**Lambert Air Management System**

Requirements Specification

Josh Snider, Matthew Lavin, and Andrew Daur

February 15th, 2022 – March 6TH, 2022

Logo

Description automatically generated

**Project Requirements Specification (RS) -- Deliverable 2**

Contents

[1. System Overview 3](#_Toc97217064)

[1.1. Problem Definition 3](#_Toc97217065)

[1.2. Project Problem Strategy 4](#_Toc97217066)

[1.3. System Architecture 4](#_Toc97217067)

[1.3.1. Information Topology Chart (ITC) 4](#_Toc97217068)

[1.3.2. Object Flow Diagram (OFD) 6](#_Toc97217069)

[1.4. Subsystems 7](#_Toc97217070)

[1.4.1. Central Database 7](#_Toc97217071)

[1.4.2. Port Information 7](#_Toc97217072)

[1.4.3. Airline & Aircraft 7](#_Toc97217073)

[1.4.4. Flights 7](#_Toc97217074)

[2. Storage Requirements 8](#_Toc97217075)

[2.1. Storage List 12](#_Toc97217076)

[3. Operational Requirements 23](#_Toc97217077)

[4. System Rules 26](#_Toc97217078)

[4.1. Referential & Data Integrity Rules 26](#_Toc97217079)

[4.2. Procedural Rules & System Security Rules 28](#_Toc97217080)

[4.2.1. System Security 28](#_Toc97217081)

[4.3. Derivation Rules 30](#_Toc97217082)

[5. Revised Project Schedule 31](#_Toc97217083)

[6. Concluding Remarks 32](#_Toc97217084)

[6.1. Other Considerations 32](#_Toc97217085)

[7. Appendices 34](#_Toc97217086)

# System Overview

Refer to Initial System Requirement document for more detailed information on the following.

## Problem Definition

The airline industry has gone through significant upheaval in recent years; challenges in staffing and consumer demand have placed considerable strain on even well-established airlines and airports. Daily operation of an airport involves the management of traffic and staff from multiple airlines operating numerous flights, detailed procedures are required to maintain efficiency. Poor airline service due to delays reflects poorly on both us and the firms we work with. The air travel industry continues to grow steadily as more and more demand hits the markets, airport staffing and scheduling gets increasingly difficult as airports try to expand to accept more customers and keep all related departments in the green. Below is a figure from World Bank that shows very clearly how year after year the air travel industry must react to increasing demand with exception of events that ban air travel en masse. If there is no solution proposed beyond simply more manpower and more processing stations for various tasks the air travel industry will very quickly run out of resources and manpower to accomplish the tasks needed to sustain the demand professionally enough that it is deemed acceptable.

Customers that are waiting around costs money for all involved airports and the air travel industry at large, a streamlined solution is required. Take for example a flight ‘x’ that is to be boarded in an hour as it has been scheduled, but for whatever reason a flight crew member ends up missing their check in time. We now need another crew member to fill in for flight ‘x’, but first you have to figure out who exactly the flight crew is by identifying and labeling the flight appropriately, then cross reference the crew information and scheduling to find a replacement, you also may want to go through delayed or canceled flights that may have crew still available to fill your need. You could do all of that with people power, but you may find that task quite daunting especially if the airports grow in size, but as discussed previously it simply has its limitation in terms of speed and efficiency, so it is not sustainable long term.

## Project Problem Strategy

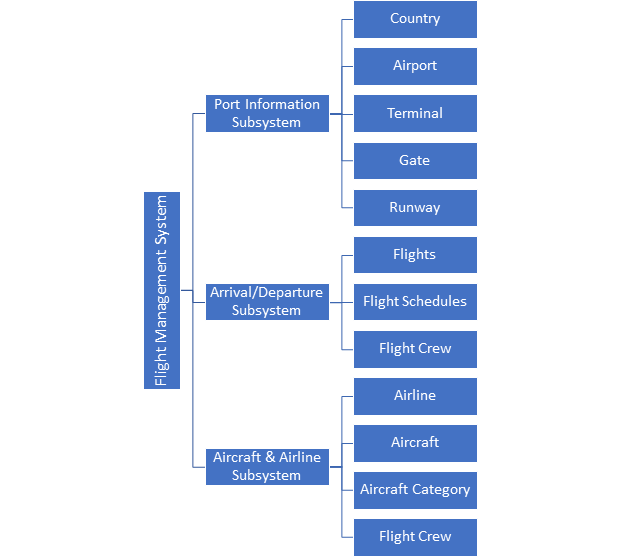
We propose an automated solution to this problem with the development of a Flight Management System (FMS) which can be configured for any port in any country. This solution will reduce manpower requirements for all departments involved in flight logistics, streamlining the decision-making process for management by providing concise and accurate information about flights, crew and boarding/disembarking controls to a variety of airports internationally. This will standardize the workflow of all impacted airlines and participating ports around the world. This process is fast enough to communicate all the way around the world in a matter of seconds about a specific crew member, a flight, a port, an aircraft type, etc. It is a time saving, cost saving, space saving, and highly efficient system if done properly. Air traffic control becomes more fluid, and they are notified about flights of concern far faster and more consistently than by traditional means, thereby avoiding more potential expenses for accidents of varying kinds. This system will be on windows devices but strive for platform independence and as such will be able to communicate with any other windows system with this software solution. This will utilize a MySQL database to serve and maintain information and form necessary relationships between our entities of note (the systems we’re keeping track of). This system is to be titled, Lambert Air Management (LAM) and may be used interchangeably with FMS for the purposes of this document.

## System Architecture

### Information Topology Chart (ITC)

The entities of the FMS are divided into three subsystems: the Port Information Subsystem (PIS), Arrival/Departure Subsystem (ADS), and Aircraft & Airline Subsystem (AAS).

* Airport infrastructure, including airports themselves, terminals, gates, and runways are defined within the PIS.
* The ADS contains information on arrivals, departures, recurring scheduled flights, and their associated flight crew.
* The AAS stores information on airlines, their respective fleets of aircraft, and their flight crews, as well as general-purpose information on models of aircraft.



**Figure 1.3.1 Information Topology Chart (ITC)**

### Object Flow Diagram (OFD)

The flow of data within the FMS is simple: the system is divided into three subsystems, each of which are connected to the central relational database. Desynchronization of data is avoided by having all subsystems’ queries being done on a shared database, eliminating the need to synchronize between distributed systems.

**Figure 1.3.2 Object Flow Diagram (OFD)**

## Subsystems

### Central Database

All system data is stored within a central relational database. This will be implemented in MySQL, a freely-available database which provides full SQL capabilities.

### Port Information

Holds all information on countries, participating airports along with their runways, terminals, and gates. This will be the first subsystem to be populated with data, as the other two build off of its information.

### Airline & Aircraft

Holds all information on participating airlines, as well as their flight crew and fleet of aircraft. This is the second system to be populated, as airlines and aircraft are rarely added to the system and rely on the existence of corresponding data in the Port Information Subsystem.

### Flights

Holds information on flights and flight schedules, as well as flight crew. Flights are stored as both recurring scheduled flights, as well as individual flights with a specific plane, set of crew, and time.

# Storage Requirements

**Figure 2.1.1 Entity Relationship Diagram(ERD)**

Through these relationship tables, one can accurately and comprehensively define the structure of the database. Attributes are not shown in this specific ERD, they will be added for our complex database specifications document. For now, including the attributes is impractical in conveying what is necessary.

Diagram

Description automatically generated

It is anticipated that the system will contain the following main information entities, along with their attribute codes and primary keys. The conventions used are summarized below:

* Each information entity referenced is identified by a reference code and a descriptive name.
* For each entity, the attributes (data elements) to be stored are identified below, as well as their assigned key.
* The entities as presented will easily transition into a set of normalized relations in a normalized relational database.
* Data elements that will be implemented as foreign keys, in the normalized relational database are identified between the “{ }” brackets, labeling which entity they refer to.
* Data elements that will be used as primary key attributes (or part of the primary key) at the database implementation time are also identified by parenthesized comments.
* For each entity, a brief comment describing the data being stored will be provided.

1. **Countries**: For defining information countries. Essential attributed include:
   * Country Code (CnCode)
   * System Name (CnName)
   * System Abbreviations (CnAbbr)

Primary Key: {CnCode}

1. **Participating Ports**: for defining information on participating ports. Essential attributes include:
   * Airport Code (PortCode)
   * Airport Name (PortName)
   * Airport Short Name (PortShort)
   * Related Country (PortCnCode **reference to E01**)

Primary Key: {PortCode}

1. **Airlines**: for identification of all airlines that use that port. Essential attributes include:
   * Airline Identification Code (AlCode)
   * Airline Name (AlName)
   * Airline Host Country (AlCnCode **reference to E01**)
   * Airline Base Port (AlPortCode **references E02**)
   * Airline Headquarters Address Line 1 (AlAddrLine1)
   * Airline Headquarters City (AlCity)
   * Airline Telephone Number(s) (AlTelephone)
   * Airline Contact Person(s) (AlContact)

Primary Key: {AlCode}

1. **Aircrafts**: for definition of all aircrafts. Essential attributes include:
   * Aircraft Code (AcCode)
   * Aircraft Name (AcName)
   * Aircraft Description (AcModelDscr)
   * Aircraft Seating (AcCapacity)
   * Aircraft Features (AcFeatures)
   * Aircraft Type Code (AcTypeCD **references O11**)
   * Aircraft Host Airline (AcAlCode **references E03**)

Primary Key: {AcCode}

1. **Runways**: for defining each runway at each port. Essential attributes include:
   * Runway Code (RwCode)
   * Runway Name (RwName)
   * Runway Length (RwLength)
   * Runway Width (RwWidth)
   * Runway Port (RwPortCode **references E02**)

Primary Key: {RwCode}

1. **Terminals**: for defining each terminal at each port. Essential attributes include:
   * Terminal Code (TCode)
   * Terminal Name (TName)
   * Terminal Letter (TLetter)
   * Terminal Port (TPortCode **references E02**)

Primary Key: {TCode}

1. **Gates**: for defining each gate at each port. Essential attributes include:
   * Gate Code (GCode)
   * Gate Namee (GName)
   * Gate Letter (GLetter)
   * Gate Port (GPortCode **references E02**)

Primary Key: {GCode}

1. **Flight Officials**: for specifying flight attendants and pilots each airline has. Essential attributes indlude:
   * Official Code (OCode)
   * Official First Name (OFName)
   * Official Middle Initial (OMIntl)
   * Official Last Name (OLName)
   * Official’s Related Airline (OAICode **references E03**)

Primary Key: {OCode}

1. **Flights**: for specifying all flights. Essential attributes include:
   * Flight Number (FNumber)
   * Flight Source Port (FSPortCode **references E02**)
   * Flight Destination POrt (FDPortCode **references E02**)
   * Flight Intermediary Port (FIPortCode **references E02**)
   * Flight’s Related Airline (FAICode **references E03**)
   * Flight’s Related Aircraft (FAcCode **references E04**)

Primary Key: {FNumber}

1. **Arrival/Departure Schedule**: for tracking all arrivals. Essential attributes include:
   * Flight Number (FNumber **references E09**)
   * Flight Date (FDate)
   * Flight First Pilot (FFPCode **references E08**)
   * Flight Second Pilot (FSPCode **references E08**)
   * Flight Main Attendant (FAtt1Code **references E08**)
   * Flight Other Attendant 2 (FAtt2Code **references E08**)
   * Flight Other Attendant 3 (FAtt3Code **references E08**)
   * Flight Other Attendant 4 (FAtt4Code **references E08**)
   * Flight Other Attendant 5 (FAtt5Code **references E08**)
   * Flight Other Attendant 6 (FAtt6Code **references E08**)
   * Flight In/Out Flag (FIOFlag)
   * Flight On-time Flag (FOnTime)

Primary Key: {FNumber, FDate}

1. **Aircraft Types**: for different types of aircrafts. Essential attributes include:
   * Aircraft Type Code (AcTypeCD)
   * Aircraft Type (AcTypeDescr)
   * Aircraft Seating Capacity (AcTypeSeating)

Primary Key: {AcTypeCD)

## Storage List

**Figure 2.2.1: Naming Conventions from** *Software Enginerring: A Methodical Approach.*



**Figure 2.2.2: Storage List for the Lambert Air Management System.**

|  |
| --- |
| **E01 – Countries [LAM\_countries\_BR]** |
| **Attributes:**   1. Country Code [ CnCode ] [ Char(3) ] 2. Country Name [ CnName ] [ VarChar(100) ] 3. Country Abbreviation [ CnAbbr ] [ VarChar(10) ] |
| **Comments:**  Entity to keep track of all participating countries for LAM system, referenced by many other entities for location information. Country Code and Country Abbreviation are not necessarily the same and could be confused with each other. Consider Country Code as alphanumeric representation of country, whereas country abbreviation is just alphabetic. Country Name is NOT NULL. CnCode can either be straight numeric or alphanumeric, recommend alphanumeric for easier interpretation.  Example: [ CnCode=’NA2’ ; CnAbbr=’USA’ ] representing United States. |

|  |
| --- |
| **E02 – Participating Ports [LAM\_participatingports\_BR]** |
| **Attributes:**   1. Port Code [ PortCode ] [ Char(6) ] 2. Port Full Name [ PortFullName ] [ VarChar(100) ] 3. Port Name [ PortName ] [ VarChar(50) ] 4. Port Country Code [ PortCnCode ] [ Char(3) ] {**Refers E01.CnCode**} |
| **Comments:**  Entity to keep track of all participating ports across various countries, shorthand name ‘PortName’ added for convenience of query. PortFullName is NOT NULL. |

|  |
| --- |
| **E03 – Airlines [LAM\_airlines\_BR]** |
| **Attributes:**   1. Airline ID Code [ AL\_IDCode] [ Char(9) ] 2. Airline Name [ AL\_Name ] [ VarChar(100) ] 3. Airline Callsign [ AL\_Callsign] [VARCHAR(32)] 4. Airline Host Country [ AL\_CnCode ] [ Char(3) ] {**Refers E01.CnCode**} 5. Airline Host Port [ AL\_PortCode ] [ Char(6) ] {**Refers E02.PortCode**} 6. Airline Primary Address [ AL\_HQAddress1 ] [ VarChar(100) ] 7. Airline Host City [ AL\_HQCity ] [ VarChar(50) ] 8. Airline Contact Phone [ AL\_HQPhone ] [ Number/Decimal(11,0) ] 9. Airline Contact Name [ AL\_Contact ] [ VarChar(100) ] |
| **Comments:**  Entity to keep track of all participating airlines, their hosting country/port, and contact information. Primary key is defined as ‘PPPPPPXXX’ where ‘P’ is the host port code, and ‘X’ is alphanumeric. AL\_Name and AL\_HQAddress1 and AL\_HQPhone are NOT NULL. |

|  |
| --- |
| **E04 – Aircrafts [LAM\_aircrafts\_BR]** |
| **Attributes:**   1. Aircraft Code [ AC\_Code ] [ Char(10) ] 2. Aircraft Name [ AC\_Name ] [ VarChar(100) ] 3. Aircraft Description [ AC\_Desc ] [ VarChar(500) ] 4. Aircraft Capacity [ AC\_NumSeats ] [ Number/Decimal(3,0) ] 5. Aircraft FeatureList [ AC\_FeatureList ] [ VarChar(MAX)/Text ] 6. Aircraft Type ID [ AC\_ATypeCode ] [ Char(3) ] {**Refers E11.ATypeCode**} 7. Aircraft Host Airline [ AC\_AL\_IDCode ] [ Char(9) ] {**Refers E03.AL\_IDCode**} |
| **Comments:**  Entity to keep track of all aircraft as children of participating airlines with a corresponding recognizable aircraft type. Feature list could be a number of different data types including: File,VarChar,Text,Blob depending on preference of storage medium for this information; For maximum compatibility across all airlines, countries, and ports, as well as minimal digital storage requirements for database, recommend either VarChar with max length or Text type. Aircraft Name and capacity NOT NULL. |

|  |
| --- |
| **E05 – Runways [LAM\_runways\_BR]** |
| **Attributes:**   1. Runway Code[ RwCode ] [ Char(8) ] 2. Runway Name [ RwName ] [ VarChar(100) ] 3. Runway Length [ RwLength ] [ Number/Decimal(4,0) ] 4. Runway Width [ RwWidth ] [Number/Decimal(4,0) ] 5. Runway Host Port [ RwPortCode ] [ Char(6) ] {**Refers E02.PortCode**} |
| **Comments:**  Entity to keep track of all runways as child of hosting port, with runway dimensions. Runway Code defined as ‘AAAAAANN’ where ‘A’ is a alphabet letter and ‘N’ is a number. Runway Name NOT NULL. |

|  |
| --- |
| **E06 – Terminals [LAM\_terminals\_BR]** |
| **Attributes:**   1. Terminal Code [ TermCode] [ Char(8) ] 2. Terminal Name [ TermName ] [ VarChar(100) ] 3. Terminal Status [ TermStatus ] [ Char(1) ] 4. Terminal Host Port [ TermPortCode ] [ Char(6) ] {**Refers E02.PortCode**} |
| **Comments:**  Entity to keep track of all terminals and associated port, complete with terminal status. TermCode defined as ‘AAAAAANN’ where ‘A’ is an alphabet letter and ‘N’ is a number. TermStatus should be open(O), closed(C), delayed(D), or unknown(?). Terminal Name NOT NULL. |

|  |
| --- |
| **E07 – Gates [LAM\_gates\_BR]** |
| **Attributes:**   1. Gate Code [ GCode ] [ Char(8) ] 2. Gate Name [ GName ] [ VarChar(100) ] 3. Gate Status [ GStatus] [ Char(1) ] 4. Gate Host Port [ GPortCode ] [ Char(6) ] {**Refers E02.PortCode**} |
| **Comments:**  Entity to keep track of all gates including associated port and status. GCode is defined as ‘AAAAAANN’ where ‘A’ is an alphabet letter and ‘N’ is a number. GStatus should be open(O), closed(C), delayed(D), or unknown(?). Gate name NOT NULL. |

|  |
| --- |
| **E08 – Flight Officials [LAM\_flightofficials\_BR]** |
| **Attributes:**   1. Flight Official ID Code [ FOffCode ] [ Char(6) ] 2. Flight Official First Name [ FOffFName ] [ VarChar(50) ] 3. Flight Official Middle Initial [ FOffMInitial ] [ VarChar(2) ] 4. Flight Official Family Surname [ FOffLName ] [ VarChar(50) ] 5. Flight Official Host Airline [ FOff\_ALCode ] [ Char(9) ] {**Refers E03.AL\_IDCode**} |
| **Comments:**  Entity to keep track of all airline staff related to flights and which airline that employs them. FOffCode defined as ‘XXXXXX’ where ‘X’ is alphanumeric. All employees must have at least a first name and family surname to be valid. |

|  |
| --- |
| **E09 – Flights [LAM\_flights\_BR]** |
| **Attributes:**   1. Flight ID Number [ FlyCode ] [ Char(6) ] 2. Flight Origin Port [ FlySrcPortCode ] [ Char(6) ] {**Refers E02.PortCode**} 3. Flight Destination Port [ FlyDestPortCode ] [ Char(6) ] {**Refers E02.PortCode**} 4. Flight Intermediate Port [ FlyInterPortCode ] [ Char(6) ] {**Refers E02.PortCode**} 5. Flight Host Airline [ FlyALCode ] [ Char(9) ] {**Refers E03.AL\_IDCode**} 6. Flight Aircraft ID [ FlyAC\_Code ] [ Char(10) ] {**Refers E04.AC\_Code**} |
| **Comments:**  Entity to keep track of all flights, and key stopping locations of the flight path. All flights will require flight officials and have at minimum two interacting ports of note. FlyCode defined as ‘XXXXXX’ where ‘X’ is alphanumeric. |

|  |
| --- |
| **E10 – Scheduling [LAM\_scheduling\_BR]** |
| **Attributes:**   1. Flight ID Number [ SchFlyCode] [ Char(6) ] {**Refers E09.FlyCode**} 2. Flight Date [ SchDepartDate ] [ Number/Decimal(8,0) ] 3. Flight Primary Pilot 1 [ SchFOff\_P1 ] [ Char(6) ] {**Refers E08.FOffCode**} 4. Flight Primary Pilot 2 [ SchFOff\_P2 ] [ Char(6) ] {**Refers E08.FOffCode**} 5. Flight Primary Crew Member 1 [ SchFOff\_A1 ] [ Char(6) ] {**Refers E08.FOffCode**} 6. Flight Primary Crew Member 2 [ SchFOff\_A2 ] [ Char(6) ] {**Refers E08.FOffCode**} 7. Flight Primary Crew Member 3 [ SchFOff\_A3 ] [ Char(6) ] {**Refers E08.FOffCode**} 8. Flight Secondary Crew Member 1 [ SchFOff\_A4 ] [ Char(6) ] {**Refers E08.FOffCode**} 9. Flight Secondary Crew Member 2 [ SchFOff\_A5 ] [ Char(6) ] {**Refers E08.FOffCode**} 10. Flight Secondary Crew Member 3 [ SchFOff\_A6 ] [ Char(6) ] {**Refers E08.FOffCode**} 11. Is Flight Departed [ SchFlightOut ] [ Number/equivalent type(1) ] 12. Is Flight On Schedule [ SchFlightOnTime ] [ Number/equivalent type(1) ] |
| **Comments:**  Entity to keep track of flight schedule and associated crew members for flights. All flights MUST HAVE 2 pilots and at minimum 3 assisting crew members to be considered launch-ready, all primary attributes must be filled. Primary Key is composite key of #1 and #2, the flight in question and the date of departure in that order. Date should be in the format yyyymmdd (year,month,day). SchFlightOut and SchFlightOnTime are effectively boolean flags, but it is easier and more cross-compatible to do a number with a length of 1 for this purpose, just flip between ‘0’ and ‘1’ for false and true respectively. |

|  |
| --- |
| **E11 – Aircraft Types [LAM\_aircrafttypes\_BR]** |
| **Attributes:**   1. Aircraft Type Code [ ATypeCode ] [ CHAR(3) ] 2. Aircraft Type Description [ ATypeDesc] [ VarChar(MAX)/Text ] 3. Aircraft Seating Capacity [ATypeSeating] [NUMBER(4)] |
| **Comments:**  Entity to keep record of recognized aircraft types, from which E04-Aircrafts is derived. Various vehicle types are catalogued here to officially stamp them as approved aircrafts for international traveling and identify features of an aircraft type that may be beneficial for specific flight routes or use-cases.  Example: Smaller aircraft types/classes may be better on fuel and hence better suited for longer distance travel to destination ports that don’t have as much demand from consumers as other ports. |

# Operational Requirements

The underlying structure of the system will be implemented within a relational database system, likely MySQL/MariaDB. All interaction with the system will be done through a separate interface, implemented using a programming language to be determined. This separation protects the database from the risks of direct access, as program logic can be used to separate users from data and implement database protections. This will also provide flexibility, as interfaces can be customized to meet the needs of various users.

1. For each entity, provide an interface within the software to retrieve its information, as well as modification, deletion, and retrieval of inquiries on the entity.
2. For complex data entry tasks, such as entering the details of a flight within the system, offer a more advanced interface designed to ease the process.
3. For public-facing logical views and other automatically generated reports, give more sophisticated output than raw SQL data. This will range from an automatically generated arrival/departure board to viewing a report of an employee’s work schedule.

The flight management system will be designed with inter-airport cooperation in mind, with the goal being a design which can be implemented at multiple sites in a way which enables data-sharing.

A read-only frontend is another potential component of the system, as the arrival and departure board of an airport is desirable to have offered over the internet in a performant way. This public front-end can be implemented as a periodic job to perform the following:

1. Retrieve the current state of the Arrival/Departure logical view
2. Push a “rendered” version of this data, i.e. formatted as a table, to a web server or other service
3. Finish job, to be repeated when prompted.

**Figure 3.1.1: IMS User Operations List**

This list contains all accessible methods for each data entity to be called when a user wishes to do something with the database. This could mean anything from having a report printout, inquiring about specific data, adding, deleting, or modifying content within the database that the user has access to. Note, the user will likely have limited access to only the tables and entities applicable to their tasks. Varies based on user access ‘tier’. More on this in section ###### System Security.

|  |  |
| --- | --- |
| **Lambert Air Management System** | |
| **Object Type** | **Operations** |
| **E01** Countries | **Add** Countries **[E01\_A]**  **Modify** Participating Ports **[E01\_M]**  **Delete** Countries **[E01\_Z]**  **Inquire** Countries **[E01\_I]** |
| **E02** Participating Ports | **Add** Participating Ports **[E02\_A]**  **Modify** Participating Ports **[E02\_M]**  **Delete** Participating Ports **[E02\_Z]**  **Inquire** Participating Ports **[E02\_I]**  **Report** Participating Ports **[E02\_P]** |
| **E03** Airline | **Add** Airline **[E03\_A]**  **Modify** Airline **[E03\_M]**  **Delete** Airline **[E03\_Z]**  **Inquire** Airline **[E03\_I]**  **Report** Airline **[E03\_P]** |
| **E04** Aircraft | **Add** Aircraft **[E04\_A]**  **Modify** Aircraft **[E04\_M]**  **Delete** Aircraft **[E04\_Z]**  **Inquire** Aircraft **[E04\_I]**  **Report** Aircraft **[E04\_P]** |
| **E05** Runway | **Add** Runway **[E05\_A]**  **Modify** Runway **[E05\_M]**  **Delete** Runway **[E05\_Z]**  **Inquire** Runway **[E05\_I]**  **Report** Runway **[E05\_P]** |
| **E06** Terminal | **Add** Terminal **[E06\_A]**  **Modify** Terminal **[E06\_M]**  **Delete** Terminal **[E06\_Z]**  **Inquire** Terminal **[E06\_I]**  **Report** Terminal **[E06\_P]** |
| **E07** Gate | **Add** Gate **[E07\_A]**  **Modify** Gate **[E07\_M]**  **Delete** Gate **[E07\_Z]**  **Inquire** Gate **[E07\_I]**  **Report** Gate **[E07\_P]** |
| **E08** Flight Officials | **Add** Flight Officials **[E08\_A]**  **Modify** Flight Officials **[E08\_M]**  **Delete** Flight Officials **[E08\_Z]**  **Inquire** Flight Officials **[E08\_I]**  **Report** Flight Officials **[E08\_P]** |
| **E09** Flight Definitions | **Add** Flight Definitions **[E09\_A]**  **Modify** Flight Definitions **[E09\_M]**  **Delete** Flight Definitions **[E09\_Z]**  **Inquire** Flight Definitions **[E09\_I]**  **Report** Flight Definitions **[E09\_P]** |
| **E10** Arrival/Departure Schedule | **Add** Arrival/Departure Schedule **[E10\_A]**  **Modify** Arrival/Departure Sche[…] **[E10\_M]**  **Delete** Arrival/Departure Schedule **[E10\_Z]**  **Inquire** Arrival/Departure Schedule **[E10\_I]**  **Report** Arrival/Departure Schedule **[E10\_P]** |
| **E11** Aircraft Types | **Add** Aircraft Types **[E11\_A]**  **Modify** Aircraft Types **[E11\_M]**  **Delete** Aircraft Types **[E11\_Z]**  **Inquire** Aircraft Types **[E11\_I]**  **Report** Aircraft Types **[E11\_P]** |

# 

# System Rules

There are a few main types of system rules that will be covered here:

* Referential & Data Integrity Rules
  + Within the data integrity rules the treatment of foreign keys will be detailed. Specifically, each foreign key in a referencing entity must have values drawn from the entity that it references. The only exception to this rule is the instance where the foreign key values is null. This is highlighted in the following subsection.
* Procedural/Security & Derivation Rules
  + Procedural and derivation rules relate to how the software system will actually work. This is further detailed in section 4.2

## Referential & Data Integrity Rules

Various restrictions on data must be applied beyond the relationship restrictions implied by foreign key. These rules ensure that entities are labeled in keeping with existing international standards, such as for codes of countries, airlines, and airplane models.

1. **Country Definitions**
   1. Country code and name must be non-blank and non-null
   2. Country code must conform to international standards (ISO 3166-1 alpha-3), i.e. Great Britain is **GBR**.
2. **Port Definitions**
   1. Port code & name must be non-blank, non-null
   2. Port code must be drawn from international standards (ICAO or IATA?), e.g. John F. Kennedy Int’l Airport is **JFK**.
   3. Port’s Related Country must be drawn from E01
3. **Airline Definitions**
   1. Airline code & name must be non-blank, non-null
   2. Airline Identification Code must be drawn from international standards, e.g. American Airlines is **AAL**.
   3. Airline’s Host Country must be drawn from E01
   4. Airline’s Base Port must be drawn from E02
4. **Aircraft Definitions**
   1. Aircraft code & name must be non-blank, non-null
   2. Aircraft’s Type Code must be drawn from E011
   3. Airline’s Host Airline must be drawn from E03
5. **Runway Definitions**
   1. Runway code & name must be non-blank, non-null
   2. Runway’s Port Code must be drawn from E02
   3. Runway length and width shall be defined in meters
6. **Terminal Definitions**
   1. Terminal code & name must be non-blank, non-null
   2. Terminal’s Port Code must be drawn from E02
7. **Gate Definitions**
   1. Gate code & name must be non-blank, non-null
   2. Gate’s Port Code must be drawn from E02
8. **Flight Officials Definitions**
   1. Flight Officials code & name must be non-blank, non-null
   2. Flight Official’s Related Airline code must be drawn from E03
9. **Flight Definitions**
   1. Flight code & name must be non-blank, non-null
   2. Flight Source Port must be drawn from E02
   3. Flight Destination Port must be drawn from E02
   4. Flight Intermediary Port must be drawn from E02
   5. Flight Source, Destination, & Intermediary Ports must not be duplicates
   6. Flight’s Related Airline must be drawn from E03
   7. Flights Related Aircraft must be drawn from E04
10. **Arrival/Departure Schedule Definitions**
    1. Arrival/Departure Schedule code & name must be non-blank, non-null
    2. Flight Number must be drawn from E09
    3. Flight First Pilot must be drawn from E08
    4. Flight Second Pilot must be drawn from E08
    5. Flight Main Attendant must be drawn from E08
    6. Flight Other Attendant 2 must be drawn from E08
    7. Flight Other Attendant 3 must be drawn from E08
    8. Flight Other Attendant 4 must be drawn from E08
    9. Flight Other Attendant 5 must be drawn from E08
    10. Flight Other Attendant 6 must be drawn from E08
    11. Flight crew FKs must be unique per-flight, i.e. no duplicates within a single row
    12. Flight crew FK referencing a specific employee cannot exist on two simultaneous flights
11. **Aircraft Type Definitions**
    1. Aircraft Type code & name must be non-blank, non-null
    2. Aircraft type code must be drawn from IATA lists, e.g. a Boeing 737-800 is **738**.

## Procedural Rules & System Security Rules

* PIS: Logistics data for ports of interest must be specified first and be completed before all other subsystems. This includes countries, airports, runways, terminals, and gates. This information will stay largely static and be widely referenced, only changing with opening or closure of an airport or renovations.
* AAS: The system must be updated with detailed information on all airlines and aircraft, after the completion of the Port Information Subsystem
* FIS: Must be entered and updated daily, such that the latest information is always available to end users. This data will be added and changed the most rapidly.

### System Security

LAM will have different user access levels depending on the mission of the user. For example, an air traffic controlling officer will want access to specific entities relating to that task, but has no need to access the entire system. Of course, an airline or port may override this by providing them access to an administrator account if for some reason they disagree, but in this way we can highlight specific areas of interest to the individual user and make sure to keep database tampering to a minimum. This has a positive net effect on overall system security by limiting points of failure in the system. The basic access levels should be as follows:

* **Scheduler** – basic access level to see and edit entities relating to flight schedules and flights themselves only. Full access to: ‘E08 – Flight Officials’, ‘E09 – Flights’, and ‘E10 -- Scheduling’. Grant access to logical views containing the same information in a different layout. No other access needed, reports to Department Management.
* **Controller** – access level to see and edit entities relating to security checkpoints and flight management as pertains to collision prevention. Full access to: ‘E04 -- Aircrafts’, ‘E05 -- Runways’, ‘E06 -- Terminals’, ‘E07 -- Gates’, ‘E11 -- Aircraft Types’. Grant access to logical views containing the same information in a different layout. Read only (view) access to ‘E10 -- Scheduling’ so the controller can see what flights are on deck or supposed to be launching currently and track their flight ID. Reports to Department Management users.
* **Department Management** – higher access level to see and edit entities relating to active ports/airlines and their children entities relating to employees. Full access to ‘E01 -- Countries’, ‘E02 -- Participating Ports’, ‘E03 -- Airlines’, ‘E08 -- Flight Officials’. Grant access to logical views containing the same information in a different layout. Read only (view) access to ‘E10 -- Scheduling’ to view available flights for an airline, it’s status, and it’s crew setup. Department management oversees Controller and Scheduler users, and directly informs them of changes to be made in the system outside of full access capabilities. Reports to Administrators.
* **Administrators** – Highest access level of user, granted access to full system control to make emergency changes or adjustments outside of the purview of other access levels. Full access to logical views used by other user tiers, and may create new logical views for them to use. To be clear, this user is NOT supposed to delete or remove default tables or views, instead they may depreciate them and grant other users access to personalized logical views or tables with updated and personalized information that they deem more beneficial, but they do have the power to do so. This user may personalize their own database by adding extra tables to keep track of specific information for their particular employer and in general ensure maintenance and updating of the system at large. It is this user’s responsibility to make sure LAM continues to deliver functionality to their employer (or from the LAM system’s perspective, a pool of users).

## Derivation Rules

Derivation rules related to how certain calculated data will be derived. Following is a schedule of logical views that will be referenced from the various INQUIRE/REPORT operations to provide useful outputs from the system.

**Figure 4.12: Schedule of Required Logical Views**

|  |  |
| --- | --- |
| Descriptive Name | Summarized Specification |
| Port Connections | Joins Port Definition (E02) to Country Definition (E01) |
| Airline Connections | Joins Airline (E03) to Port (E02) and Country (E01) |
| Crew of Specific Flight | Joins Flight Crew (E08) with Flights (E09) and Scheduling (E10) |
| Crew Schedule | Joins Flight Crew (E08) with Scheduling (E10) |
| Airframe Schedule | Joins Aircraft (E04) with Aircraft Type (E11), Flights (E09), and Scheduling (E10) |
| Arrival/Departure Board | Joins Flights (E09), Scheduling (E10), Airline (E03), Port (E02), Terminal (E06), and Gate (E07) |
| Per-Gate Schedule | Joins Flights (E09) with Scheduling (E10), Airline (E03), and Gate (E07) |
| Delays & Cancellations | Arrival/Departure board with added selection of only cancelled or delayed flights. |

# Revised Project Schedule

No changes have been made to the time of initial software delivery; the revised schedule now accounts for the initial maintenance and support which will be needed at launch.

|  |  |  |
| --- | --- | --- |
| Activity # | Activity Description | Weeks |
| A | Initial System Requirement | 1 |
| B | Requirements Specification | 2 |
| C | Design Specification | 3 |
| D | Database Creation | 3 |
| E | Prototype Development | 10 |
| F | Software Documentation | 3 |
| G | Software Install/Delivery/Training | 2 |
| H | Initial Maintenance | 1 |
|  | **Total duration** | **25** |
| Assumption: Project team of three software engineers | | |

# Concluding Remarks

This concludes the Requirement Specification document. All of the information included in this document will be vital to designing our database. The next deliverable will expand upon the Design Specifications of the Lambert Air Management System.

## Other Considerations

There was some discussion amongst the team about particular details of subsystems that we could add into our database, and lots of ideas were logically sound, but it is a delicate balance for this system where we must be careful not to step on the toes of how any one port or airline wishes to do business. To be specific, there was the case about whether or not to include seat class information as an attribute of ‘E09 Flights’ in LAM to assist in tracking how many of each type of seat, 1st class, common, etc, the airline could sell to it’s customers. While this information would be handy to airlines to not have to track by other means, there was fear that some airlines may want to split the seats up differently depending on their customer demographics/ demand for a specific flight in which case having this hard coded into the database may be detrimental for those airlines as they may choose to simply ignore this portion of information for their purposes or use a different management system that suits them better. Or take for example, what if an airline wanted more information about their faculty stored in ‘E08 Flight Officials’ table of the database such as pay rate or job title, this is another case where while it is true the database could assist in this way, what if an airline has a different set of titles that don’t conform to the rest of the system, or maybe somebody has a title change event that now must be updated in the database for it to work properly? Trying to keep this balance of functionality versus flexibility in mind, the LAM system should always strive more for the latter while fulfilling time sensitive needs for airlines and ports to achieve a fast and fully featured toolset for all potential system users, regardless of the specifics of how they do business. This way, any airline and port may use this product to maintain daily needs in a timely and efficient manner, while leaving the door open for them to supplement and maintain specifics of their system and business in any way they feel best suits them.

In short, this management system should be refined to serve any and all ports or airlines that wish to streamline their daily operation work to take a heavy workload off their backs by solving the problem (outlined in 1.1) while maintaining the flexibility to accommodate different management styles and business details within the industry. In that way this system alienates minimal potential users and reduces the risk and nightmare of having to personalize the system for different users based on their needs. We know that users need to keep track of these basic entities (outlined in 2.1, ESG) to achieve functionality; regardless of how the users wish to accomplish individual tasks within these subsystems, it is left for them to decide what is best, this system is merely a vessel of the needed information to make the decision. If it is discovered that there is other entities or attributes that will improve the functionality or features of this system across the entire industry without personalizing for any one user, maintaining the flexibility described here, this system can always be updated for all users without losing what is already included.

# Appendices

[1] Foster, Elvis C. *Software Engineering: A Methodical Approach. Second Edition.* CRC Press. 2022

Link to the updated text: [Software Engineering. A Methodical Approach Second Edition](https://cloudflare-ipfs.com/ipfs/bafykbzacea7dqh624ol4dil3oh3q2aecequ4dbzjxpkdadahdguds2bao2suq?filename=Elvis%20C.%20Towle%20Jr.%2C%20Bradford%20A.%20Foster%20-%20Software%20Engineering_%20A%20Methodical%20Approach-CRC%20Press%20%282021%29.pdf)

[2] One World – Nations Online Project. *Country Codes List*. Copyright 1998-2022 nationsonline.org

Link to their website: [Country Code List](https://www.nationsonline.org/oneworld/country_code_list.htm#G)

[3] World Bank Group. “International Tourism, Number of Arrivals.” Data. Accessed March 3, 2022. https://data.worldbank.org/indicator/ST.INT.ARVL.

[4] U.S. Department of Transportation, Federal Aviation Administration. “*Air Traffic Organization Policy*”. October 15, 2015.