



Momentum Software Engineering

School of Computing
Science
Simon Fraser University
8888 University Drive
Burnaby, B.C.
Canada. V5A 1S6

CMPT 275 Group T

Momentum Software Engineering's

Flight the Freights Manager Requirement Specification Revision 4

June 19, 2002

Approved by: (By alphabetical order)

Vincent Chu _____
Steven Essen _____
Sung Hwang _____
David Jeong _____



Release History Page

- Revision 1 -May 31, 2002
 -Written jointly by David Jeong, Sung Hwang, Steven Essen
 and Vincent Chu. Compiled together into one document.
- Revision 2 -June 1, 2002
 -Grammar and spelling mistakes are fixed. Each section is
 reviewed by all the members in the team. Minor corrections
 are made in some of the sections.
- Revision 3 -June 3, 2002
 -Final checking by all the team members. A few grammar
 mistakes and a few format inconsistencies are fixed.
- Revision 4 - June 19, 2002
 -Fixed grammar throughout, updated Section 8.1 and added
 Prototype ORD, added MB to glossary, modified state
 diagram and description in Section 5.0, updated Section 6



Table of Contents

- Title Page.....p.4
- Release History Page.....p.5
- Table of Contents.....p.6
- Conventions used in this Document.....p.7
- 1.0 Introduction**
 - 1.1 System Overview.....p.8
 - 1.2 Business Objectives.....p.9
 - 1.3 Software Project Outline.....p.9
- 2.0 Environment**
 - 2.1 Context Diagram and Brief Overview.....p.10
 - 2.2 Hardware Requirement.....p.10
 - 2.3 Operating System Requirement.....p.13
 - 2.4 Database.....p.13
 - 2.5 Development Environment Needed.....p.13
 - 2.6 Maintenance Environment.....p.14
- 3.0 User Interface and Operations**
 - 3.1 User Interface Sophistication Needed.....p.15
 - 3.2 Use Case Scenarios.....p.16
 - 3.3 Scenarios Started by Other Event Sources.....p.20
- 4.0 Retained Data Model**
 - 4.1 Object Relationship Diagram.....p.21
 - 4.2 Discussion of Object Relationship Diagram.....p.23
- 5.0 State Model**
 - 5.1 State Diagrams.....p.25
 - 5.2 State Discussion.....p.26
- 6.0 Data/Process/State Dictionary**
 - 6.1 Object Dictionary.....p.27
 - 6.2 Attributes Dictionary.....p.28
 - 6.3 State Dictionary for the Airfreight/Cargo Object.....p.30
 - 6.4 Event Dictionary for the Airfreight/Cargo Object.....p.30
 - 6.5 Process Dictionary.....p.30
- 7.0 Performance/Capacity**
 - 7.1 Response Time Requirement.....p.31
 - 7.2 Throughput (Rate of Events/Data/Interaction).....p.31
 - 7.3 File Capacity.....p.31
- 8.0 Prototype and Future Releases**
 - 8.1 Features Not Needed in the Prototype.....p.33
 - 8.2 Long Range Plans for Future Releases.....p.33
- 9.0 Acceptance Criterion**
 - 9.1 Hard Requirements.....p.35
 - 9.2 Goals of the System.....p.35
 - 9.3 Acceptance Test Requirements.....p.35
- 10.0 Miscellaneous Requirements**
 - 10.1 Development and Operation Security.....p.36
 - 10.2 Method of Installation.....p.36
 - 10.3 Back-up of System.....p.36
- Appendix A**
 - Glossary of Terms.....p.37



Conventions used in this manual

We will adopt several conventions in this requirement specification:

- Section titles are **large and bold**
- Subsection titles are underlined
- Listed items are **bold but small**
- Special keywords are *italicized*. The definition of these keywords can be found in appendix A. Only the first occurrence of the keyword in the document will be *italicized*.



Section 1.0 – Introduction

1.1 System Overview

The airfreight load management system will manage freight items (both standard containers and odd sized pallets) which are assigned to aircraft flights. This system will be used exclusively by Air Kanata.

This system is needed because, right now, Air Kanata has a paper-based system and the company wants to efficiently automate Air Kanata's airfreight business functions in order to save paper, increase efficiency and productivity, and accurately record and store information.

The seven major functions of the system, briefly, are:

Adding a customer and a cargo item

If the customer is new, then the system will require his/her name and phone number. If the customer has previously dealt with the company, then no other information is needed. Once the customer is in the system, cargo items can be entered.

Assigning the item an airway bill ID number

Once the cargo owner and weight are entered into the system, this function gives the cargo a unique airway bill ID. This ID helps the system to track the cargo and retrieve its information.

Assigning cargo items to a flight

First, a clerk (or user) will check whether a flight has enough capacity to deliver a cargo to the destination. The system will determine this information based on three criteria: the total number of free footprints (based on how many containers that can fit in the free space), the total remaining weight that a certain flight can hold, and the largest rectangular pallet that could be assigned to the remaining space.

If there is room on the aircraft for the designated flight, a clerk will assign the cargo to that flight.

Changing the one aircraft model to another for a particular flight

In case an aircraft for a certain flight has broken down, the system allows the user to bring a replacement aircraft for that flight. Cargo items on the broken aircraft will be transferred to the replacement. If some cargo items can't be placed on the replacement, these cargo items are de-assigned and later, re-assigned to new flights.

Generating a printed report

The system is able to generate a printed report of a flight and all of its cargo items to use by the loaders and pilots.

Deleting a flight and its associated cargo items

Once a flight has been completed, the system could delete the flight and all the cargo that were carried on that flight.



1.2 Business Objectives

Momentum Software Engineering’s customer is Air Kanata, the Canada’s leading cargo shipping company. Currently, their competitors are Federal Express, UPS, and Purolator.

Momentum Software Engineering will develop the airfreight load management system exclusively for Air Kanata. Although Canada is a bilingual country, using both English and French, the system will use the English language throughout since the system will be only used in BC, the Prairies, Ontario, Atlantic Canada, and the territories. A French version could be developed in the future.

1.3 S/W project outline

This project will start promptly on May 23, 2002 because Momentum Software Engineering, the company hired to design and build the system, does not know the airfreight business. Momentum needs some time to research the business rules and understand the improvements which Air Kanata requires.

Momentum Software Engineering’s project manager estimates that the total effort for this project is about 170 person-hours. If 4 people work 1 hour a day, we can complete the project within 56 days.

There could be extra costs for the project. We might need to buy new computer equipments if we later find out that our computer system is not powerful enough to develop the software. Furthermore, we might need to work overtime and pay Momentum’s workers overtime salary.

The following table lists the major milestones and dates of the project:

Task/Milestone	Date
Requirements Specification Document	June 5, 2002
Draft User Manual	June 19, 2002
Design Specification	July 4, 2002
Programming / Unit-testing the various modules	July 18, 2002
Writing Integration and Test Report / Demo system to customer	August 2, 2002



Section 2.0 – Environment

2.1 Context Diagram

The customer merely requires the use of keyboard for data entry to the system. Furthermore, only a screen and printer will be used as the output. This is a single *user* standalone software system, so there is no need to connect it to a computer network.

Please refer to Figure 2.1.1 (Level 0 Context Diagram for our software)

The diagram is rather straightforward – it shows that the keyboard is the input device. The system will process the information and allow the user to perform the desired system functions. As shown on the right hand side of the diagram, the system outputs the response to three different destinations: screen, printer (for direct output), and files (for storing the data).

In the future, however, we can use other means to input to the system. For example, we can use a barcode scanner to scan the airway bill ID number of the cargo. We can also use an electronic balance to automatically enter the weight of the cargo.

Please refer to Figure 2.1.2 (Level 0 Context Diagram for our future software)

2.2 Hardware Requirement

The customer does not require our software system to run on a specific hardware system that satisfies certain requirements. Momentum has decided that the system will be programmed using *Java* and run on a Java virtual machine. As suggested by developer of the language, a computer with the following specifications is recommended for executing Java programs:

- PC system or iMac System
- A minimum of 32 MB RAM (128 MB RAM is recommended)
- A 486 or better processor chip (Pentium 233 MHz or higher is recommended.)

The system does not involve processor-intensive operations. Momentum's analysts suggest that the above requirements are good enough for our purposes.

A mouse is not required for our system; only a standard *QWERTY* keyboard is needed. There are no specific requirements for the monitors/terminals – any monitors that can show texts (even with a single colour) would be fine. However, it is recommended to use a 13" monitor or better in order to make the text legible. The customer wants to use the cheapest printer possible. Momentum's analyst suggests using the cheapest ink-jet printer currently available on the market, and those printers usually come with a parallel cable for the parallel port. Dot-matrix printers are now obsolete since they print too slowly which is unsuitable for the system. Any printer that allows the operating system to send out print commands would suit the software system well.



Momentum Software Engineering

*School of Computing
Science
Simon Fraser University
8888 University Drive
Burnaby, B.C.
Canada. V5A 1S6*



Momentum Software Engineering

*School of Computing
Science
Simon Fraser University
8888 University Drive
Burnaby, B.C.
Canada. V5A 1S6*



In summary, a computer with the following specification is required for our system:

PC system with parallel port for an ink-jet printer

Parallel ink-jet printer with 150dpi and width of 8.5"; 12 pages per minute. Color ink-jet is not required

12" Monitor (with color or not)

A minimum of 32 MB RAM

A QWERTY keyboard

A 486 or better processor chip (Pentium 233 MHz or higher is recommended.)

25 Mb hard disk space

A 3.5" 1.44MB floppy disk drive

A mouse (for running Windows, although the system would not require a mouse)

Note: An iMac system e.g. PowerPC should also work with the system as the system is programmed in a cross platform language, Java. However, the software would not be tested on such system, and would be tested only on IBM compatible PCs.

2.3 Operating System

As indicated in 2.2, the system is written in Java. Java would work on any operating system that satisfies the minimum hardware requirements, given that *Java Virtual Machine* is installed. Windows 95, 98, 98SE or later is recommended. Note that Java Virtual Machine does not come with the Windows XP package. Windows XP users can download an updated version of Java Virtual Machine from www.sun.com

2.4 Database

The system will not be implemented on top of any data base management system. The system will be completely written in Java, and we will develop our own file-handling code.

2.5 Development Environment Needed

When developing the system, Momentum will be using the language Java™ 2.0. We will use Forte™ for Java, an integrated development environment developed by Sun™. Forte™ requires Java 2 Platform. It will be run on an Intel x86 with Windows 95 or later installed.

Forte™ and Java 2 Platform both require the following system requirements:

- 350MHz Pentium II or later
- 128 Mbytes RAM or more
- 110 Mbytes hard disk space or more.



Momentum Software Engineering

*School of Computing
Science
Simon Fraser University
8888 University Drive
Burnaby, B.C.
Canada. V5A 1S6*

2.6 Maintenance Environment

When a new release has come out, the new program package would be sent to the customer in person. The customer merely has to copy all the files from the package into the directory that the customer has our software system installed on. The old program would be overwritten. However, because the data files (i.e. files that contain information about the customers, cargos, aircraft models, and flights) are stored separately from the main program, they would not be lost during system upgrade.

Depending on the nature of upgrade, data files with customer information may or may not require conversion to the new format. However, this conversion will be done automatically with a utility program that comes with the new program package. Unfortunately, the system would have to be shut down temporarily – usually for a few minutes, and the exact length would depend on whether data conversion is required.

When a new software package is released, it will be delivered in person on 1.44MB floppy disks.



Section 3.0 – User Interface and Operation

3.1 User Interface Sophistication Required

The user interface would be a scrolling terminal. The interface would be text-only, and all the functions provided in the form of menu would be accessed via the keyboard. There will be no graphics other than text-based graphics, and animations would be limited to the scrolling of texts.

3.2 Use Case Scenarios

Definition of terms and conventions that will be used in this sub-section only:

Bold text indicates the command that the user will invoke, and the *italicized* text is the brief description of the sequence of prompts the user will see for that particular command.

“**Cargos**” refers to both containers and pallets.

“**Added to the system**” means the profile or information is stored in a file and will be later used to retrieve information.

A note on exception: Since the system will be used by a single person, there is no need to be concerned with multiple users overwriting files at the same time. Also, power failure during the data entry process may cause the lost of any unsaved data.

Operational exceptions but not input/output (I/O) exceptions will be considered; for example, “The name the user enters is longer than the length that the system can handle”, “the format of telephone number/flight designator is invalid” are not covered in this subsection.

Brief list of all the use cases of the system:

- 1) Adding a new customer to a system
- 2) Deleting a customer
- 3) Adding a new cargo item to the system and associate it with a customer
- 4) Adding a new aircraft model type
- 5) Adding a new flight and associate it with an existing plane model type
- 6) Inquiring the capacity on a particular flight
- 7) Assigning a cargo item to a particular flight
- 8) Listing out all the unassigned cargo
- 9) Changing the aircraft model for a flight
- 10) Printing out a report about cargos on a certain flight
- 11) Deleting a flight
- 12) De-assigning cargos from flight
- 13) Deleting a cargo item
- 14) Deleting an aircraft model



Detailed description of all the use cases of the system:

1) Adding a new customer to a system

- *The user is required to enter a unique name that does not already exist in the system.*
- *The user is asked to enter a phone number that fits the correct format.*

Result

- The new customer with the name and phone number will be added to the system.

Exceptions

- A key uniqueness exception occurs if a duplicate name is entered into the system.
- System runs out of disk space when attempting to create and store new customer profile.

2) Deleting a customer

- *The user is required to enter the customer name that needs to be deleted.*

Result

- The customer with that unique name will be deleted. All the cargos that belong to him will also be deleted.

Exceptions

- Referential non-integrity results when the customer name does not exist.
- Referential non-integrity results when some or all of the cargos that are supposedly under the customer's name no longer exist.

3) Adding a new cargo item to the system and associate it with a customer

- *The user is asked to enter the name that the new cargo belongs to.*
- *If the customer name does not already exist, the add function will be called. (Use Case #1)*
- *The customer name should now exist in the system. The user is asked to provide the weight of the cargo.*
- *The user is asked to indicate whether the cargo is of standard size or is an odd size pallet. If it is an odd size pallet then the user is asked to enter the dimensions of the cargo.*

Result

- A customer profile will be created if the customer is new.
- The new cargo with attributes about its weight and dimensions will be added to the system and it's associated with a customer. The new cargo will be assigned a new unique airway bill ID.

Exceptions

- System runs out of disk space when attempting to create and store new customer and/or cargo profile.
- Referential non-integrity results when the customer name does not exist when adding the cargo item.



4) Adding a new aircraft model type

- *The user is asked to enter the model designator.*
- *The user is asked to enter a maximum weight capacity and a maximum length of the plane.*

Result

- A new aircraft model with the appropriate weight capacity and maximum length will be created and saved to file.

Exceptions

- The model designator that is entered already exists in the system. It contradicts the uniqueness of the key.
- System runs out of disk space when the user wants to create a new plane model.

5) Adding a new flight and associate it with an existing plane model type

- *The user is asked to enter the 7-10-character flight designator of the flight.*
- *The user is asked to enter the aircraft model type that the user wants to associate the flight with. If the aircraft model name does not already exist, do #4).*

Result

- The new flight number will now be associated with the aircraft model.

Exceptions

- Referential non-integrity results when the aircraft model assigned to the flight does not exist.
- The flight already exists
- System runs out of disk space when the user wants to create a new flight.

6) Inquiring the capacity on a particular flight

- *The user is asked to enter the 7-10-character flight designator of the flight.*

Result

- The user is presented with the total number of *footprints* free, the total remaining weight that the plane can handle and the largest rectangular pallet that could be assigned to the remaining space (refer to *inquiry report* for format in Appendix A).

Exceptions

- Referential non-integrity results when aircraft model assigned to flight does not exist, and thus the remaining capacity cannot be determined.
- Referential non-integrity also results when enumerating through all the cargos already assigned to the plane, one or more airway bill ID no longer exists.
- The particular flight ID entered does not exist.

7) Assigning a cargo item to a particular flight

- *The user is asked to enter airway bill ID.*
- *The user is asked to enter the 7-10-character flight designator of the flight that the user wants to add the cargo to.*



- *The user is presented with the total number of footprints free, the total remaining weight that the plane can handle and the largest rectangle pallet that could be assigned to the remaining space (refer to inquiry report for format in Appendix A).*
- *If it is possible to add the new cargo, allow the user to add the cargo.*

Result

- A new cargo will be added to the specified flight if the flight has enough space and weight capacity available.

Exceptions

- Referential non-integrity results when airway bill ID refers to a cargo that does not exist.
- Referential non-integrity results when the flight designator refers to a flight that does not exist.
- Referential non-integrity results when the plane model in the assigned flight does not exist.
- An exception also results if there is an assignment of more cargo weight than the assigned aircraft can carry or assignment of more cargo footprint space than the assigned aircraft actually has.

8) Listing out all the unassigned cargo

- *The user is asked to enter the 7-10-character flight designator of the flight.*

Result

- A list of the unassigned cargo will be listed out

Exceptions

- Referential non-integrity results when enumerating through all the cargos already assigned to the plane, one or more airway bill ID no longer exists.
- Referential non-integrity also results when the flight designator does not designate an existing flight.

9) Changing the aircraft model for a flight

- *The user is asked to enter the 7-10-character flight designator of the flight that breaks down.*
- *The user is asked to enter the model designator of the aircraft that the user wants to assign the broken-down flight to.*
- *De-assign any cargo that does not fit on the new aircraft model, and the user is presented with a list of de-assigned cargo.*

Result

- The flight that was broken down will be assigned with a new aircraft model. The flight designator would stay the same. All the cargos that were originally assigned to a broken down flight will stay assigned (if possible) to the flight after the flight is associated with a new associated aircraft model. All the cargos that no longer fits on the new aircraft model will be de-assigned and therefore, become unassigned.



Exceptions

- Referential non-integrity results when enumerating through all the cargos already assigned to the plane and one or more airway bill ID no longer exists.
- Referential non-integrity also results when the aircraft model designator entered does not designate an existing aircraft model.

10) Printing out a report

- *The user is asked to enter the 7-10-character flight designator of the flight.*

Result

- A report with specific format (refer to printed report in Appendix A for format) will be sent to the printer.

Exceptions

- Referential non-integrity results when enumerating through all the cargos already assigned to the plane and one or more airway bill ID no longer exists.
- Referential non-integrity results when the flight designator entered does not designate an existing flight.

11) Deleting a flight

- *The user is asked to enter the 7-10-character flight designator of the flight.*

Result

- The flight with the specified flight designator will be deleted. All the cargos associated with it will also be deleted.

Exceptions

- Referential non-integrity results when enumerating through all the cargos already assigned to the plane, one or more airway bill ID no longer exists.
- Referential non-integrity results when the flight designator entered does not designate an existing flight.

12) De-assigning cargos from flight

- *The user is asked to enter the unique Airway Bill ID of the cargo*

Result

- The cargo with that specific Airway Bill ID will be de-assigned from the flight.

Exceptions

- The Airway Bill ID must actually exist, and the cargo should be assigned to a flight.

13) Deleting a cargo item

- *The user is asked to enter the unique Airway Bill ID of the cargo.*

Result

- The cargo with that specific Airway Bill ID will be deleted.

Exceptions

- The cargo item requested does not exist.



Momentum Software Engineering

*School of Computing
Science
Simon Fraser University
8888 University Drive
Burnaby, B.C.
Canada. V5A 1S6*

- Referential non-integrity results if the cargo is deleted but not de-assigned from the flight.



14) Deleting an aircraft model

- *The user is asked to enter the unique model designator.*

Result

- The plane with the model designator will be deleted from the system.

Exceptions

- The system needs to check and see if the aircraft model that is about to be deleting actually exists in our system.

3.3 Scenarios Started by Other Event Sources

The system is designed to be a single-user system. The system is not expected to be connected to the network. As a result, there are no scenarios that are initiated by network messages received; the user initiates all of the scenarios.



Section 4.0 – Retained Data Model

4.1 Object Relationship Diagram

The Object Relationship Diagram of the airfreight load management system is shown in figure 4.1.1. Please refer to the data dictionary in Section 6 for details on the objects and their attributes.



4.2 Discussion of the Object Relationship Diagram

4.2.1 – Relationships and Cardinalities

There are six main objects in the Object Relationship Diagram: Customer, Flight, Airfreight/Cargo, Pallet, Container, and Aircraft. All of them are closely related to each other.

-Relationship R1

The Customer object relates to the Airfreight/Cargo object through the Customer Name field of the Airfreight/Cargo object and Customer Name field of the Customer object. Each customer can have zero, one, or many cargos within the system. Each cargo must belong to exactly one customer. A customer can be stored in the system and have no cargos – this avoids the user from creating an entirely new record for a previous customer. So when a new customer comes in, he/she will be added to the system.

-Relationship R2

The Airfreight/Cargo object relates to the Flight object through the Flight ID field of both the Airfreight/Cargo and Flight objects. An airfreight can be still unassigned to a flight. Also, a flight can have no cargos because the cargos have not been assigned yet to that flight. Plus, each cargo has only one flight associated with it or else there may be confusion over which flight actually has the cargo. A flight can also have as many cargo items as it is capable of storing.

-Relationship R3

The Flight object relates to the Air Cargo object through the Model ID field of both the Flight and Aircraft Model objects. An aircraft model might not have been assigned to a flight yet and a flight cannot exist without a model. Each aircraft model can belong to zero, one, or more than one flight. In addition, a flight can only have one aircraft model associated with it.

-Relationship between Airfreight/Cargo, Pallet, and Container

The Pallet and Container objects are two types of Airfreight/Cargo with the only difference being that the Pallet has a variable length and width and the Container has a fixed size. Height is not considered in the system.

4.2.2 – Uniqueness of Primary Keys

Customer – As requested by the company, the customer name will be the unique identifier of the customer.

Airfreight/Cargo – The Airway Bill ID are assigned uniquely in order to identify each air cargo (container or pallet).



Flight – As requested by the company, the Flight ID will identify each flight.

Aircraft Model – As requested by the company, the model ID will identify each aircraft model.

4.2.3 – Data Exception Handling

The following exception handling will be done:

Before assigning a flight to an air cargo, the system will check automatically for the existence of the flight before assigning. OR Before assigning an air cargo to the flight, the system will check automatically for the existence of the air cargo before assigning.

An airfreight/cargo must be associated with an existing customer; the system will check for this before adding the cargo to the system.

Before assigning an aircraft model to a flight, the system will also check automatically for the existence of the aircraft model before assigning.

When de-assigning an airfreight/cargo from a particular flight, the airfreight/cargo must exist on that flight. If not, the system, via an error message, will request the user to enter an existing flight.

When changing the aircraft model or inquiring about capacity or requesting a printed report on all the airfreights/cargos on a flight, the flight entered must exist. If not, the system, via an error message, will request the user to enter an existing flight.

Any entry (i.e. length, width, weight, airway bill ID, flight ID, phone number, model ID) that does not conform to the range, size, and/or structure is invalid and the system, via an error message, will inform the user to enter valid information.

Before deleting a customer, flight (and its airfreights/cargos), and an aircraft model, that object must exist. If not, the system, via an error message, will request the user to enter an existing flight.

The system must make sure that an air cargo is not assigned to more than one flight.

If there are two customers with the same name, the user must type a different name for the second customer.

If a customer's name is more than thirty characters, either shorten the first name or use the first initial.

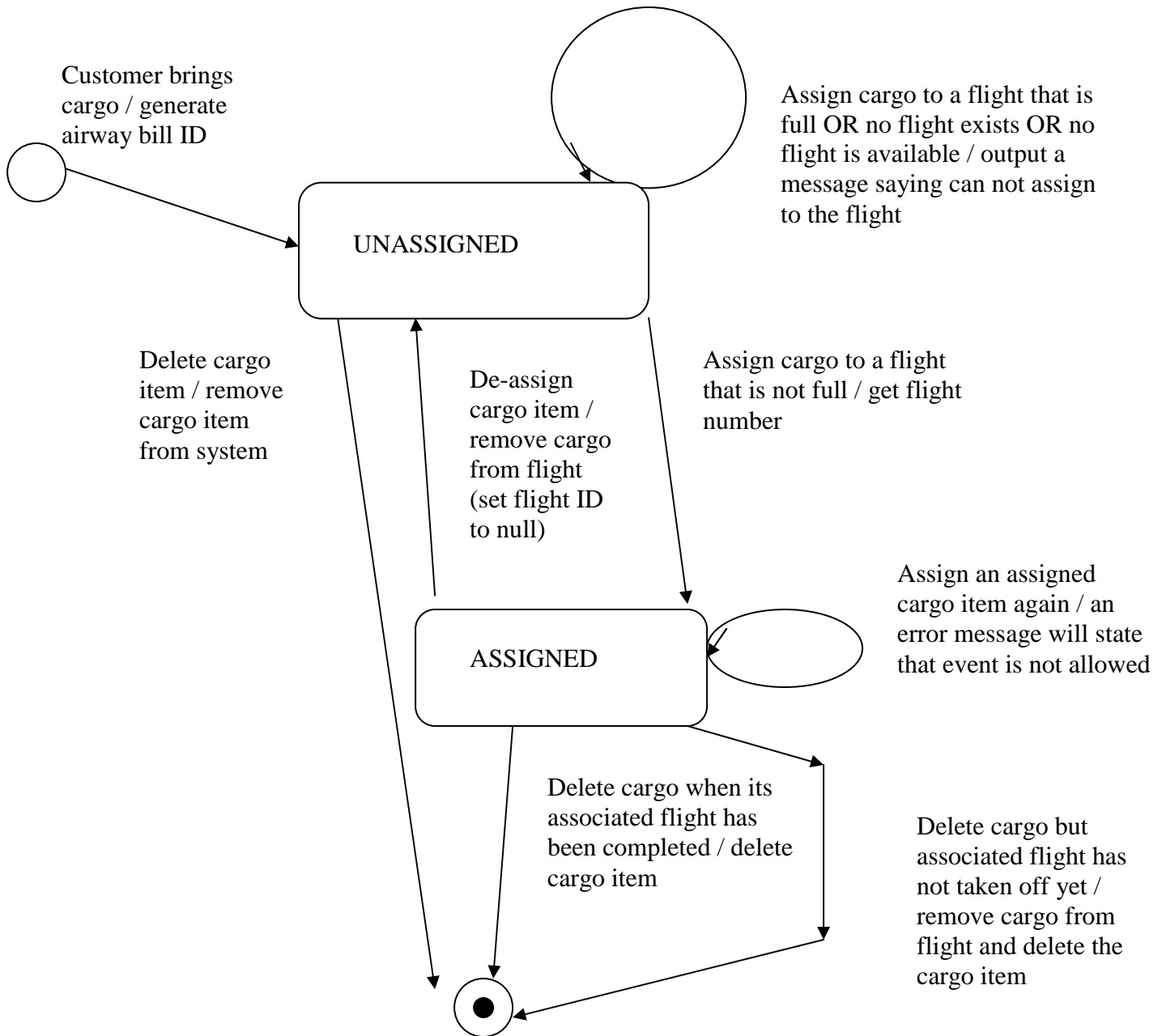
Section 5.0 – State (Control) Model

5.1 State Diagrams

5.1.1 – State Diagram for the Airfreight/Cargo Object

Note: For events not shown in a particular state, it is assumed that the events would not change the state and an error message will be shown.

Figure 5.1.1.1 (below) shows the state diagram for the Airfreight/Cargo object.





5.2 Discussion of the State Diagrams

5.2.1 – Discussion of the State Diagram for the Airfreight/Cargo Object

The lifecycle of the Airfreight/Cargo object goes like this:

The customer brings a cargo item to be shipped via a flight to a destination. The cargo record is created and at this point, the cargo is unassigned. The cargo remains unassigned if flights are unavailable or non-existent. The cargo also remains unassigned if the flight to which cargo is being assigned is full. The unassigned cargo can also be deleted without any problems. If there are flights with available space, the cargo is assigned to one of those flights.

Now, once the cargo is assigned, the following things could occur:

- The aircraft assigned to the flight breaks down so a replacement aircraft model takes over the flight. If there is room on the replacement for the cargo, the cargo is still assigned. However, if this is not the case, the cargo is de-assigned from the flight and becomes unassigned.
- The cargo is deleted before the flight takes off. Then, the cargo is removed from the flight and then, deleted.
- The flight is carried out (eventually) and the cargo is actually shipped/delivered to the destination. This cargo is then, deleted and therefore, removed from the system.



Section 6.0 – Dictionaries

6.1 – Objects

6.1.1 Aircraft Model

Synonym: Aircraft or Model or Airplane

A plane used to fly cargo to a destination; it has a cargo floor to store the cargos: standard containers and pallets. The cargos are packed on the cargo floor with little wasted area. This object stores information about the plane. Its attributes are model ID, maximum weight, and maximum length.

6.1.2 Cargo or Airfreight

Synonym: Freight, Air Cargo

This object stores the airfreight/cargo information. Its attributes are airway bill ID, flight ID, customer name, and weight. Subclasses of cargo are Container and Pallet.

6.1.3 Container

Synonym: Container or Airfreight Container

This object represents containers of a single standard size: 2 meters wide, 1.5 metres high, and 1.75 metres long. These containers are used to carry the airfreight/cargo. It does not have further attributes other than those in the Airfreight object.

6.1.4 Customer

This object represents the person who pays for his/her package to be shipped to a destination. This object stores information about this person. Its attributes are name and phone number.

6.1.5 Flight

Synonym: Airplane Flight

This object stores information about the flight. It represents a route taken by an aircraft model/airplane from one destination to another in order to ship the air cargo/airfreight. Its attributes are flight ID and model ID.

6.1.6 Pallet

Synonym: Odd-Sized Pallet

This object represents a 20 centimetre (0.20 metre) high wood platform on which the cargo is placed. These pallets have various widths and lengths. Its attributes, in addition to those in the Airfreight object, are length and width.



6.2 – Attributes

Conventions used in this subsection (based on Backus-Naur Form or BNF):

digit	(“0” “1” “2” “3” “4” “5” “6” “7” “8” “9”)
letter	(“A” .. “Z” “a” .. “z”)
“ ”	space
“-”	dash/hyphen
“(”	parenthesis - start
)”	parenthesis – end
any	letter, digit, or any other character from the keyboard except a space

6.2.1 Airway Bill ID

This attribute is found in the Airfreight/Cargo object. It is the unique identifier of a freight/cargo.

Type:	integer
Length:	1-6 digits
Range:	1-999999
BNF:	digit {digit} ⁵

6.2.2 Flight ID

This attribute is contained in the Airfreight/Cargo object and Flight object. It is the unique identifier of an airplane flight.

Type:	alphanumeric
Length:	7-10 characters
Format:	flight_number:month day (no space between month and day) where flight number is an integer from 1 to 9999 followed by a colon and then the 3-letter month (either JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, or DEC) and the 2-digit day (01, 02, ..., 30, 31)
BNF:	digit {digit} ³ “:” (“JAN” ”FEB” “MAR” “APR” “MAY” “JUN” “JUL” “AUG” “SEP” “OCT” “NOV” “DEC”) (“0” “1” “2” “3”) digit

6.2.3 Length

This attribute is contained in the Pallet object. It is the length of a pallet.

Type:	positive real number
Units:	metres
Precision:	to the nearest 0.01 metres
Accuracy:	rounded up from measured value by the clerk to precision before entry
Range:	0-199.99
BNF:	{digit} ² digit {“.”}{digit}digit

6.2.4 Maximum Length



This attribute is contained in the Aircraft Model object. It is the maximum length of the cargo floor allowable on an aircraft model.

Type: positive real number
Units: metres
Precision: to the nearest 0.01 metres
Accuracy rounded up from measured value by the clerk to precision before entry
Range: 0-199.99
BNF: {digit}²digit {“.”{digit}digit}

6.2.5 *Maximum Weight*

This attribute is contained in the Aircraft Model object. It is the maximum weight of air cargo/freight allowable on an aircraft model.

Type: positive real number
Units: kilograms
Precision: to the nearest 0.1 kilograms
Accuracy rounded up from measured value by the clerk to precision before entry
Range: 0-999999.9
BNF: {digit}⁵digit {“.”digit}

6.2.6 *Model ID(Model Number)*

This attribute is found in the Flight object and Aircraft Model object. It contains the unique identifier of an aircraft model.

Type: alphanumeric
Length: 1 to 8 characters
BNF: {(any | “ ”)}⁷ any

6.2.7 *Name or Customer Name*

This attribute is contained in the Customer object and Air Freight/Cargo object. It stores the name (first and last) of the customer. This must be unique.

Type: string
Length: 1 to 30 characters
BNF: any {(any | “ ”)}²⁹

6.2.8 *Phone Number or Phone*

This attribute is contained in the Customer object. It stores the phone number of the customer.

Type: alphanumeric
Length: 13 characters
Format: (area code) 3-digit number – 4-digit number
BNF: “(” digit digit digit “)” digit digit digit “-“ digit digit digit digit

6.2.9 *Weight*



This attribute is contained in the Airfreight/Cargo object. It is the weight of a piece of freight/cargo.

Type: positive real number
Units: kilograms
Precision: to the nearest 0.1 kilograms
Accuracy: rounded up from measured value by the clerk to precision before entry
Range: 0-999999.9
BNF: {digit}⁵digit {"."digit}

6.2.10 Width

This attribute is contained in the Pallet object. It is the width of a pallet.

Type: positive real number
Units: metres
Precision: to the nearest 0.01 metres
Accuracy: rounded up from measured value by the clerk to precision before entry
Range: 0-6.00
BNF: digit {"."}{digit} digit}

6.3 – State Dictionary for the Airfreight/Cargo Object

6.3.1 Assigned

This state represents an airfreight/cargo that has been assigned a flight ID. This air freight/cargo is available to be delivered via the assigned flight. If the aircraft model scheduled to fly breaks down, the cargo can also be put on a replacement aircraft model or it can be de-assigned if the replacement aircraft is full. The assigned cargo item is deleted after its associated flight has been completed or it can be deleted before the flight has taken off.

6.3.2 Unassigned

This state represents an airfreight/cargo that has not been assigned a flight ID yet because the customer recently brought it in or the airfreight/cargo was recently de-assigned due to the replacement aircraft cannot store it. This airfreight/cargo is available for assignment to a flight. The unassigned cargo item can also be deleted.

6.4 – Event Dictionary for the Airfreight/Cargo Object

See the arrows in the state diagram in figure 5.1.1.1 for the events and corresponding actions.

6.5 – Process Dictionary

6.5.1 Arranging cargos on an aircraft

The customer demanded that Momentum Software Engineering should provide an *algorithm* for arranging cargos on a flight that works in an efficient manner. Thus, the



algorithm should attempt to maximize the usage of the available area for cargo storage on the aircraft.

Section 7.0 - Performance/Capacity

7.1 Response Times

Using a computer system with Intel Pentium II 233MHz/compatible processor or higher, Momentum Software Engineering guarantees the following response times for our software system:

- Adding a cargo brought in by a customer should take no more than 3 seconds after a clerk finishes entering the cargo information.
- Assigning a cargo to a flight should take no more than 2 seconds.
- Changing the aircraft model on a flight and the subsequent de-assigning of cargos from that flight should take no more than 5 seconds.
- Inquiring about the capacity of the aircraft model on a certain flight should take no more than 5 seconds.

7.2 Throughput

The maximum throughput our system can handle is:

- 100 flights / day
- Adding 20 cargos / minute
- 400 customers / day
- 500 inquiries / day
- Assigning 30 cargos / minute
- 70 printed report lines / minute

7.3 File Capacity

Maximum Likely Number of Records per File (estimated):

Record Files	Calculation	Maximum Number of Records
Container	10 flights/day · 31 days · 300 containers	93000
Pallet	10 flights/day · 31 days · 50 pallets	15500
Customer	77500 + 22500 (allowance)	100000
Flight	10 flights/day · 31 days	310
Aircraft Model		200



Momentum Software Engineering

School of Computing
 Science
 Simon Fraser University
 8888 University Drive
 Burnaby, B.C.
 Canada. V5A 1S6

Approximate Size per Record and Estimated Amount of Disk Storage:

Record Files	Size Per Record	Number Of Records	Disk Storage
Container	$10 \cdot 2 + 4 + 30 \cdot 2 + 4 = 88$ bytes	93000	8184000 bytes
Pallet	$10 \cdot 2 + 4 + 30 \cdot 2 + 4 + 8 = 96$ bytes	15500	1488000 bytes
Customer	$30 \cdot 2 + 13 \cdot 2 = 86$ bytes	100000	8600000 bytes
Flight	$10 \cdot 2 + 8 \cdot 2 = 26$ bytes	310	8060 bytes
Aircraft Model	$8 \cdot 2 + 4 + 4 = 24$ bytes	200	4800 bytes
TOTAL:			18284860 bytes
TOTAL:			Less than 20 MB

Probable Maximum Size of Disk Needed: 25 MB

Estimated size of executable program code: 1000 KB



Section 8.0 – Prototype and Future Releases

8.1 Features Not Needed in the Prototype

- Minimum Requirements

The following items are the minimum requirements of the *prototype*:

- Adding a new customer
- Adding a new cargo (container or pallet)
- Creating a new flight
- Assigning cargo items to a flight
- De-assigning a cargo item from a flight
- Reassign all cargo on a flight to a different flight
- Deleting a cargo item
- Deleting a flight and all its cargo
- Printing a detailed report listing all cargo items on a flight
- Inquiring about the capacity remaining on a flight in terms of area, weight, and dimensions (of the largest pallet that can fit)

- Functions Not Required in Prototype

The following items could be left out in the initial prototype implementation:

- the aircraft model will be left out
- deleting customer
- deleting aircraft model
- listing of all unassigned cargos
- changing a flight's aircraft model when there is de-assigning of cargo(s)
- updating customer, aircraft model, and airfreight/cargo

8.2 Long Range Plans for Future Releases

One possible future feature is generating a printed report containing all cargo items that a particular customer has on a particular flight. The clerk would input the customer name and a flight ID; the clerk is, in effect, requesting all the cargos owned by the customer which are on the specified flight. The customer name and flight ID must be valid; that is, the customer and flight indicated must both exist in the files. If there are no cargos associated with the customer on the flight, then a message would appear stating that no cargo records were found for this customer on a specified flight. If there are cargos, then all the cargos on the flight will be listed in tabular form with columns labelled airway bill ID, weight, and length and width (if the cargo is on a pallet). The table header would state the customer's name and the flight number.

Another possible future feature is building automation into the system, namely automatically assigning cargos to flights without much user intervention. The user would be asked to input not only the weight and dimensions of the cargo but also the destination



Momentum Software Engineering

*School of Computing
Science
Simon Fraser University
8888 University Drive
Burnaby, B.C.
Canada. V5A 1S6*

of the cargo and by which date the cargo should be delivered. Furthermore, the flight entity would also contain the destination and the flight date in addition to the aircraft model. When a customer brings a cargo item to be shipped and provides its destination and arrival deadline, the system will automatically search for flights with flight dates before the imposed deadline and the same destination. Once a match is found, and if the flight still has the weight and space capacity to hold that extra cargo, the cargo will be automatically added to the flight upon the clerk's approval. The future system will keep searching for matching flights until none are available to hold the still unassigned cargos. Therefore, the system would leave the cargo unassigned, and the clerk would be notified about it.



9.0 – Acceptance Criterion

9.1 Hard Requirements

The system must meet the measurable requirements set out in Sections 7.1, 7.2, and 7.3 as well as handle correctly the operational and data exceptions defined in Sections 3.2 and 4.2.3. The required functions, as stated in Section 8.1: Minimum Requirements, must work in the system. The system also must run on hardware based on the requirements set forth in Section 2.2.

9.2 Goals of the System

One goal is that the system must be user friendly. Air Kanata does not want to spend a lot of time and money to train its workers for using the system. The system and its interfaces must be easy to learn and use since the users of the system may have little or no computer knowledge. Also, the user manuals will be clear and concise.

A second goal is that the system increases productivity by at least 20%. Some of the reasons are:

- The clerk can process customer information and cargos much faster than before as well as assign cargos to flight quickly.
- The amount of paperwork required is reduced; mostly every functions are automated by the system.
- Less time can be spent on training of new employees. These workers will be able to learn the airfreight load management system quickly due to the user friendliness of the system.

A third goal is that the system is reliable (i.e. the system does not crash), accurate (because the system deals with very important data), upgradeable (for possible future features), and maintainable.

9.3 Acceptance Test Requirements

The bulk of the system testing will be done by Momentum Software Engineering at their offices located at Simon Fraser University in Burnaby. Management and the employees of Air Kanata are welcome to attend and observe the tests. First, each unit of the system will be tested separately. Then, all units will be combined together and tested in order to see if the system meets the measurable requirements. Note that all of the functions and exceptions handling will be tested as well.

The rest of the testing will take place on-site at Air Kanata. Air Kanata's management and employees will evaluate and test the system to see if the system meets their expectations. They will test the command-line interface, user friendliness of the system, and functionality.



Section 10 – Miscellaneous Items

10.1 Development and Operation Security

Momentum Software Engineering will protect the system code during development by not publicly distributing or displaying the code and keeping the code in a safe place (e.g. a locked safe). Also, Momentum Software Engineering will construct a system where users will need to log into the system with a user ID and password and log off the system. This will maintain data security and prevent intruders from entering the system. Air Kanata should keep the door(s) to the room (where the system resides) locked after hours so no unauthorized person may gain access to the system.

10.2 Method of Installation

Momentum Software Engineering will install the system at Air Kanata's Vancouver office via an executable setup program on a 1.44 MB floppy disk which also contains a copy of the system code. A copy of the installation disk will be given to Air Kanata. Once installation has been completed, Momentum Software Engineering will train the users how to use the system.

10.3 Back-up of System

Air Kanata is responsible for their own backup. Data loss is the sole responsibility of Air Kanata. Momentum Software Engineering would recommend that, at the end of business hours each day, Air Kanata uses the system to back-up (copy) files containing the records of customers, flights, cargo, and aircraft models to a disk or disks. Momentum Software Engineering will not be responsible for any loss of income (or business) or increase in costs due to system failures and the loss or corruption of data.



Appendix A – Glossary of Terms

Algorithm

A sequence of instructions or steps written in a computer program that would complete a required task or solve a particular problem.

Aircraft Model

Please refer to data dictionary in section 6.0.

Aircraft Model ID / Model Number

Please refer to data dictionary in section 6.0.

Airfreight

Used interchangeably with cargo. See cargo.

Air Kanata

The airline freight company for whom the system is being developed

BNF

Abbreviation for Backus-Naur Form. It is a way to specify the exact syntax or format of data attributes.

Cargo / Container / Pallet (Odd size pallets)

Refers to both containers and pallets. Note that cargo, containers, and pallets are NOT terms that can be used interchangeably. Containers are 2 meters wide, 1.5 meters high, and 1.75 meters long. Pallets, or odd-sized pallets, do not have a specific dimension. However, pallets are treated as rectangular, and they are on a 20cm high platform. Please refer to data dictionary in section 6.0 for detail description about cargo/container/pallet.

Clerk

The user of our software system. See user.

Customer

In our requirement specification, customer refers to a person who brings in a cargo for the airline company to ship. Please refer to data dictionary in section 6.0 for its role in our system.

Footprint

Each footprint is 2 meter wide by 1.75 meter long (based on the length and width of the container).



Inquiry Report

For detailed description about the length/range and format for each attributes, please refer to data dictionary. Exceptions are noted in parenthesis.

The inquiry report is displayed when the clerk wants to find out whether a particular cargo can be assigned to a flight. The inquiry report includes:

- Total remaining weight on a flight before the aircraft model overloads (in kilograms to the nearest 0.1 kg)
- Largest area on an aircraft for a pallet (the length and width in metres (to the nearest 0.01 m) and footprints (to the nearest footprint))
- Total remaining area on a flight that are available in terms of the number of containers based on the footprint concept.

iMac

Macintosh Computer.

JAVA

A cross-platform language developed by Sun™

JAVA Virtual Machine

Java requires something called the Virtual Machine to execute programs (class files) written in Java. The Virtual Machine is a piece of software that can be downloaded freely off the Internet.

Keyboard

The keyboard is used to enter the following information:

- New Customer Name
- Phone Number of the Customer
- Dimensions of the cargo
- Aircraft Model ID
- Flight Designator
- Airway Bill ID
- Weight of the Cargos
- Aircraft Model Maximum Length
- Aircraft Model Maximum Weight



MB

This stand for megabyte. A term used to describe capacity of memory (RAM) or disks.

Parallel Port

The computer sends information to other devices through ports. Usually, printers are connected to a computer through parallel ports.

PC

Personal Computer.

Printed Report

The printed report (about a flight) includes:

For detailed description about the length/range and format for each attributes, please refer to data dictionary. Exceptions are noted in parenthesis.

- Flight designator
- Aircraft model that the flight is associated with
- A list of cargos that the flight contains
 - Airway Bill ID
 - Standard container or pallet
 - Weight (in kilograms (kg) and pounds (lbs))
 - Dimensions
 - Customer name
 - Customer's phone number
- Total number of containers
- Total number of pallets
- Total weight in kilograms (kg) and pounds (lbs)

Note 1: the precision of pounds is to the nearest unit lbs.

Note 2: the accuracy of the conversion from kg to lbs is ± 10 lbs.

Note 3: the cargo list can be unordered

Prototype

A simple/small program that shows the customer the software system can do the job and that the software company is following the customer's requirements. It has a minimal subset of functions that allows the customer of the software to have a "sneak preview".



QWERTY keyboard

Standard keyboard with keys QWERT and Y on the upper left hand section of the keys for keying in alphabets.

RAM

Random Access Memory for temporary storage of data inside the computer during operation

Referential Non-integrity

Referential non-integrity occurs when attributes in an object does not refer to existing objects. For example, every flight is associated with an aircraft model. The aircraft model that the flight is associated with is stored as an attribute in the flight. However, if the aircraft model is accidentally deleted and the flight is not updated, the flight now refers to a non-existing aircraft model – thus, referential non-integrity occurs.

Screen

The screen displays the following information to the user

- Customer Profile
 - Displaying customer name
 - Displaying customer phone number
- User Interface
 - Displaying the functions the user could perform
- Plane Model Profile
 - Displaying plane model designator
 - Displaying plane maximum width and maximum length
- Flight Model Profile
 - Displaying the flight designator
 - Displaying the list of cargos that the flight contain
- List of unassigned Cargo
- Inquiry Report

System / Software System

Refers to Momentum Software Engineering's Flight the Freights Manager

User

A person who uses our software system, namely the clerk. The user will enter the information required and perform functions provided by the system.