Airport Management System

Software Requirements Document

V 1.0

Table of Contents

Introduction

1.1 About this document

1.1.1 Purpose & Scope of the document

The purpose of the software requirements document is to systematically capture requirements for the project and the system **Airport Management System(AMS)** application. AMS is a system used for adding the plane details along with pilot details and adding the hanger and adding the allotting hangers to the planes and displaying the details of the hangers scheduling.

The scope of this document is limited to addressing the requirements from a user, quality, and non-functional perspective. It is recommended that design aspects are not added in this document

1.1.2 Intended Audience

Project Team

1.2 About the Software System

The client would like to develop an independent application **Airport Management System(AMS)** . The core essence for the system is to adding the plane information along with pilot details, adding the hanger and adding the allotting hangers to the planes, scheduling of the hangers availability.

The following section will cover aspects related to Property Insurance Management System.

1.2.1 Purpose

The Airport Management System (AMS) core essence for the system is to adding the plane information along with pilot details, adding the hanger and adding the allotting hangers to the planes, scheduling of the hangers availability.

The following are the important modules in the system

a) Adding Planes

b) Adding Pilots

c) Adding Hangers Details

d) Allotting the planes into hangers.

e) Get Hanger Status

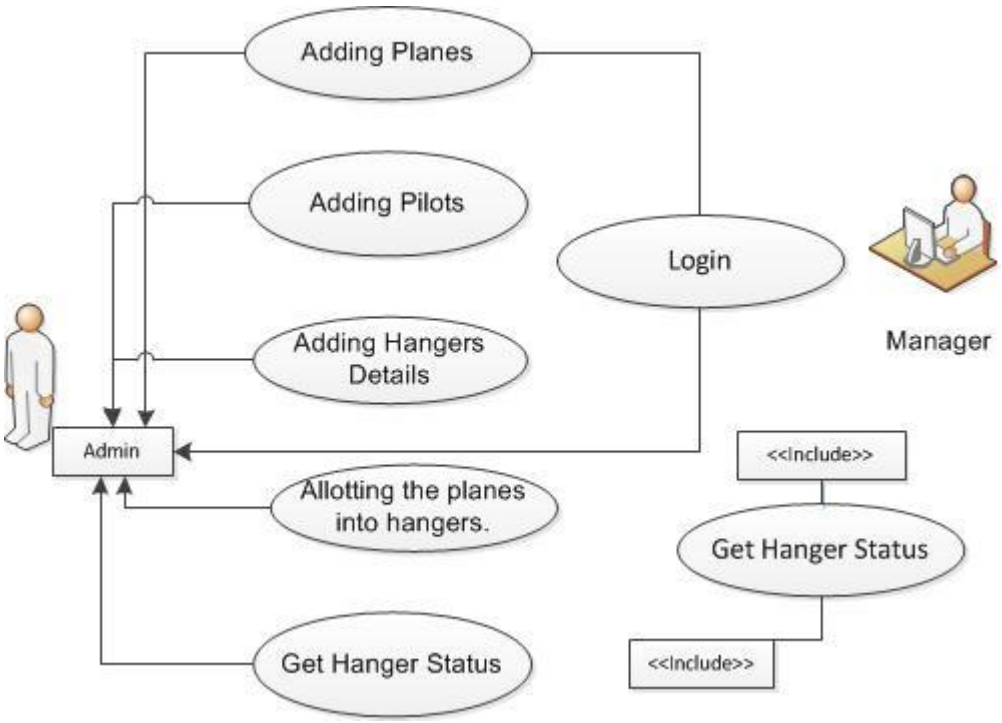
1.2.2 Scope of the system

The scope of the system is explained through its modules as follows

- Adding Planes - by administrator to add the plane details along with the owner details.
- Adding Pilots - administrator to add the pilot details along with the address details. The pilots are having one license number, social security number. Many pilots can stay in a single address.
- Adding Hangers Details - by administrator to add the hanger details along with the manager and manager address details. Each Hanger has a respective manager.
- Allotting the planes into hangers. - by manager search for the empty hangers and allocate the plane into hanger based on the availability of hangers
- Get Hanger Status - manager search for the status of the hanger for the given dates.

Use Case Diagram

Airport Management System



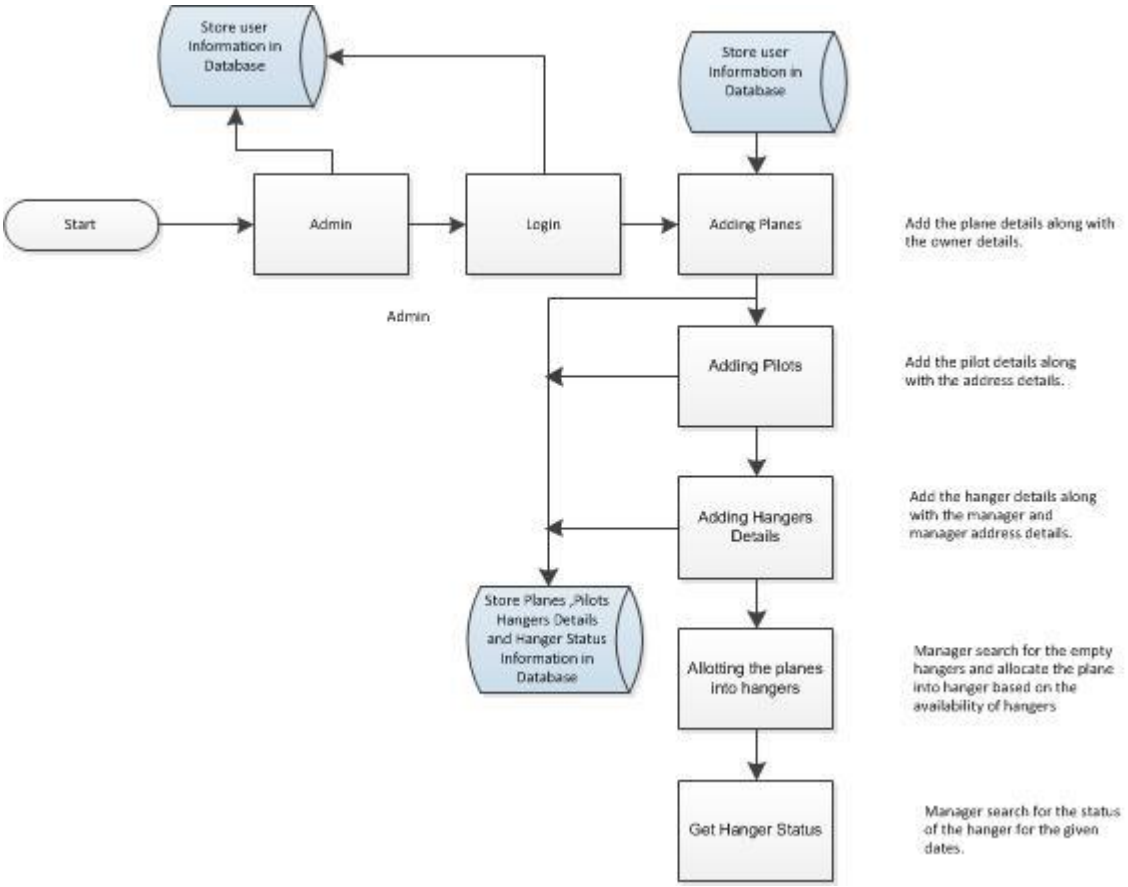
1.2.3 Exclusions

The system will operate only on the modules discussed above and will not include any additional functionality.

1.2.4 System Perspective

Airport Management System is an independent software system developed to manage the activities like AMS Adding Planes, Adding Pilots, Adding Hangers Details, Allotting the planes into hangers and to get the hanger Status by using the Java/ Dotnet architecture.

1.2.5 System diagram



1.2.6 Architecture diagram

Physical Architecture:

A physical architecture is an arrangement of physical elements, (system elements and physical interfaces) that provides the designed solution for a product, service, or enterprise. It is intended to satisfy logical architecture elements and system requirements. Auto Identification Process follows a three layered architecture namely presentation layer, business logic layer and data access layer.

- **Presentation Tier** is the tier in which the users interact with an application. Presentation Tier contents Shared UI code, Code Behind and Designers used to represent information to user.

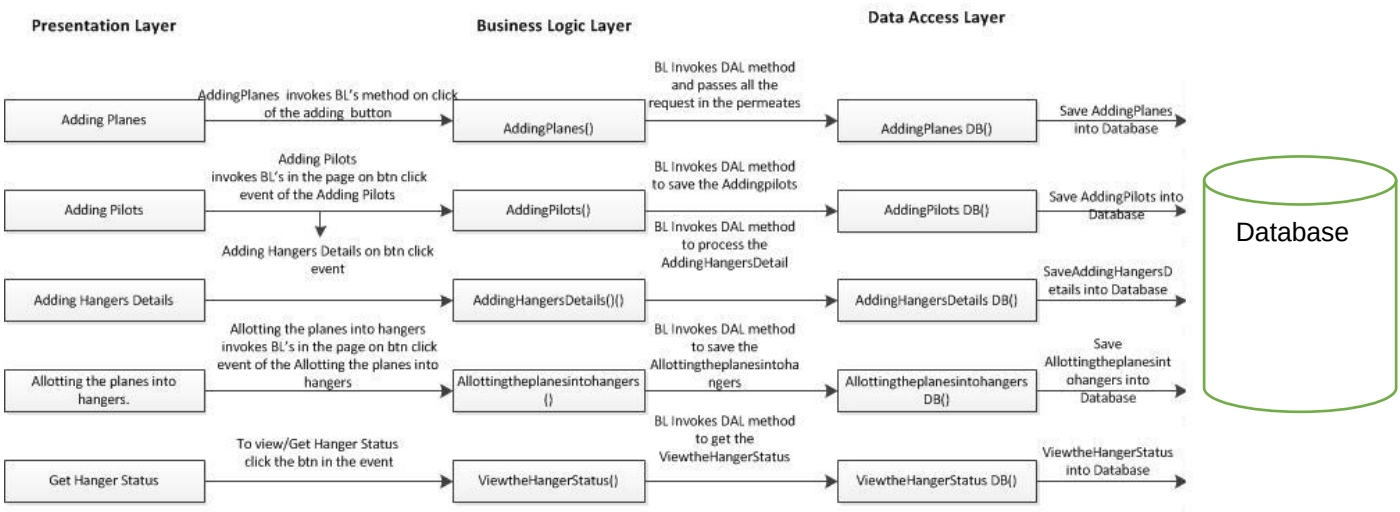
- **Business Tier** is mainly working as the bridge between Data Tier and Presentation Tier. All the Data passes through the Business Tier before passing to the presentation Tier. Business Tier is the sum of Business Logic Layer, Data Access Layer and Value Object and other components used to add business logic.

- **Data Tier** is basically the server which stores all the application's data. Data tier contents Database Tables, XML Files and other means of storing Application Data.

Logical Architecture:

The Logical Architecture defines the Processes (the activities and functions) that are required to provide the required User Services. Many different Processes must work together and share information to provide a User Service. The Processes can be implemented via software, hardware, or firmware. The Logical Architecture is independent of technologies and implementations

3-Layered Architecture Logical Diagram



1.2.7 System Environment

The Airport Management System application will be operated from the client server with parallel processor support. When a user connects to the Web Server, the Web Server will interact with the Database after processing the business logic to transfer data to and from a database.

- Java/J2EE or .NET Framework application deployment.
- My Sql/Sql server database for data storage

1.2.8 User Characteristics

The application should be user friendly; hence the user should not need any software and hardware knowledge. The user may be new to the client products having never used them before.

The types of users likely to use the system are as follows,

- Client
- Delivery Assurance Group, Process Engineering group
- Developers, testers, other associates in the project

1.2.9 Impact of the System

This is a new product which is developed for internal users. Expected impact of the product is to automate existing manual processes in order to make them more efficient and cost effective.

1.2.10 Assumptions, Dependencies, Risks / Constraints

Assumptions:

The AMS Admin must possess transaction data along with the Adding Planes, Adding Pilots, Adding Hangers, Allotting the planes into hangers and the status for the hangers' availability.

Dependencies:

If there is any similar Identification process required for other applications of the same client, this system should be able to accommodate such a reusability dependency.

Risks:

If there is any failure in identification processes, the system should be able to handle such a financial risk.

1.2.11 Design Constraints

Application should have single login feature. Login functionality should be the welcome feature for the application.

Only administrator can access the application.

None of the application features can be accessed without login.

Frameworks

- **SpringBoot Rest API for connecting to database**

- **Client App using Spring Framework/ Spring Boot**
- **Use RabbitMQ for sending the messages**

System Requirements

1.3 Functional Requirements

1.3.1 Airport Management System - Adding Plane

Airport Management System **Req-2.1.1**	**Adding Plane**
When the users enter the following link **http://localhost:8080/ AirportManagementsystem** in the browser, Home page appears which will have a short welcome message; a brief introduction to the purpose of the Airport Management System, the user can choose appropriate options. Under home page, we need below interface and Home page Interface Requirements. Adding Planes, Adding Pilots, Adding Hangers, Allotting the planes into hangers and Get Hanger Status form should have basic attributes/fields as mentioned in UI Interface Screen section. At any given point of time, the user can Deselect/Reset all or can go to home page by the clicking home.	
UI Interface Screen	Manufacturer Name - Name of the plane manufacturer Registration No - Registration Number of the plane Model No - Model No of the plane Plane Name - Name of the plane Capacity - capacity of the plane Email - Email address of the owner, IsOwnerExist and If the above status is not available the dynamically populate the following the fields House No - House no of the owner Address Line1 - Locality of the owner. City - City of the Owner State - State of the Owner Country - Country of the Owner Pin No - Pin No of the Owner

	Add Plane/ Reset - Used to register the plane details into the table Button used to reset/clear the field values
Trigger	Administrator triggers the functionality once he/she enter into the system
Pre-Conditions	The administrator should have the details of the plane to be added into the system.
Post Conditions	The system should generate the plane id, address id, owner id
Success End Condition	On success a message should be displayed **"Plane added successfully <generated plane id>"**
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. Administrator logs in to the system. 2. Administrator clicks on add pilot option. 3. System should display all the necessary fields in the screen. 4. Get all the necessary information. 5. Write a java Script code for date picker for date fields. 6. Enter all the Necessary information in the given form. 7. Enter the address details and click on add pilot button. 8. Check email id, mobile no, social security NO, licence no already existing or not. 9. If any business exception occurs corrects the details and submit.
Business Rules & Validations	• The pilot name should contain only alphabets and space. • The licence number should be 10 digit numeric fields which should not be same for any two pilots. • The city should contain only alphabets and space.

	• The state should contain only alphabets and space. • The country should contain only alphabets and space. • The pin No should contain only numbers. • The pin No length should contain exactly 7. • The email address should contain the alpha numeric and special characters only one @ symbol should be available. • If the email id already exists then raise the business exception. • If the mobile no already exist then raise the business exception. • Check if the house no and city is already existing or not if not then generate the address id and return it. • If existing get the address id and insert into the system. • The Social security number should not be same for any other pilot and managers. • The entered licence number should not be same for other pilot. • Finally insert the pilot details into the system. • The Pilot id is generated as • Pilot ID= Last 4 characters from Social Security No + unique Number (starts with 31). The Address id is generated as Address ID= starting 3 characters from city + unique Number (starts with 101). Insert the above details into the database.

1.3.2 Airport Management System - Adding Pilot

Airport Management System **Req-2.1.2**	**Adding Pilot**
By administrator should be able to add the pilot details along with the address details. The pilots are having one license number, social security number. Many pilots can stay in a single address	

UI Interface Screen	Pilot Name - Name of the pilot License No - License Number of the pilot Social Security No - Social security number of the pilot Date of Birth - date of birth of the pilot.(A Java Script) date picker Gender - gender of the pilot Mobile No - Mobile No of the pilot Email Address - email address of the pilot House No - House no of the pilot Address Line1 - locality of the pilot City - City of the pilot State - State of the pilot Country - Country of the pilot Pin No - Pin No of the pilot Add Pilot/ Reset - Submit Button used to register the plane details into the table Button used to reset/clear the field values
Trigger	Administrator triggers the functionality once he/she enter into the system
Pre-Conditions	The administrator should have the details of the pilots to be added into the system
Post Conditions	The system should generate the pilot id, address id.
Success End Condition	On success a message should be displayed **"Pilot added successfully <generated pilot id>"**
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. Administrator logs in to the system. 2. Administrator clicks on add pilot option. 3. System should display all the necessary fields in the

	screen. 4. Get all the necessary information. 5. Write a java Script code for date picker for date fields. 6. Enter all the Necessary information in the given form. 7. Enter the address details and click on add pilot button. 8. Check email id, mobile no, social security NO, licence no already existing or not. 9. If any business exception occurs corrects the details and submit.
Business Rules & Validations	• The pilot name should contain only alphabets and space. • The licence number should be 10 digit numeric fields which should not be same for any two pilots. • The city should contain only alphabets and space. • The state should contain only alphabets and space. • The country should contain only alphabets and space. • The pin No should contain only numbers. • The pin No length should contain exactly 7. • The email address should contain the alpha numeric and special characters only one @ symbol should be available. • If the email id already exists then raise the business exception. • If the mobile no already exist then raise the business exception. • Check if the house no and city is already existing or not if not then generate the address id and return it. • If existing get the address id and insert into the system. • The Social security number should not be same for any other pilot and managers. • The entered licence number should not be same for other pilot. • Finally insert the pilot details into the system. • The Pilot id is generated as • Pilot ID= Last 4 characters from Social Security No + unique Number (starts with 31). The Address id is generated as Address ID= starting 3 characters from city + unique Number (starts with 101). Insert the above details into the database.

1.3.3 Airport Management System - Adding Hanger Details

1.3.3 Airport Management System **Req-2.1.3**	**Adding Hanger Details**
Adding Hanger Details module. The details for the other modules will be given in the respective modules functional specification. By administrator to add the hanger details along with the manager and manager address details. Each Hanger has a respective manager.	
UI Interface Screen	• Hanger Location - location of the hanger in airport • Hanger capacity - No of planes can hold inside the hanger • Manager Name - Name of the manager • Social Security No - Social security number of the manager. • Date of Birth - Date of birth of the manager. (A Java Script date picker). • Gender - gender of the manager • Mobile NO - Mobile No of the manager • Email Address - email address of the manager. • House No - House no of the manager • "Address Line1 - Address Line1-locality of the manager." • City - City of the manager • State - State of the manager • Country - Country of the manager • Pin No - Pin No of the manager • Add Hanger - Submit Button used to register the plane details into the table • Reset - Button used to reset/clear the field values
Trigger	Administrator triggers the functionality once he/she enter into the system
Pre-Conditions / Assumptions	The administrator should have the details of the hanger and the manager to be added into the system.

Post Conditions	The system should generate the hanger number, manager id, address id.
Success End Condition	On success a message should be displayed "**Hanger added successfully <generated Hanger id>"**
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. Administrator logs in to the system. 2. Administrator clicks the Add Hanger option. 3. System should display all the necessary fields in the screen. 4. Get all the necessary information. 5. Enter all the Necessary information in the given form. 6. Enter the address details and click on add hanger button. 7. Check for the email id, mobile no and social security no already existing in the system or not. 8. If any business exception occurs corrects the details and submit.
Business Rules & Validations	• The hanger location contains only alphabets and space. • The hanger capacity contains only numbers. • The Date of Birth should be in the format of dd-mm-yyyy. • Mobile no should contain only numbers and length should exactly 10 digits. • The city should contain only alphabets and space. • The state should contain only alphabets and space. • The country should contain only alphabets and space. • The pin No should contain only numbers. • The pin No length should contain exactly 7. • The email address should contain the alpha numeric and special characters only one @ symbol should be available. • If the email id already exists then raise the business exception. • If the mobile no already exist then raise the business exception. • Check if the house no and city is already existing or not if not then generate the address id and return it. • If existing get the address id and insert into the system.

	• Check the social security number already exists for any pilot and manager. If existing throw the business exception. • Finally insert the hanger details into the system. • The Hanger No is generated as • Hanger No= starting 3 character from hanger location + unique Number (starts with 101). The Address id is generated as Address ID= starting 3 characters from city + unique Number (starts with 101). The manager id is generated as Manager ID= Last 4 characters from Social Security No+ unique Number (starts with 31).

1.3.4 Airport Management System - Allotting the Planes into Hanger

1.3.4 Airport Management System Req-2.1.4	**Allotting the Planes into Hanger**
The manager search for the empty hangers and allocate the plane into hanger based on the availability of hangers Hanger: It is a closed structure to hold aircraft or spacecraft in a protective storage.	
UI Interface Screen	**Hanger Search Screen:** From date - From date of booked To date - To date booked **Available Hanger Detail Screen:** Hanger Location - location of the hanger in airport Hanger Number - hanger Number Manager Name - Name of the manager Social Security No - Social security number of the manager Book - It is a radio button in all the rows Book Hanger - Submit Button used to book the hanger Reset - Button used to reset/clear the field values **Book HangerScreen:**

	Hanger Location - location of the hanger in airport Hanger Number - hanger Number Plane Id - plane id Book - Submit Button used to book the hanger Reset - Button used to reset/clear the field values
Trigger	Managers triggers the functionality once he/she enter into the system
Pre-Conditions / Assumptions	The administrator should have the details of the empty hanger and the details of the plane to allocate into hangers
Post Conditions	The system should allocate the plane into hanger
Success End Condition	On success a message should be displayed "**Hanger allocated for the plane<plane Id> from date<from date> to date <to date>**
Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. Administrator logs in to the system. 2. Clicks on Allocate Hanger option. 3. System should ask for start and end date of allocation. 4. System should display all the available (if the capacity > count of total allocation) hanger details on the basis of user start and end date inputs. 5. Select any one of the hanger and book. 6. Enter the plane id. 7. Click on allocate hanger button. 8. If any business exception occurs corrects the details and submit.
Business Rules & Validations	• Display all the available hanger details in the screen in a tabular form. • From date and to date should be in dd-mm-yyyy format. • From date cannot be back date. It may start from today. • From date cannot be blank' • To date should be greater than equals from date. If to date is blank, then take the from date as to date. • If the administrator selects any one of the hanger then display the next page. • To enter the plane id. Plane Id should come in a combo

	drop down. Only unallocated planes of that duration should be coming the drop down. • From date should be less than to date. Insert the above details into the database.

1.3.5 Airport Management System - Get Hanger Status

Airport Management System **Req-2.1.5**	**Get Hanger Status**
Manager search for the status of the hanger for the given dates. Hanger: It is a closed structure to hold aircraft or spacecraft in a protective storage.	
UI Interface Screen	**Hanger Detail Screen:** Hanger Number-hanger number. (A drop down box should be fetched from the system). From date booked- from date of booked. (A java Script date picker). To date Booked-To date booked. (A java Script date picker). Get Hanger Status - Submit Button used to get the hanger details. Reset - Button used to reset/clear the field values. **Hanger Status Screen:** Hanger Number- Hanger number. Hanger Location – location of the hanger in airport. From date booked- from date of booked. To date Booked-To date booked. Plane Id-Booked plane id.(Blank in case of available) Status- status of the hanger (available or booked).
Trigger	Managers triggers the functionality once he/she enter into the system
Pre-Conditions / Assumptions	The manager should have the details of the hanger no from date and to date.
Post Conditions	The system should display the status of the hanger.
Success End Condition	It should display the status of the sub hanger in the given dates

Failed End Condition	The system should throw an error message if one of the validations is wrong. The error message and error code needs to be identified during design. The validation and business rules are mentioned in the business rules section.
Steps & Actions	1. Administrator should log in to the system. 2. Administrator should click on view hanger status option. 3. System should display all the hanger number in the drop down box. 4. Select any one of the hanger. 5. Enter from date and to date. 6. Click on GetHangerStatus button. 7. If any business exception occurs corrects the details and submit.
Business Rules & Validations	• From date should not be greater than to date. • If any plane is not booked in the given date display status as available. • Else display status as booked.

1.4 Nonfunctional Requirements

1.4.1 UI Requirements: Inn

1. The front end should be user-friendly and pleasant
2. Any error message or exception displayed to the user should be user-readable (and not technical)
3. All entered values should be validated
4. The UI screens are designed with any theme color along with the logo of the company.
5. Exceptions should not be shown on console.
6. No debug message should come on console.
7. All the errors and exceptions should be logged to log file.
8. Log file path should be configurable.
9. Database connections should be configurable.
10. It should not be hard coded.

11. Requirements involving UI features should be taken care in console accordingly

1.4.2 Performance Requirements: Pnn

1. All front-end pages should be served up in less than 5 seconds post click when up to 1000 users are on the application concurrently.
2. Since it's a financial operations application, transactions should be safe and fast.

1.4.3 Installation, Deployment & Operational Requirements: Onn

1. Application Java and J2EE deployment with My SQL database or SQL Server database for persistence

   Or

   Application .NET Framework deployment with SQL Server database for persistence
2. This system must be up 24X7

1.4.4 Maintainability & Portability Requirements: MPnn

1. If the allocated memory space for the existing application needs to be increased then it should be possible without any impact to performance

1.4.5 Security Requirements Snn:

1. No unauthorized users should be able to log on to the system
2. User Id/Login Id should have combination of letter and numeric values, none of the special characters are allowed (#,@,$,%,&).
3. Validate the user id/login id and password, accordingly to the user it will redirect to the appropriate page.

2.0 Requirements / Use Case Summary Table

Business Requirement ID	Software Requirements ID	Short Description	Requirement Provider (Originator)	Priority	Complexity	Requirement type	Remarks

BR001	SR001	Adding Planes	Client	High	Medium	UI screen & backend processing	-
BR002	SR002	Adding Pilots	Client	Medium	Medium	UI screen & backend processing	-
BR003	SR003	Adding Hangers Details	Client	High	Medium	UI screen & backend processing	-
BR004	SR004	Allotting the planes into hangers	Client	Medium	Medium	UI screen & backend processing	-
BR005	SR005	Get Hanger Status	Client	High	High	UI screen & backend processing	-

Make /Buy analysis

2.1 Reusable components

The HMS can be made reusable as such to handle various Airport Management System.

Annexure

NA

Change Log

Please note that this table needs to be maintained even if a Configuration Management tool is

used.

<table>
<tr><th>Version Number</th><th colspan="4">Changes made</th></tr>
<tr><td>V1.0</td><td colspan="4">First version developed by Sagar S on Dec 25 2014.</td></tr>
<tr><td rowspan="6">V1.1</td><td colspan="4"><If the change details are not explicitly documented in the table below, reference should be provided here></td></tr>
<tr><td>Page no</td><td>Changed by</td><td>Effective date</td><td>Changes effected</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="6">V1.2</td><td colspan="4"><If the change details are not explicitly documented in the table below, reference should be provided here></td></tr>
<tr><td>Page no</td><td>Changed by</td><td>Effective date</td><td>Changes effected</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>