CSCI222 – Group B4 Assignment 1

System Requirements Specification

1. - Introduction

1.1 - Purpose

The purpose of this document is to specify the functional and non-functional requirements of an country club facilities booking 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 **‘Country Club Facilities Booking System’.**

Two categories of users will utilise this system, namely:

* Staff members
* Club members (Customers)

Staff members will be divided into five categories:

* Staff
* Club Manager
* Administrator

**Staff**

* A generic category that will handle customer-related tasks.

**Club Manager**, a staff member tasked to:

* Give different fees, priorities, access rights to club members when booking facilities based on their membership rankings.
* Give discounts entitlement pertaining to club’s services to said club members.
* Create, delete, amend and update club’s facilities for booking
* Set their various charging rates based on peak and off-peak periods etc.
* View rental behaviour, facilities and management-related reports.

**Administrator**, 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, CCFBS is divided into subsystems:

**Club Membership Rankings Subsystem**

* Gives club members different membership rankings based on the amount they paid
* Gives different special privileges to club members based on their membership rankings

**Facilities Management Subsystem**

* Allows club manager to create, delete, amend and update club’s facilities for bookings
* Set various charging rates based on peak and off-peak periods, etc.

**Search Portal Subsystem**

* Allows club members to view available online facilities for booking or club’s services to request.
* Advanced search features allows members to search by criterias like facilities, charging rates, availability or any other criteria available in the database.

**Members Profile Subsystem**

* Manage club members (create, edit, delete profiles)
* Update members’ membership rankings, privileges
* Storing a club member’s booking preferences
* Gives notifications to club members whenever their booking starts or ends, etc.

**Reporting Subsystem**

* Generate various summary reports such as,
* Usages of the different facilities
* Rental behaviour of club members
* Discount rates given, etc

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 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 VMWare Ubuntu-based machine within the company before operation.

2.2.1 User interfaces

**The user interface to be provided to guests**

* Must be a text based console interface.
* Provide different options that other user interfaces.
* Have the lowest system access rights in the system.

**The user interface to be provided to club members**

* Must be a text based console interface
* Provides more privileges compared to guests, such as booking and searching of facilities.

**The user interface provided to staff members**

* Must be a text based console interface.
* Provides many of the same options provided to the club member
* Some additional options must be provided for the staff member to effectively perform their jobs (e.g. edit profiles, bookings, etc).

**The user interfaces provided to management level staff members**

* Must be a text based console interface.
* Give the user options specific to the tasks each manager needs to perform
* Primarily allows the manager access to their respective sub-system.

**The user interface provided to the Administration staff**

* Must be a text based console interface.
* Provide full access to all staff and management functionalities,
* As well some extra privileges such as the ability to grant access to staff members

2.2.2 Hardware interfaces

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

2.2.3 Software Interfaces

The system is self-contained. No external software is required for operation. C++ is used to code the system.

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 CCFBS 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 CCFBS is to manage the club and allow club members to search, view and/or book facilities and provide a means for club members to manage (includes creation, editing and deletion), utilise and use the information.

**A club member should be able to use these primary functions:**

* Create an account and/or login with that account
* Obtain more membership privileges based on the amount he/she pays for it
* Book facilities for him/herself and/or others. A booking consists of information on the facility booked, booking fees, booking time and duration and booking queue list
* View available online facilities for booking or club’s services to request
* Search facilities by special criterias such as charging rates, availability, etc
* Manage his/her own member profile
* Manage his/her own payment details

**A Club Manager should be able to use these primary functions:**

* View a club member’s personal details
* View a club member’s payment details
* View information detailing a club facility
* Give membership rankings to club members
* Give special privileges to club members based on their membership rankings
* Modify a club member’s personal details
* Modify a club member’s booking information
* Search for a club member

**An administrator should be able to use these primary functions:**

* Access all staff functionalities
* Access all management level functionalities
* Grant access rights to staff members

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

2.4 User Characteristics

* Staff members have good knowledge of the tasks and processes of club member’s activities
* Club managers have a high level of knowledge in the different club facilities, different rankings of club membership, along with each and every privileges embedded within each rankings
* Administrators have good knowledge of all tasks and processes of all system users and sub systems. Also has knowledge of system installations

2.5 Constraints

The system should obey the following constraints:

* User Authentication: For a user to book a club facility 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 CCFBS 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 Feb 2016.

2.6 Assumptions and Dependencies

The following assumptions and dependencies are defined:

* All club members should have an email address
* All staff members should have an email address
* All users of the system are proficient in English language

3.0 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 all critical and essential functions are implemented.
* Optional – lowest importance priority. It would be desirable but not necessary to implement.

3.1 User Management Functions

This section defines functionalities provided for club members, staff members, Club managers, club facilities manager, search system manager, profile manager, administrators.

**3.1.1 Club member Functions:**

This section includes all functions that a club member 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 club facilities. No search criteria exists due to guest’s limitations. | | |
|
| Rationale: A guest user wants to search for facilities available for booking. | | |
|
| Originator: Guest | | |
| Fit Criterion: Guest searches for facilities available for booking. | | |
|
|
| 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 into the system. | | |
|
| Originator: Guest | | |
| Fit Criterion: Guest registers into the system. | | |
|
|
| Dependencies: The account to be registered must not exist before registration. | |  |
|
| 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 into the system. | | |
|
|
| Dependencies: The account to be logged in 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 club member to search for club facilities. Search criterias include facilities, charging rates, availability, etc. | | |
|
| Rationale: A club member wants to view available online facilities for booking or club’s services to request. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member searches for available online facilities or club’s services to request. | | |
|
|
| 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 club member to book a facility. Options include which facility booked, time and duration of booking. | | |
|
| Rationale: A club member wants to book a club facility. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member books a club facility. | | |
|
|
| Dependencies: Must book a valid and available club facility. | |  |
|
| 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 club member to view the booking(s) they have made with the system. The information provided includes the name of facility book, time and duration of the booking. | | |
|
| Rationale: A club member wants to view a booking he/she has made. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member views a booking he/she has made before. | | |
|
|
| Dependencies: The booking to view 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 club member to revise the details of his/her booking before confirming it. | | |
|
| Rationale: A club member wants to revise booking details before confirming. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member revises booking details before confirming. | | |
|
|
| Dependencies: The booking to be revised must exist | |  |
|
| 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 club member to cancel his/her own booking. | | |
|
| Rationale: A club member wants to cancel his/her own booking. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member cancels his/her own booking. | | |
|
|
| Dependencies: The booking to be cancelled 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 club member to book a club facility for another club member. Details to input include the account details for the aforementioned club member in addition to the booking options. | | |
|
| Rationale: A club member wants to book a club facility for another club member. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member books a club facility for another club member. | | |
|
|
| Dependencies: The other club member account must exist, facility must exist and available for booking | |  |
|
| 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 interface for a club member to view his/her own personal details. Details include the data the club member inputted during his registration, his membership ranking, and his credit card number (if already registered) | | |
|
| Rationale: A club member wants to view his/her own personal details. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member views his/her own personal details. | | |
|
|
| Dependencies: Account 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 interface for a club member to set and/or modify his/her own personal details. Details include editing the data the club member inputted during his registration and his credit card number (if already registered) | | |
|
| Rationale: A club member wants to set and/or modify his/her own personal details. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member sets and/or modify his/her own personal details. | | |
|
|
| Dependencies: Account must 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 interface for a club member to add in his/her own credit card info. | | |
|
| Rationale: A club member wants to add in his/her own credit card info. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member adds in his/her own credit card info. | | |
|
|
| Dependencies: Card number is valid and in 16 digits. | |  |
|
| 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 club member logged in to log out of the system. | | |
|
| Rationale: A club member wants to log out of the system. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member logs out of the system. | | |
|
|
| Dependencies: Account must currently be in a logged-in state. | |  |
|
| 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 alert a club member if he/she is in a ‘no booking’ status to notify the club member that he/she cannot book a facility currently. | | |
|
| Rationale: Club member in a ‘no booking’ status should not be able to book facilities. | | |
|
| Originator: Club member | | |
| Fit Criterion: Club member in a ‘no booking’ status is notified that he/she currently cannot book in any club facilities. | | |
|
|
| Dependencies: Club member currently in a ‘no booking’ status. | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.2 Club manager functions**

|  |  |  |
| --- | --- | --- |
| Requirement# F\_15 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text-based user interface for a Club manager to give membership rankings and privileges towards club members based on the amount they spent. | | |
|
| Rationale: A Club manager wants to give membership rankings and privileges towards club members based on the amount they spent. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager gives membership rankings and privileges to club members. | | |
|
|
| Dependencies: Club member profile to be granted membership ranking 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 Club manager to edit his/her own personal details. Details editable are similar to what a club member can edit. | | |
|
| Rationale: A Club manager wants to edit his/her own personal details. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager edits his/her own personal details. | | |
|
|
| Dependencies: Club manager account must exist. | |  |
|
| Priority: Critical |  |  |
|
| 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 Club manager to view a club member’s bookings. | | |
|
| Rationale: A Club manager wants to view a club member’s bookings. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager views a club member’s bookings. | | |
|
|
| Dependencies: Club member account must exist. | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_18 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text-based user interface for a Club manager to modify a club member’s booking information. Modifications include discounting fees and giving priorities for said club members. | | |
|
| Rationale: A Club manager wants to modify a club member’s booking information. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager modifies a club member’s booking information. | | |
|
|
| Dependencies: Club member’s account 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 Club manager to give club members with certain membership rankings a discount entitlement pertaining to some of the club’s facilities (dining in club restaurant, karaoke, billiard, etc) | | |
|
| Rationale: A Club manager wants to give discounts entitlement pertaining to some of the club’s facilities to club members with certain membership rankings. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager gives discount entitlement to some club members. | | |
|
|
| Dependencies: Club member account must exist, membership ranking must be sufficient. | |  |
|
| 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 Club manager to give club members with certain membership rankings a discount entitlement pertaining to some of the club’s facilities (dining in club restaurant, karaoke, billiard, etc) | | |
|
| Rationale: A Club manager wants to give discounts entitlement pertaining to some of the club’s facilities to club members with certain membership rankings. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager gives discount entitlement to some club members. | | |
|
|
| Dependencies: Club member account must exist, membership ranking must be sufficient. | |  |
|
| 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 club manager to view rental behaviour, facilities and management related reports. | | |
|
| Rationale: A club manager wants to view rental behaviour, facilities and management-related reports. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager views rental behaviour, facilities and management-related reports. | | |
|
|
| Dependencies: The reports 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 Club manager to view a club member’s personal details. | | |
|
| Rationale: A Club manager wants to view a club member’s personal details. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager views a club member’s personal details. | | |
|
|
| Dependencies: Club member account 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 Club manager to view information related to club facilities. Information include the number of bookings, time and duration of bookings, etc. | | |
|
| Rationale: A Club manager wants to view information related to a club facility. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager views information related to a club facility. | | |
|
|
| Dependencies: Search system must be functional. | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

|  |  |  |
| --- | --- | --- |
| Requirement# F\_24 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text-based user interface for a Club manager to create, delete, amend or update a club facility for booking. | | |
|
| Rationale: A Club manager wants create, delete, amend or update a club facility for booking. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager create, deletes, amends or updates a club facility for booking. | | |
|
|
| Dependencies: If creating, club facility must not exist before. In every other option, club facilities to delete, amend or update must exist. | |  |
|
| 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 Club manager to set various charging rates based on peak and off-peak periods. | | |
|
| Rationale: A Club manager wants to set various charging rates based on peak and off-peak periods. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager sets various charging rates based on peak and off-peak periods. | | |
|
|
| Dependencies: None | |  |
|
| 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 Club manager to notify a club member whenever their booking starts or ends. | | |
|
| Rationale: A Club manager wants to notify a club member whenever their booking starts or ends. | | |
|
| Originator: Club manager | | |
| Fit Criterion: Club manager notifies a club member whenever their booking starts or ends. | | |
|
|
| Dependencies: Club member must exist and is booking a club facility. | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.3 Admin functions**

|  |  |  |
| --- | --- | --- |
| Requirement# F\_27 | Requirement Type: Functional | Event/Use Case: N/A |
| Description: The system should provide a text-based user interface for an admin to add a club member or club manager. | | |
|
| Rationale: An admin wants to add a club member or club manager. | | |
|
| Originator: Admin | | |
| Fit Criterion: Admin adds a club member or manager. | | |
|
|
| Dependencies: Club member or manager account to add must not exist beforehand. | |  |
|
| 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 allow an admin to access all other functionalities of the system. | | |
|
| Rationale: An admin wants to access all other functionalities of the system. | | |
|
| Originator: Admin | | |
| Fit Criterion: Admin accesses all other functionalities of the system. | | |
|
|
| Dependencies: None. | |  |
|
| 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 an admin to view rental behaviour, facilities and management related reports. | | |
|
| Rationale: An admin wants to view rental behaviour, facilities and management-related reports. | | |
|
| Originator: Admin | | |
| Fit Criterion: Admin views rental behaviour, facilities and management-related reports. | | |
|
|
| Dependencies: The reports must exist | |  |
|
| Priority: Critical |  |  |
|
| Supporting Materials: none | | **Volere** |
|
|
| **Copyright © Atlantic Systems Guild** |

**3.1.4 Non-functional requirements**

|  |  |  |
| --- | --- | --- |
| Requirement# NF\_1 | Requirement Type: Non  Functional | Event/Use Case: N/A |
| Description: The system should operate on both Windows and VMWare Ubuntu systems. | | |
| Rationale: Users may use system on either Ubuntu or windows architectures. | | |
|
| Originator: none | | |
| Fit Criterion: System operates on either windows or Ubuntu 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 customer’s 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** |