**System Requirements Specification**

**1.0 - Introduction**

**1.1 Purpose**

The purpose of this document is to specify the functional and non-functional requirements of an electronic flight management system. This document aims to be a comprehensive overview of the system that illustrates requirements, constraints and the external factors that could affect the system. This document will be useful for the system development team and potential end users.

**1.2 Scope**

The system to be constructed will be a ‘Flight Management System’.

Three categories of users will utilise this system: Customers, staff members and travel agents.

Staff members are further broken down into 6 categories:

* ***Staff*:** A generic category that will handle many customer related tasks.
* ***Booking manager:*** A staff member who handles several more complex tasks relating to the flight bookings of customers.
* ***Flight Manager:*** A staff member who handles tasks relating to the management of airport information, aircraft information, route information, scheduling information and flight information.
* **Profile manager:** A staff member who handles tasks relating to the creation and modification of customer and staff profiles.
* **Service Manager:** A staff member who handles tasks relating to the available flight service menus and service item information (e.g. food and drink services.).
* **Administration:** A staff member who has permission to perform all staff and management tasks. Also performs tasks relating to staff member creation and permissions.

In order to properly manage the full functionality, flight management system is comprised of 5 sub-systems:

* **Reservation system:** This system handles all functionality relating to the booking process. All flight searching and flight selection options are handled here.
* **Profile System:** This system handles all customer profile creation and modifications as well as travel agent profile creation and modifications.
* **Flight System:** This system handles all tasks relating to the creation and modifications of airport information, aircraft information, route information, scheduling information and flight information.
* **Report system:** This system handles all report creation tasks. Information is obtained from all other sub-systems and compiled to form reports for management level staff members.
* **Service System:** This system handles the creation and modification of flight services information and service item information (e.g. food and drink services.).

1.3 References:

* Volere Requirements template. Available from http://www.volere.co.uk/template.htm

**2.0 - Overview / Description**

**2.1 Product Perspective:**

Currently all booking and management activities occur manually. This product aims to create a more efficient method of performing these actions, and in turn saving the company time, money and effort.

**2.2 System interfaces:**

The system is self-contained. No external systems are required for operation. The database system has been custom built specifically for this program based on text file access, and is contained within the system.

Deployment will require the program to be loaded on a windows or linux based machine within the company before operation.

**2.2.1 User Interfaces:**

The user interfaces to be provided to customers must be a text based console interface. User interfaces given to customers provide different options that other user interfaces. Customers have the lowest system access rights in the system.

The user interface provided to staff members must be a text based console interface. The UI here provides many of the same options provided to the customer. However, some additional options must be provided for the staff member to effectively perform their jobs (e.g. search for customers, edit bookings).

The user interfaces provided to management level staff members must be a text based console interface. The UI here will give the user options specific to the tasks each manager needs to perform. Primarily it will allow the manager access to their respective sub-system.

The user interface provided to the Administration staff must be a text based console interface. The UI here will provide full access to all staff and management functionalities, as well some extra privileges such as the ability to grant access to staff members.

The user interface provided to the travel agents must be a text bases console interface. The UI here will provide access to functionality travel agents need to perform their role.

**2.2.2 Hardware Interfaces:**

The program must be able to execute on any windows or linux based pc within the company.

**2.2.3 Software Interfaces**:

The system is self-contained. No external software is required for operation.

**2.2.4 Communication Interfaces:**

The system is not web enabled, nor does it include client/server architecture. No Communication is required outside of the system.

**2.2.5 Memory constraints:**

No memory constraints are strictly defined. However, the program needs to run as efficiently as possible, and perform tasks in a prompt manner.

**2.2.6 Operations:**

The Flight Management System user interfaces must be easy to use. Only a minimum amount of training should be needed to perform actions with the system. No specific skills should be needed to utilise the system.

Installation should be simple enough for the system administrator to handle. No additional training should be required to set up the system.

The database should import and store information from text files. Also it should be robust enough to handle improper shut down cases (such as power loss) as effectively as possible.

**2.3 Product Functions:**

One of the main functions of the Flight Management system is to manage the flight company’s flight information and provide a means for users to utilise, create, delete and modify the information.

A customer should be able to use these primary functions:

* Book flights for themselves or others. A booking consists of a Customer, flight, route, services, seating, price, and arrival/departure times.
* View booked flights
* Manage their own personal details
* Manage their own payment details

A staff member should be able to use these primary functions:

* Book flights for a customer. A booking consists of a Customer, flight, route, services, seating, price, and arrival/departure times.
* View customers booked flights.
* View a customer’s personal details.
* Modify a customer’s personal details
* Modify a customer’s booking details
* Search for a customer.

A Booking manager should be able to use these functions:

* Modify a customer’s flight details.
* Modify customers seating if destination seat is occupied.
* Set customers no-fly/watch status.
* View system reports relating to the reservation system.

A Service Manager should be able to use the following functions:

* Modify service items details.
* Add new service items to the database.
* Add new flight services to the database.
* Remove service items from the database.
* Remove flight services from a flight.
* View reports relating to the service system.

A Profile Manager should be able to use the following functions:

* Modify personal details of a customer.
* Modify a customer’s frequent flier points.
* View reports relating to the profile system.
* Modify travel agents details.
* Modify staff details.
* Give discounts to travel agents.
* Give discounts to customers.

A Flight Manager should be able to use these functions:

* Add/remove aircrafts, schedules, airports, routes, and flights.
* Modify aircrafts, schedules, airports, routes, and flights.
* View system reports relating to the flight system.

An Administrator should be able to perform these functions:

* Access all staff functionalities.
* Access all management level functionality.
* Grant access to staff members.

A Travel agent should be able to perform these functions:

* Must be able to book flights for customers.
* Must be able to edit own details.
* View all bookings made for customers.
* Modify existing bookings made for customers.

All user functionalities have been defined from client meeting requirements elicitation so that the Flight Management System can perform all tasks required by the client.

**2.4 User Characteristics:**

* Staff members have good knowledge about the tasks and processes of customer activities
* Booking managers have a high level knowledge in the tasks and processes of customer booking flights.
* Profile managers have high level knowledge of the tasks and processes of system profiles and their management.
* Service Managers have high level knowledge of the tasks and processes of the service information system.
* Flight Managers have a high level knowledge of the tasks and processes of the flight management system.
* Administrators have good knowledge of all tasks and processes of all system users and sub systems. Also has knowledge of system installations.
* Travel agents have good knowledge of the tasks and processes of booking customers.

**2.5 Constraints:**

The system should obey the following constraints:

* User Authentication: For a user to book a flight or for a staff member to use the system, The system should require a log In and authentication process.
* All users should have appropriate permissions when accessing the system.
* Database must be robust. It should be easy for users to save/backup the database to prevent corruption/data loss.
* Database integrity. The Flight management system involves several sub-systems that access the database. It is imperative that these sub-systems interface properly to maintain database integrity.
* System must be documented, developed and released by the end of may 2015.

**2.6 Assumptions and Dependencies:**

The following assumptions and dependencies are defined:

* All customers must have an email address.
* All Travel agents must have an email address.

**3. Specific Requirements**

Each requirement has a priority level associated with them. Priority levels are defined as:

* Critical – Highest level. Defines core functionality.
* Essential – Secondary to critical. Should be implemented immediately after core functionality has been implemented.
* Low – Non essential functions. System can operate without these functions. Should be implemented after
* Optional – lowest importance priority. It would be desirable but not necessary to implement.

**3.1 User Management Functions.**

This section defines functionalities provided for customers, staff members, travel agents and Administrators.

**3.1.1 Customer Functions:**

This section includes all functions that a customer can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_1 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for an unregistered user to search for flights. Search criteria consists of a date and a route. | | |
|
| Rationale: A guest user wants to search for flights. | | |
|
| Originator: Guest | | |
| Fit Criterion: Guest searches for flights | | |
|
|
| Dependencies: none | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_2 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for an unregistered user to register to the system. Registration consists of Personal details and login details. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. Login details consist of a username and password. | | |
|
| Rationale: A guest user wants to register to the system | | |
|
| Originator: Guest | | |
| Fit Criterion: Guest registers to the system | | |
|
| Dependencies: none | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_3 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a guest user to log into the system. Log in details consist of a username and password. | | |
|
| Rationale: A guest user wants to log into the system | | |
|
| Originator: Guest | | |
| Fit Criterion: Guest logs in to the system | | |
|
| Dependencies: The log in details of the user must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_4 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to search for flights. Search criteria consists of a date and a route. The system should display a list of available flights from the given start date to 7 days after the start date | | |
|
| Rationale: A customer wants to view available flights | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer views available flights from a given date to 7 days past the given date | | |
|
|
| Dependencies: none | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_5 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to book a flight with the system. A booking consists of a flight search, route, services, and seat selection. | | |
|
| Rationale: A customer wants to book a flight | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully books a flight. | | |
|
| Dependencies: The flight to be booked must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_6 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to view and select service options for a flight. A flight service consists of a one or more items, prices and quantity. | | |
|
| Rationale: A customer wants to choose services for a flight. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully chooses flight services | | |
|
| Dependencies: The service to be selected must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_7 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to view and select seating options for a flight. A seating choice consists of a seating class and price. | | |
|
| Rationale: A customer wants to choose a seat on a flight | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully chooses seating. | | |
|
| Dependencies: Seating to be chosen must be available | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_8 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to allow the system to automatically allocate seating for a flight. A seating choice consists of a seating class and price. | | |
|
| Rationale: A customer wants to the system to automatically allocate seating. | | |
|
| Originator: Customer | | |
| Fit Criterion: System allocates seating for customer. | | |
|
| Dependencies: The seat to be selected must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_9 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to allow the customer to add teir credit card details. Credit card details consists of a full name, card number, and expiry date. | | |
|
| Rationale: A customer wants to add credit card details to their profile. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully adds credit card details. | | |
|
| Dependencies: The customer profile to be edited must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_10 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to revise and modify booking details before confirming the booking. A booking consists of a flight search, route, services, seat selection and payment details. | | |
|
| Rationale: A customer wants revise or modify booking details before confirmation. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully revises and modifies booking details. | | |
|
| Dependencies: The customer profile to be edited must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_11 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface with the option for a customer to book a returning flight. A returning flight option will add extra flight selections, service selections and seating selections during booking. | | |
|
| Rationale: A customer wants to book a returning flight | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully books a returning flight. | | |
| Dependencies: A departing flight must already exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_12 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to view all bookings they have made. A booking consists of a flight, route, services, seating, price, and arrival/departure times. | | |
|
| Rationale: A customer wants to view all booking they have made. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully views previous bookings. | | |
|
| Dependencies: Customer bookings must already exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_13 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to set and modify their own personal details. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. | | |
|
| Rationale: A customer wants to set or modify personal details. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully sets or modifies personal details. | | |
|
| Dependencies: The customer profile to be edited must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_14 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to view their own personal details. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. | | |
|
| Rationale: A customer wants to view personal details. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully views personal details. | | |
| Dependencies: The customer profile to be viewed must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_15 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to cancel any bookings they have made. A cancelled flight incurs a cancellation fee to the customer. | | |
|
| Rationale: A customer wants to cancel a booked flight. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully cancels a booking. | | |
| Dependencies: the booking to be deleted must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_16 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to close their account. A close of account will remove their login privileges. Customer details such as no-fly status must still be kept. | | |
|
| Rationale: A customer wants to close their account. | | |
| Originator: Customer | | |
| Fit Criterion: Customer successfully closes their account. | | |
|
| Dependencies: The account to close must exist. | | |
|
| Priority: Essential. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_17 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a customer to book a flight for another person. All detail about the customer the booking is for must be entered. Customer details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. | | |
|
| Rationale: A customer wants to book a flight for another person | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully books a flight for another person. | | |
|
| Dependencies: none. | | |
|
| Priority: Essential. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_60 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should alert a booking manager if a 'watch' no fly status customer books a flight. | | |
|
| Rationale: The booking manager can take appropriate action after being alerted to the customers booking. | | |
|
| Originator: Customer | | |
| Fit Criterion: Booking manager is successfully alerted when 'watch' status customer books a flight. | | |
|
|
| Dependencies:Customer with 'watch 'status must exist, booking manager must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_66 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface that allows the customer to choose a customer age category when booking. Customers should need to specify either 'Adult' or 'Child' when navigating the booking procedure. | | |
|
| Rationale: Ticketing requires a specification of customer age. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer successfully specifies customer age while booking. | | |
|
| Dependencies:Customer who is booking must exist. | | |
|
| Priority: essential |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.2 Travel Agent Functions:**

This section includes all functions that a Travel Agent can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_18 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Travel agent to book a flight for a customer. A booking consists of a flight search, route, services, and seat selection. Customer personal details must also be obtained during booking, Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. | | |
|
| Rationale: A travel agent wants to book a flight for a customer. | | |
|
| Originator: Travel Agent | | |
| Fit Criterion: Travel agent successfully books a flight for a customer. | | |
|
| Dependencies: Flight to be booked must exist | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_19 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Travel agent to modify their profile details. Travel agent details consist of name, phone number and email. | | |
|
| Rationale: A travel agent wants to edit profile details. | | |
|
| Originator: Travel Agent | | |
| Fit Criterion: Travel agent successfully modifies profile details. | | |
|
| Dependencies: Profile to be modified must exist | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_20 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Travel agent to log into the system. Log in details consist of a username and password. | | |
|
| Rationale: A travel agent wants to log into system. | | |
|
| Originator: Travel Agent | | |
| Fit Criterion: Travel agent successfully logs into system | | |
|
| Dependencies: Travel agent login details must exist in system. | | |
|
| Priority: Critical. |  |  |
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_21 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Travel agent to view all bookings made from its profile. A booking consists of a Customer, flight, route, services, seating, price, and arrival/departure times. | | |
|
| Rationale: A travel agent wants view bookings made from its profile. | | |
|
| Originator: Travel Agent | | |
| Fit Criterion: Travel agent successfully views bookings | | |
| Dependencies: Bookings to be viewed must exist. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_22 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Travel agent to modify bookings made from its profile. A booking consists of a Customer, flight, route, services, seating, price, and arrival/departure times. | | |
|
| Rationale: A travel agent wants modify bookings made from its profile. | | |
|
| Originator: Travel Agent | | |
| Fit Criterion: Travel agent successfully modifies bookings | | |
| Dependencies: Bookings to be modified must exist. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_23 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Travel agent to modify customer’s personal details that have registered to the system through a travel agent. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. | | |
|
| Rationale: A travel agent wants modify a customers personal details. | | |
|
| Originator: Travel Agent | | |
| Fit Criterion: Travel agent successfully modifies customers personal details. | | |
|
| Dependencies: Customers profile to be modified must exist. | | |
| Priority: low |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.3 Staff Functions:**

This section includes all functions that a staff member can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_24 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to log into the system. Log in details consist of a username and password. Staff members also have an associated role. | | |
|
| Rationale: A staff member wants to log into the system. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully logs into the system. | | |
|
| Dependencies: Staff member log in details must exist in the system. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_25 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to create a booking for a customer. A booking consists of a Customer, flight, route, services, seating, price, and arrival/departure times. | | |
|
| Rationale: A staff member wants to create a booking for a customer. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully creates a booking for a customer. | | |
|
| Dependencies: Customer profile to be booked for must exist. Flight to be booked for must exist. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_26 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to create an account for a customer. A customer account personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. Log in details consist of username and password. | | |
|
| Rationale: A staff member wants to create an account for a customer. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully creates an account for a customer. | | |
|
| Dependencies: none. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_27 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to view a customer’s personal details and login details. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. Log in details consist of username and password. | | |
|
| Rationale: A staff member wants to view a customers personal details. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully views a customers personal details. | | |
|
| Dependencies: Customer to be viewed profile must exist. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_28 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to view a customers Booking details. A booking consists of a Customer, flight, route, services, seating, price, arrival/departure times and agency name. | | |
|
| Rationale: A staff member wants to view a customer’s booking details. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully views a customer’s booking details. | | |
| Dependencies: Customer to be viewed profile must exist. Customer to be viewed bookings must exist. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_29 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to modify a customer’s personal details. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. Log in details consist of username and password. | | |
|
| Rationale: A staff member wants to modify a customer’s personal details. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully modifies a customer’s personal details. | | |
|
| Dependencies: Customer to be modified profile must exist. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_30 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to modify a customer’s Booking details. A booking consists of a Customer, flight, route, services, seating, price, arrival/departure times and agency name. | | |
|
| Rationale: A staff member wants to modify a customer’s booking details. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully modifies a customer’s booking details. | | |
|
| Dependencies: Customer to be accessed profile must exist. Customer to be modified bookings must exist. | | |
|
| Priority: Critical. |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_31 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to search for a customer’s profile. Search criteria consist of a customer’s email address. A customers profile consists of personal details and booking details. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. booking details consists of a Customer, flight, route, services, seating, price, and arrival/departure times. | | |
|
| Rationale: A staff member wants to search for a customer’s profile. | | |
|
| Originator: Staff | | |
| Fit Criterion: Staff member successfully searches for a customer’s profile. | | |
|
| Dependencies: Customer to be searched profile must exist. | | |
|
| Priority: Critical. |  |  |
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_32 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a staff member to modify a customers seating details in a booking. Seating details consist of a class type and seat number. Seating can only be modified if destination seat is unoccupied. | | |
|
| Rationale: A staff member wants to modify a customer’s seating details in a booking. | | |
|
| Originator: Staff | | |
| Fit Criterion: Destination seat is unoccupied. Staff member successfully modifies seating details of a booking. | | |
|
|
| Dependencies:Booking to be modified must exist. Destination seat must be unoccupied. | | |
|
| Priority: Essential |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.4 Booking Manager Functions:**

This section includes all functions that a staff member can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_33 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a booking manager to modify a customer’s booking details. A booking consists of a Customer, flight, route, services, seating, price, and arrival/departure times. | | |
|
| Rationale: A Booking Manager wants to modify a customer’s booking details. | | |
|
| Originator: Booking Manager | | |
| Fit Criterion: Booking Manager successfully modifies a customer’s booking details. | | |
|
| Dependencies:Booking to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_34 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a booking manager to modify a customer’s seat allocation for a booking. The booking manager has the permissions to modify seating even if the destination seat is occupied. Seating consists of a flight, seat number and seat class. | | |
|
| Rationale: A Booking Manager wants to modify a customer’s seating details for a booking. | | |
|
| Originator: Booking Manager | | |
| Fit Criterion: Booking Manager successfully modifies a customer’s seating details for a booking. | | |
|
| Dependencies:Booking to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_35 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a booking manager to modify a customer’s no-fly status. A not fly status can be either null, 'watch' or 'no-fly'. | | |
|
| Rationale: A Booking Manager wants to modify a customers no fly status. | | |
|
| Originator: Booking Manager | | |
| Fit Criterion: Booking Manager successfully modifies a customers no fly status. | | |
|
| Dependencies:Customer to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_36 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a booking manager to access the report system to view reports relating to the reservation system. A report is a computation or compilation of informative data produced from the system. | | |
|
| Rationale: A Booking Manager wants to view reservation reports. | | |
|
| Originator: Booking Manager | | |
| Fit Criterion: Booking Manager successfully views reservation reports. | | |
|
| Dependencies:none | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.5 Service Manager Functions:**

This section includes all functions that a staff member can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_37 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Service Manager to modify service items details. A service item consists of an item, cost and availability. | | |
|
| Rationale: A Service Manager wants to modify service item details. | | |
|
| Originator: Service Manager | | |
| Fit Criterion: Service Manager successfully modifies a service item. | | |
|
| Dependencies:Service item to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_38 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Service Manager to add new service items. A service item consists of an item, cost and availability. | | |
|
| Rationale: A Service Manager wants to add a new service item. | | |
|
| Originator: Service Manager | | |
| Fit Criterion: Service Manager successfully adds a new service item. | | |
|
| Dependencies:none. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_39 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Service Manager to add new flight services to a flight. A flight service is a list of service items that are required for a flight. A flight service consists of a flight, service item and amount. | | |
|
| Rationale: A Service Manager wants to add a new flight service. | | |
|
| Originator: Service Manager | | |
| Fit Criterion: Service Manager successfully adds a new flight service. | | |
|
| Dependencies:Flight to be added to must exist. Item to be added must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_40 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Service Manager to remove a service item. A service item consists of an item, cost and availability. Removal of service item will cascade removal of references to this service item. | | |
|
| Rationale: A Service Manager wants to remove a service item. | | |
|
| Originator: Service Manager | | |
| Fit Criterion: Service Manager successfully removes a service item. | | |
|
| Dependencies:Service item to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_41 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Service Manager to remove a flight service from a flight. A flight service consists of a flight, item and amount. | | |
|
| Rationale: A Service Manager wants to remove a flight service. | | |
|
| Originator: Service Manager | | |
| Fit Criterion: Service Manager successfully removes a flight service. | | |
|
| Dependencies:Flight to be modified must exist. Flight service to be removed must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_42 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Service Manager to view reports from the report system that relate to the service system. A report is a computation or compilation of informative data produced from the system. | | |
|
| Rationale: A Service Manager wants to view service system reports | | |
|
| Originator: Service Manager | | |
| Fit Criterion: Service Manager successfully views service system reports. | | |
| Dependencies:none. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.6 Profile Manager Functions:**

This section includes all functions that a staff member can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_43 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Profile Manager to modify personal details of a customer. Personal details consist of title, first name, last name, gender, date of birth, phone number, email, address, state, country, credit card details, and passport details. | | |
|
| Rationale: A Profile manager wants to modify a customers profile. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Profile manager successfully modifies a customers profile. | | |
|
| Dependencies:Customers profile to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_44 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Profile Manager to modify a customer’s frequent flier points. Frequent flier points are an integer that accumulates when a customer books a flight. | | |
|
| Rationale: A Profile manager wants to modify a customers frequent flier points. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Profile manager successfully modifies a customers frequent flier points. | | |
|
| Dependencies:Customers profile to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_45 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Profile Manager to modify a travel agents profile. A travel agents profile consists of a name, phone number and email address. | | |
|
| Rationale: A Profile manager wants to modify a travel agents profile. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Profile manager successfully modifies a travel agents profile. | | |
|
| Dependencies:Travel agents profile to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_46 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Profile Manager to modify a staff members details. Staff details consist of a username and a password. | | |
|
| Rationale: A Profile manager wants to modify a staff member’s profile. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Profile manager successfully modifies a staff member’s profile. | | |
|
| Dependencies:Staff members profile to be modified must exist. | | |
|
| Priority: Essential |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_47 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Profile Manager to give discounts to customers. This discount could be due to sufficient frequent flier points, or due to purchasing large amount of flights. Discount consists of a percentage value. | | |
|
| Rationale: A Profile manager wants to discount a customer. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Profile manager successfully discounts a customer. | | |
|
| Dependencies:Customers profile to be modified must exist. | | |
|
| Priority: Essential |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_48 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Profile Manager to give discounts to Travel agents. This discount could be due to sufficient bookings. Discount consists of a percentage value. | | |
|
| Rationale: A Profile manager wants to discount a travel agent. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Profile manager successfully discounts a travel agent. | | |
|
| Dependencies:travel agents profile to be modified must exist. | | |
|
| Priority: Essential |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_49 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Profile Manager to view system reports relating to the profile system. A report is a computation or compilation of informative data produced from the system. | | |
|
| Rationale: A Profile manager wants to view profile reports. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Profile manager successfully views profile reports. | | |
|
| Dependencies:none | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_65 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should assign frequent flier points automatically to customers who book flights. Frequent flier points are determined as a percentage of the price of a flight. | | |
|
| Rationale: Profile manager wants frequent flier points to be assigned automatically. | | |
|
| Originator: Profile Manager | | |
| Fit Criterion: Frequent flier points are automatically assign when a customer books a flight. | | |
|
| Dependencies:none. | | |
|
| Priority: low |  |  |
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.5 Flight Manager Functions:**

This section includes all functions that a staff member can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_50 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to add a new aircraft to the system. An aircraft consists of a name, amount in service, first class seats, business class seats, premium economy seats, economy seats and total seats. | | |
|
| Rationale: A Flight manager wants to add an aircraft to the system. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully adds a new aircraft to the system. | | |
|
| Dependencies:none | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_51 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to add a new schedule to the system. A schedule consists of a flight ID, plane, route, depart time and arrival time. | | |
|
| Rationale: A Flight manager wants to add a schedule to the system. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully adds a new schedule to the system. | | |
| Dependencies: Flight ID to be added must exist, plane to add must exist, route to add must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_52 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to add a new airport to the system. An airport consists of an id, name, city, country, IATA, Latitude, longitude, altitude, timezone, DST and tz. | | |
|
| Rationale: A Flight manager wants to add an airport to the system. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully adds a new airport to the system. | | |
|
| Dependencies:none | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_53 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to add a new route to the system. A route consists of an ID, source airport, destination airport, codeshare and stops. | | |
|
| Rationale: A Flight manager wants to add a route to the system. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully adds a new route to the system. | | |
|
| Dependencies:Source airport must exist, destination airport must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_54 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to modify or remove an aircraft. An aircraft consists of a name, amount in service, first class seats, business class seats, premium economy seats, economy seats and total seats. Removal of this aircraft will cascade removal to all references in the system. | | |
|
| Rationale: A Flight manager wants to modify an aircraft. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully modifies an aircraft. | | |
|
| Dependencies: aircraft to be modified/removed must exist . | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_55 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to modify a schedule. A schedule consists of a flight ID, plane, route, depart time and arrival time. Removal of this schedule will cascade removal to references in the system. | | |
|
| Rationale: A Flight manager wants to modify/remove a schedule | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully modify/remove a schedule. | | |
|
| Dependencies: Schedule to be modified must exist. | | |
|
| Priority: Critical |  |  |
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_56 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to modify or remove an airport. An airport consists of an id, name, city, country, IATA, Latitude, longitude ,altitude, timezone, DST and tz. Removal of this airport will cascade removal to all references in the system. | | |
|
| Rationale: A Flight manager wants to modify/remove an airport. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully modify/remove an airport. | | |
| Dependencies:airport to modify exists | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_57 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to modify or remove a route. A route consists of an ID, source airport, destination airport, codeshare and stops. Removal will cascade removal to all references in the system. | | |
|
| Rationale: A Flight manager wants to modify/remove a route | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully modified/removed a route. | | |
|
| Dependencies:Route to be modified must exist. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_58 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based user interface for a Flight manager to view system reports that relate to the flight system. A report is a computation or compilation of informative data produced from the system. | | |
|
| Rationale: A Flight manager wants to view flight reports. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Flight manager successfully views flight reports. | | |
|
| Dependencies:none. | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_64 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should alert all customers that are booked on flights that have been cancelled. The alert should appear the first time a customer logs in after a flight has been cancelled. | | |
|
| Rationale: Customers need to know that their flight is no longer valid. | | |
|
| Originator: Flight Manager | | |
| Fit Criterion: Customers are alerted when they are booked on a cancelled flight. | | |
|
| Dependencies:none. | | |
|
| Priority: essential |  |  |
|
| Supporting Materials: none | | **Volere** |
| **Copyright © Atlantic Systems Guild** |

**3.1.6 Administrator Functions:**

This section includes all functions that an administrator can use to perform required tasks

|  |  |  |
| --- | --- | --- |
| Requirement# F\_67 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based console user interface to allow an administrator to add new staff members to the system. | | |
|
| Rationale: Admin wants to grant an access to a staff member. | | |
|
| Originator: Administrator | | |
| Fit Criterion: Administrator adds new staff member. | | |
|
| Dependencies:Staff member to be granted access must exist | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_68 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based console user interface to allow an administrator to access all staff member functionality | | |
|
| Rationale: Admin wants to access any functionality from staff members | | |
|
| Originator: Administrator | | |
| Fit Criterion: Administrator access staff functionality | | |
|
| Dependencies:none | | |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.2 System Requirements**

This section includes all functions that a staff member can use to perform required tasks.

|  |  |  |
| --- | --- | --- |
| Requirement# F\_61 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text based interface for a customer to reschedule cancelled flights. A list of closest upcoming flights on the cancelled flights route shold be displayed to the customer. | | |
|
| Rationale: A customer wants to reschedule a cancelled flight. | | |
|
| Originator: Customer | | |
| Fit Criterion: Customer is books a flight to replace a cancelled flight. | | |
|
|
| Dependencies:Customer with previous cancelled flight must exist. Upcoming flight on correct route must exist. | | |
|
| Priority: optional |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_62 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a GUI for all user interactions. | | |
|
| Rationale: A user wants to use a GUI to operate the system. | | |
| Originator: User | | |
| Fit Criterion: Customer uses a GUI to operate the system. | | |
|
| Dependencies:none. | | |
|
| Priority: optional |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_63 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should allow multiple users to access the system simultaneously. | | |
|
| Rationale: Many users want to access the system at the same time. | | |
|
| Originator: Un Registered User | | |
| Fit Criterion: Many customers access the system concurrently. | | |
|
| Dependencies:none. | | |
|
| Priority: optional |  |  |
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.3 Non Functional Requirements**

This section includes all functions that are defined as non-functional requirements.

|  |  |  |
| --- | --- | --- |
| Requirement# NF\_1 | Requirement Type: Non Functional | Event/Use Case: N/A |
| Description: The system should operate on both Windows and Linux systems. | | |
| Rationale: Users may use system on either linux or windows architectures. | | |
|
| Originator: none | | |
| Fit Criterion: System operates on either windows or linux systems. | | |
|
| Dependencies:none. | | |
|
| Priority: essential |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# NF\_2 | Requirement Type: Non Func | Event/Use Case: N/A |
| Description: A customers login password must be displayed as astrixes during login. | | |
|
| Rationale: Displaying astrixes instead of text increases system security | | |
|
| Originator: Customer | | |
| Fit Criterion: Customers login password is displayed with astrixes. | | |
|
| Dependencies:none. | | |
|
| Priority: Critical |  |  |
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |