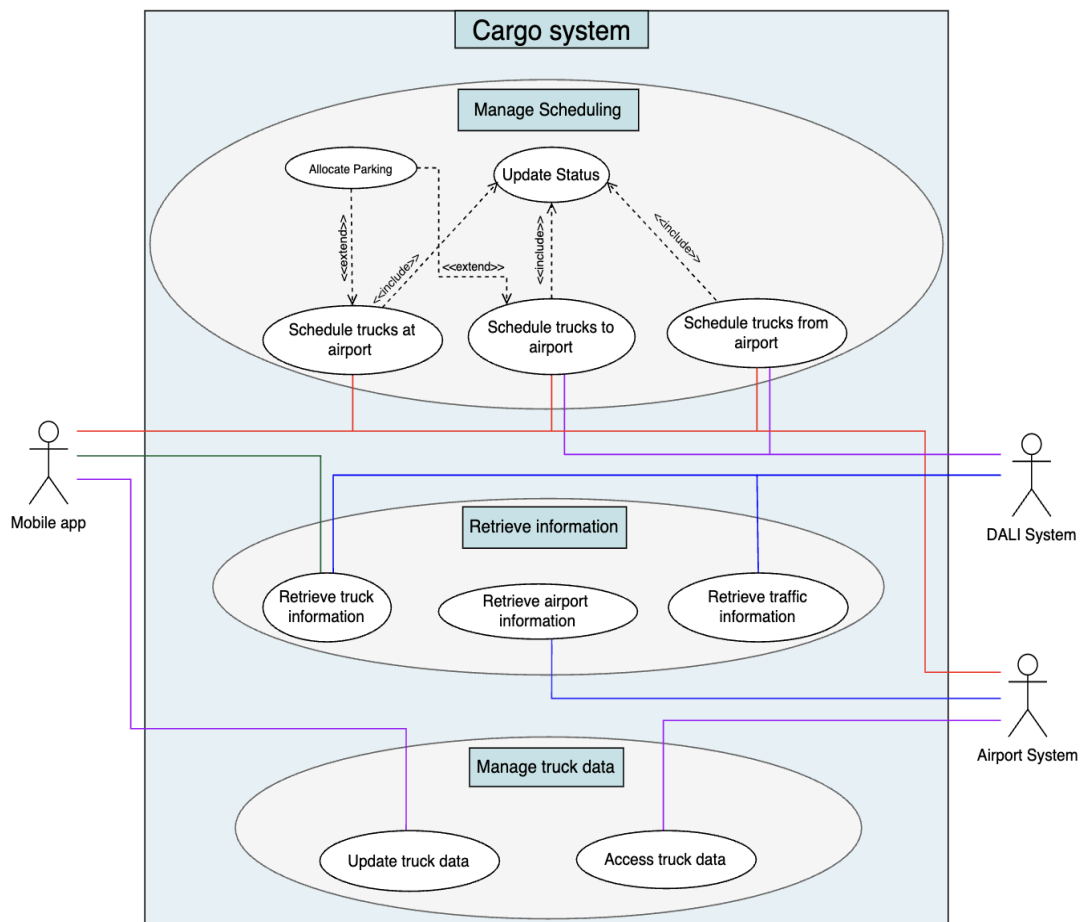
Revision History.....	2
1. Use Case Diagrams.....	4
1.1 Level 0 UCD.....	4
1.2 Level 1 UCD.....	4
2. Use Case Specification in Brief Format.....	5
3. Use Case Prioritization.....	7
4. Use Case 1 Specification in Fully Dressed Format.....	8
4.1. Use Case 1 – Manage Scheduling.....	8
4.2 <<include>> Use Cases.....	10
4.3 <<extends>> Use Cases.....	11
Appendix A: Glossary.....	13
Appendix B: References.....	14

## 1. Use Case Diagrams

### 1.1 Level 0 UCD



### 1.2 Level 1 UCD



## 2. Use Case Specification in Brief Format

Update Truck Data	
<b>ID:</b> UC-01	<b>Type:</b> base
<b>Primary Actor:</b> Mobile App	
<b>Secondary Actor:</b> Airport System	
<b>Brief Description:</b> <ul style="list-style-type: none"> <li>- The Mobile App sends the truck update information to the system.</li> <li>- The Airport system updates the information about the truck</li> </ul>	

Access Truck Data	
<b>ID:</b> UC-02	<b>Type:</b> base
<b>Primary Actor:</b> DALI System	
<b>Secondary Actor:</b> Airport System	
<b>Brief Description:</b> <ul style="list-style-type: none"> <li>- The DALI system requests information about a particular truck.</li> <li>- The Airport system provides information about the requested truck</li> </ul>	

Request Traffic Information	
<b>ID:</b> UC-03	<b>Type:</b> base
<b>Primary Actor:</b> Time System	
<b>Secondary Actor:</b> DALI System	
<b>Brief Description:</b> <ul style="list-style-type: none"> <li>- The system (Time) sends request for traffic information at scheduled time.</li> <li>- The DALI system provides information about the current traffic information</li> </ul>	

Retrieve Airport Information	
<b>ID:</b> UC-04	<b>Type:</b> base
<b>Primary Actor:</b> Time System	
<b>Secondary Actor:</b> Airport System	
<b>Brief Description:</b> <ul style="list-style-type: none"> <li>- The system (Time) sends request for airport information at request time.</li> <li>- The Airport System provides information about the current airport status</li> </ul>	

Schedule Trucks at Airport	
<b>ID:</b> UC-05	<b>Type:</b> base
<b>Primary Actor:</b> Mobile App	
<b>Secondary Actor:</b> Airport System	
<b>Brief Description:</b> <p>The Mobile App contacts the system in order to initiate the scheduling process.</p> <p>The Airport system responds to the scheduling request with either a confirm or a deny.</p>	

Schedule Trucks to Airport
----------------------------

<b>ID:</b> UC-06	<b>Type:</b> base
<b>Primary Actor:</b> Mobile App	
<b>Secondary Actors:</b> Airport System, DALI System	
<b>Brief Description:</b> The Mobile App makes a request to the DALI system to schedule a truck going to the airport. The DALI system contacts the airport system, which confirms or denies the request based on preexisting availability.	

Schedule Trucks from Airport	
<b>ID:</b> UC-07	<b>Type:</b> base
<b>Primary Actor:</b> Mobile App	
<b>Secondary Actors:</b> Airport System, DALI System	
<b>Brief Description:</b> The Mobile App makes a request to the DALI system to schedule a truck going from the airport. The DALI system contacts the airport system, which confirms or denies the request based on whether it can fit into the existing schedule.	

Retrieve Truck Information	
<b>ID:</b> UC-08	<b>Type:</b> base
<b>Primary Actor:</b> Mobile App	
<b>Secondary Actor:</b> DALI System	
<b>Brief Description:</b> The Mobile App requests the status of a particular logistics vehicle to the DALI system. The DALI system responds with an up-to-date status of the truck.	

Update Status	
<b>ID:</b> UC-09	<b>Type:</b> Include
<b>Primary Actor:</b> Mobile App	
<b>Secondary Actor:</b> Airport System, DALI System	
<b>Brief Description:</b> This use case ensures real-time updates of truck scheduling statuses, dock assignments, and route information. It is executed every time trucks are scheduled to, at, or from the airport to maintain accurate synchronization among all involved systems.	

Allocate Parking	
<b>ID:</b> UC-10	<b>Type:</b> Extends
<b>Primary Actor:</b> Mobile App	
<b>Secondary Actor:</b> Airport System, Mobile App	
<b>Brief Description:</b>	

This use case allocates trucks temporarily to a parking lot when docks are congested, or Cargo is delayed.

### 3. Use Case Prioritization

#### High Priority: Manage Scheduling

- This is the most crucial use case as it directly impacts how freight trucks operate within and around the airport, involving scheduling, route optimization, and dock assignment. It has a high dependency on airport systems and regulatory requirements.
- It is essential for truck movement within the airport, determining the correct entry point, assigned dock, and parking area. Moderate complexity but high impact.
- It defines optimized routes from external locations to the airport, leveraging DALI for navigation. High business value but slightly less user impact than scheduling at the airport.
- It also ensures trucks exit the airport through the correct route. It has a significant impact on logistics but slightly lower impact than intra-airport scheduling.

#### Medium Priority: Manage Truck data.

- It involves updating and accessing truck data, which is important for system accuracy but does not directly affect scheduling in real time.
- It allows retrieval of stored truck data but has minimal real-time impact on scheduling operations.

#### Low Priority: Retrieve information.

- It supports decision-making but does not directly impact truck scheduling. Its dependency on external systems (airport, DALI) increases complexity.
- It retrieves airplane and cargo scheduling details, which indirectly help scheduling but are not directly required for truck operations.

Priority	Use Case	Justification
High	Manage Scheduling (Schedule trucks at airport, Schedule trucks to airport, Schedule trucks from airport)	It has the highest priority since it has a direct impact on truck scheduling, airport coordination, and DALI system integration.
Medium	Manage truck data	It ensures truck records are up to date but does not directly impact scheduling.
Low	Retrieve information	It mainly supports other processes rather than directly influencing scheduling or execution.

## 4. Use Case 1 Specification in Fully Dressed Format

### 4.1. Use Case 1

Use Case – Schedule trucks to Airport	
<b>ID:</b> UC-05	<b>Type:</b> Base

**Brief Description:** This use case manages the scheduling of trucks going to the airport. It includes assigning appropriate docks based on real-time dock availability, cargo loading/unloading assignments, and optionally managing parking lot assignments when docks are congested.

**Primary Actors:** Mobile App

**Secondary Actors:** DALI System, Airport System, GoogleMaps

**Preconditions:**

- The truck is registered and authorized within the system.
- Cargo details (type, priority, and loading/unloading requirements) are available.
- Real-time dock availability and parking status information are accessible.

**Main Flow:**

**[include: AuthenticateUser]**

1. The MobileApp requests the cargo schedule list.
  2. The system returns the list of cargo items to the MobileApp.
  3. The user selects an item from the list
  4. The MobileApp opens the CargoDetail screen.
  5. The Mobile App requests the cargo details from the backend
  6. The system returns the cargo details to the MobileApp
  7. The user triggers the button to start navigation
  8. The Mobile App begins navigation to the cargo's destination using Google Maps.
  9. While navigating to the airport (loop every X seconds):
    - 9.1. The MobileApp queries GoogleMaps for the current location.
    - 9.2. GoogleMaps responds with the current location
    - 9.3. The MobileApp sends the current location to the DALI System
    - 9.4. DALI system which responds with traffic alert suggestions.
    - 9.5. The Mobile App displays any alerts to the driver.
    - 9.6. The MobileApp sends the current location and cargo ID to the Airport System.
  10. The MobileApp sends a request for the docking port
- [extension: DelayedCargo, ParkingFull, ParkingUnavailable, DockReassignment]
11. The Airport sends the parking/docking information
  12. The Mobile displays the docking information to the user

**Postconditions:**

- The truck is successfully assigned to an optimal dock.
- Docking schedules are accurately updated within the Airport System.
- Truck operators receive prompt scheduling confirmation notifications.

**Alternative Flows: Dock Congestion (Parking Lot Assignment):**

Delayed Cargo

1. The Airport System sends a **CargoDelayed** message to the Mobile App.
2. The Mobile App informs the user about the delay with an appropriate message.

ParkingUnavailable

1. If a suitable dock is not available due to congestion, but parking is available:
2. The Airport System sends an **AssignParkingLot** message to the Mobile App.
  - 2.1. The Mobile App displays the assigned parking lot and notifies the operator.
3. The truck proceeds to the parking area and awaits further instructions.



**ParkingFull**

1. If both docks and parking lots are unavailable:
  - 1.1. The Airport System responds with a **ParkingFull** message to the Mobile App.
  - 1.2. The Mobile App displays the suggestion and guides the driver to a nearby resting area.
2. The truck halts temporarily and remains on standby.

**DockReassignment**

1. Once a dock becomes available:
  - 1.1. The Airport System sends a **DockReassignment** message to the MobileApp from the parking lot (or resting area) to the optimal dock.
  - 1.2. The Mobile App notifies the truck driver immediately and updates navigation instructions.
2. The navigation resumes toward the assigned dock for unloading or loading.

**Non-Functional Requirements:**

- The system should provide dock availability updates within **5 seconds** of a request.
- The system should support **concurrent scheduling requests** with less than **2%** performance degradation.

**Technology and Data Variation List:**

- The system must integrate with **Google Maps & Directions API** for location updates.
- The dock assignment system should support **API-based communication** with airport systems.
- Data should be stored in a **distributed database** for high availability and fault tolerance.

**Open Issues:**

- How to handle dock reassignments in case of last-minute scheduling changes?

Use Case – Schedule trucks from airport	
<b>ID:</b> UC-07	<b>Type:</b> Base
<b>Brief Description:</b> This use case manages the scheduling of truck departures from the airport. It optimizes truck exit routes based on cargo destination, cargo priority, real-time traffic conditions provided by the DALI system. It ensures that accurate departure information is communicated to relevant systems.	
<b>Primary Actors:</b> Mobile app	
<b>Secondary Actors:</b> DALI System, Airport System, Google Maps	
<b>Preconditions:</b> <ul style="list-style-type: none"> <li>- Cargo ID is known</li> <li>- The truck has completed loading/unloading operations at the dock.</li> <li>- Real-time traffic data is accessible from the DALI System.</li> <li>- The user is authenticated and has access to delivery operations</li> </ul>	
<b>Main Flow:</b> <ol style="list-style-type: none"> <li>1. The user enters a CargoID in the Mobile App.</li> <li>2. The Mobile App sends a request to retrieve the cargo's destination.</li> <li>3. The system fetches the destination from the CargoItem database</li> <li>4. The system returns the destination to the Mobile App.</li> <li>5. The user triggers the button to start navigation</li> <li>6. The Mobile App initiates navigation toward the cargo's destination.</li> <li>7. While navigating (loop every 5 seconds until destination is reached):               <ol style="list-style-type: none"> <li>7.1. The Mobile App queries Google Maps for the current location.</li> <li>7.2. Google Maps returns the current GPS coordinates.</li> <li>7.3. The Mobile App sends the current location update to the DALI system.</li> <li>7.4. The DALI system returns a traffic alert suggestion (if any).</li> <li>7.5. The Mobile App processes the alert and optionally notifies the user.</li> </ol> </li> <li>8. The user marks the delivery as complete.</li> <li>9. The Mobile App sends a delivery status update to the system.</li> <li>10. The system updates the delivery status in the database.</li> <li>11. The Mobile App receives confirmation that the status has been updated.</li> </ol>	
<b>Postconditions:</b> <ul style="list-style-type: none"> <li>- The cargo delivery status is successfully updated in the system.</li> <li>- All location and traffic updates during navigation are processed in real-time.</li> <li>- The cargo and delivery updates are confirmed to the operator in a timely fashion</li> <li>- Truck departure schedules are accurately confirmed.</li> <li>- The Airport System maintains updated and accurate departure records.</li> <li>- Truck operators receive timely, optimized route and departure information.</li> </ul>	
<b>Alternative Flows:</b> <u>TrafficDelay</u> , <u>GPSSignalLost</u> , <u>DALISystemUnreachable</u> <ul style="list-style-type: none"> <li>- If location retrieval fails (e.g., due to GPS signal loss), the system retries and optionally notifies the user of degraded tracking.</li> <li>- If significant traffic delays, the system recalculates alternative routes immediately.</li> <li>- The truck operator is promptly notified of route changes via the Mobile App.</li> </ul>	

<ul style="list-style-type: none"> <li>- If the DALI system is unreachable, navigation continues without traffic alerts, and a fallback alert is displayed</li> </ul>
<b>Non-Functional Requirements:</b> <ul style="list-style-type: none"> <li>- The system must ensure exit route recommendations are updated every 10 seconds.</li> <li>- The system should maintain secure data transmission when communicating truck departure details to the airport system.</li> <li>- The system must ensure that location updates and suggestions are processed within 3 seconds of the request.</li> <li>- The system shall support concurrent location tracking for multiple deliveries.</li> <li>- The MobileApp must gracefully handle intermittent connectivity during navigation.</li> </ul>
<b>Technology and Data Variation List:</b> <ul style="list-style-type: none"> <li>- Integration with DALI API to ensure up-to-date traffic intelligence and alert suggestions.</li> <li>- Google Maps API is used for real-time GPS location.</li> <li>- Cargo delivery status is persisted in a distributed backend database.</li> </ul>
<b>Open Issues:</b> <ul style="list-style-type: none"> <li>- How to handle unexpected road closures affecting exit routes?</li> <li>- How to escalate critical traffic delays to airport authorities for resolution?</li> <li>- Should delivery confirmation require geo-fencing validation (e.g., driver must physically reach a predefined delivery zone)?</li> <li>- Should offline delivery completion be allowed and synchronized later?</li> </ul>

#### 4.2 <<include>> AuthenticateUser

Use Case: Authenticate User	
<b>ID:</b> UC 09	<b>Type:</b> Include
<b>Brief Description:</b> This use case handles the process of authenticating a user into the system using a username and password. It is included in other use cases that require a validated and authorized session before continuing (e.g., scheduling, navigation, delivery updates).	
<b>Primary Actors:</b> Mobile App	
<b>Secondary Actors:</b>	
<b>Preconditions:</b>	
-	
Main Flow:	
<ol style="list-style-type: none"> <li>1. The user opens the Mobile App, which initializes the session.</li> <li>2. The Mobile App prompts the user to enter their username and password.</li> <li>3. The Mobile App sends the credentials to the system.</li> <li>4. The System validates the credentials:</li> </ol>	
[extension: <u>InvalidUsernamePassword, NetworkErrorTimeout</u> ]	
<ol style="list-style-type: none"> <li>5. The System forwards the successful authentication response to the Mobile App.</li> </ol>	
<b>Postconditions:</b>	

- On success: The user is authenticated, the session is initialised, and the user may proceed with the intended operation.
- On failure: No further operations requiring authentication are permitted until login succeeds.

**Alternative Flows:**InvalidUsernamePassword:

1. The system sends an error message back to the Mobile App.
2. The Mobile App displays an error to the user.
3. The truck driver may retry the login or exit the app.

NetworkErrorTimeout:

1. The Mobile App cannot reach the backend due to a network issue.
2. A timeout or connection error is raised.
3. The user is notified with a message like “Unable to connect. Please check your internet connection.”

**Non-Functional Requirements:**

- Authentication should complete within 2 seconds under normal conditions.
- Credentials must be transmitted over HTTPS to ensure security.
- Support for rate-limiting and account lockout must be enforced to prevent brute-force attacks

**Technology and Data Variation List:**

- Communication with UserDB to ensure proper authentication and authorization

**Open Issues:**

- What if the UserDB is not responding?
- What if the Truck driver does not exist?

## Appendix A: Glossary

Term	Definition
Mobile App	Application used by truck operators for scheduling and tracking.
DALI System	System managing real-time traffic and route optimization.
Airport System	System managing airport operations, including truck dock assignments.
Dock assignment	Allocating a specific loading/unloading bay for a truck.
Parking allocation	Assigning temporary parking spots to trucks due to congestion.
API integration	The method of connecting systems for seamless data exchange.
System availabilities	The uptime and responsiveness of the Airport and DALI Systems.
Network failure handling	Contingency measures when connectivity issues arise.
Include	An include relationship is a relationship in which one use case (the base use case) includes the functionality of another use case (the inclusion use case). The include relationship supports the reuse of functionality in a use-case model.
Extend	Extend relationship is to specify that one use case (extension) extends the behavior of another use case (base).

### **Appendix B: References**

Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition). In *Prentice Hall PTR eBooks*.