Application Overview of e-Enabler

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 28/04/2021 | 1.0 | Mahipal Reddy Keshava Reddy Gari | Application Overview Document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Approval Signatures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Title** | **Document Version** | **Signature** | **Date** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

[**OBJECTIVES 3**](#_Toc73378897)

[**1. BUSINESS 3**](#_Toc73378898)

[**1.1 Business Overview 3**](#_Toc73378899)

[**1.2 Business Process 3**](#_Toc73378900)

[**2. FUNCTIONAL 3**](#_Toc73378901)

[**2.1 Functional Overview 3**](#_Toc73378902)

[**2.2 FUNCTIONAL DETAILS 5**](#_Toc73378903)

[**3. TECHNICAL DETAILS 6**](#_Toc73378906)

[**3.1 Technical Overview 6**](#_Toc73378907)

[**3.2 Building and Deployment Process 6**](#_Toc73378908)

[**3.3 Application Architecture 7**](#_Toc73378911)

[**3.4 Source Code Control 7**](#_Toc73378912)

[**3.5 URL’s 7**](#_Toc73378913)

[**3.6 Technical Environments 8**](#_Toc73378914)

[**3.7 Hardware and Software Pre-requisites 8**](#_Toc73378915)

[**3.8 System Interface 8**](#_Toc73378916)

[**4. DATASETS & DATABASES 8**](#_Toc73378917)

[**4.1 Database Design / Database Dictionary 8**](#_Toc73378918)

[**5. PROCESS RELATED 9**](#_Toc73378919)

[**5.1 Major Design Assumptions 9**](#_Toc73378920)

[**5.2 System Access and Security 9**](#_Toc73378921)

[**5.3 Log Files 9**](#_Toc73378922)

[**5.4 Size /Complexity/Criticality 9**](#_Toc73378924)

[**5.5 Communication Protocols 10**](#_Toc73378925)

[**5.6 Test Environment 10**](#_Toc73378926)

[**5.7 Scope of Application Team 10**](#_Toc73378927)

[**6. SOFTWARE PRODUCTION ENVIRONMENT 10**](#_Toc73378928)

[**7. CONTACT DETAILS 10**](#_Toc73378929)

[**Other References 11**](#_Toc73378931)

# OBJECTIVES

The objective of the Application Overview Document (AOD) is to provide members of the delivery team with an overview of the application. The AOD describes the function of the application, the structure of the applications, the application configuration and the technical environment. This document refers to any existing related documentations.

This AOD document will be used to support the application and will always be kept updated throughout the life of the application.

# BUSINESS

## **Business Overview**

Jetstar Airways Pty Ltd, operating as Jetstar, is an Australian low-cost airline (self-described as "value-based") headquartered in Melbourne. It is a wholly owned subsidiary of Qantas. Jetstar is part of Qantas' two brand strategy of having Qantas Airways for the premium full-service market and Jetstar for the low-cost market. The airline operates an extensive domestic network as well as regional and international services from its main base at Melbourne Airport, using a mixed fleet of the Airbus A320 family and the Boeing 787 Dreamliner.

eEnabler is a custom web application used for software configuration management. This application reduced the use of excel sheets and manual management of different aircraft software’s.

It removes manual effort for maintaining records and updating of software parts for aircrafts.

## Business Process

eEnabler web application enables efficient management of airplane software parts configuration for Jetstar specifically for Boeing 787 and A380 fleet. It keeps the record of all software parts for the ongoing maintenance of aircraft to remove the manual effort with record keeping and updating of software parts

The main important functions of this application are:

* It generates Authorised Software Configuration (ASC) for Jetstar operating Aircrafts
* Creates software EC and manages EC’s
* Generates Airplane configuration reports for aircrafts
* Detects software configuration mismatches for the aircrafts that results in closing software EC’s or parts installation.

# FUNCTIONAL

## **Functional Overview**

The eEnabler has the functionalities such as :

* Creating new User Accounts
* Managing System Settings
* One Time Notification Settings
* Locking Aircraft
* Aircraft Down time Details
* Outstanding ACARS Responses
* Creation of SEC
* Generating ASC and Exception Reports
* Importing Software Mask & Baseline Configuration

This eEnabler uses keywords that is specific to airline industry and to the management of software engineering parts. Before configuring the objects and operating the application users must be familiar with the terminology.

The key terms how they are used is described below:

**Aircraft (A/c):**

Each aircraft in eEnabler is identified by its registered name.

**Loadable Software Airplane Parts (LSAP):**

LSAP represents the software engineering parts that are loaded on the aircraft.

**Maintenance Information System (MIS):**

It is the system used by airline for managing the maintenance lifecycle of an aircraft. eEnabler is designed to interact with MIS.

**SLID and Nomenclature:**

SLID is a software location Identifier that identifies the software location on aircrafts.

Software Nomenclature is a description of a software. If SLID is given there is a list of nomenclatures that must be present at this location.

**Software Part Number:**

This number represents the specific software and its version, it is formatted as combination of 15 Alphanumeric values.

**Software Mask :**

The software mask is a configuration file that is given by manufacturer of the aircraft which is a template containing the LSAP configuration for particular aircraft.

A software mask is imported for every A/c type and series. All Boeing B787-B series shares the common mask for all its aircrafts.

**Software Engineering Control (SEC) :**

Engineering Control (EC) is a set of tasks that is to be performed at every a/c maintenance. It is performed as per the given work order. A work order can be specific for single aircraft but same EC may be used for multiple work orders.

Software Engineering Control (SEC) is a task that captures all maintenance tasks related to software parts. It is an extension of EC managed by MIS. Moreover, Software EC specifically covers the installation of one or more software components to one or more SLIDs.

Software ECs are authorised by an authorised manager before being executed in maintenance. An SEC that is being executed as per the work order is referred as complied with the aircraft.

**Part Allocation :**

Within the context of eEnabler the information of a SEC is a list of software parts and details with a list of SLIDs where the software parts are to be installed is known as list of part allocation.

**Work Order :**

It is a collection of SECs that will undergo in a maintenance lifecycle.

**Software Configuration (SC) :**

It captures the complete set of software installed on an aircraft at a given point of time. There are multiple hardware modules that hosts different software of different nomenclature.

Since each Software EC ultimately impacts the software configuration of an aircraft, the initial

(baseline) configuration together with all Software ECs that are planned (authorized) or executed

(complied with) determines different types of software configurations, representing different stages of the maintenance planning and execution activities.

The **Baseline Software Configuration** shows the state at the initial delivery of the aircraft.

The **Authorized Software Configuration** represents a planned configuration that has been authorized and needs to be complied with an A/C as a part of Work Order.

The **As-Flying Software Configuration** is exported from the aircraft and captures the software

configuration as currently present in an aircraft at a given time. It is the result of the baseline

configuration and all accumulated Software ECs that were complied with for the aircraft.

**One Cycle Notification :**

eEnabler system will provide notification of SC issues within a time span of single flight cycle depending on the system availability and the Radar range for sending ACARS messages.

## **FUNCTIONAL DETAILS**

## User Interaction and Design

eEnabler interacts with PAGN system for communication between eEnabler and Aircrafts through ACARS messages.

eEnabler system creates ACARS messages to enquire information about as-flying configuration of particular parts. A/c sends responses via ACARS and jetstar uses PAGN system for messaging with the aircrafts.

**PAGN (Private ACARS Ground Network) :**

This system is used for messaging the A/c via ACARS, basically acts as a kind of transmission medium

Jetstar’s NAS server is used to exchange messages between the eEnabler and PAGN.

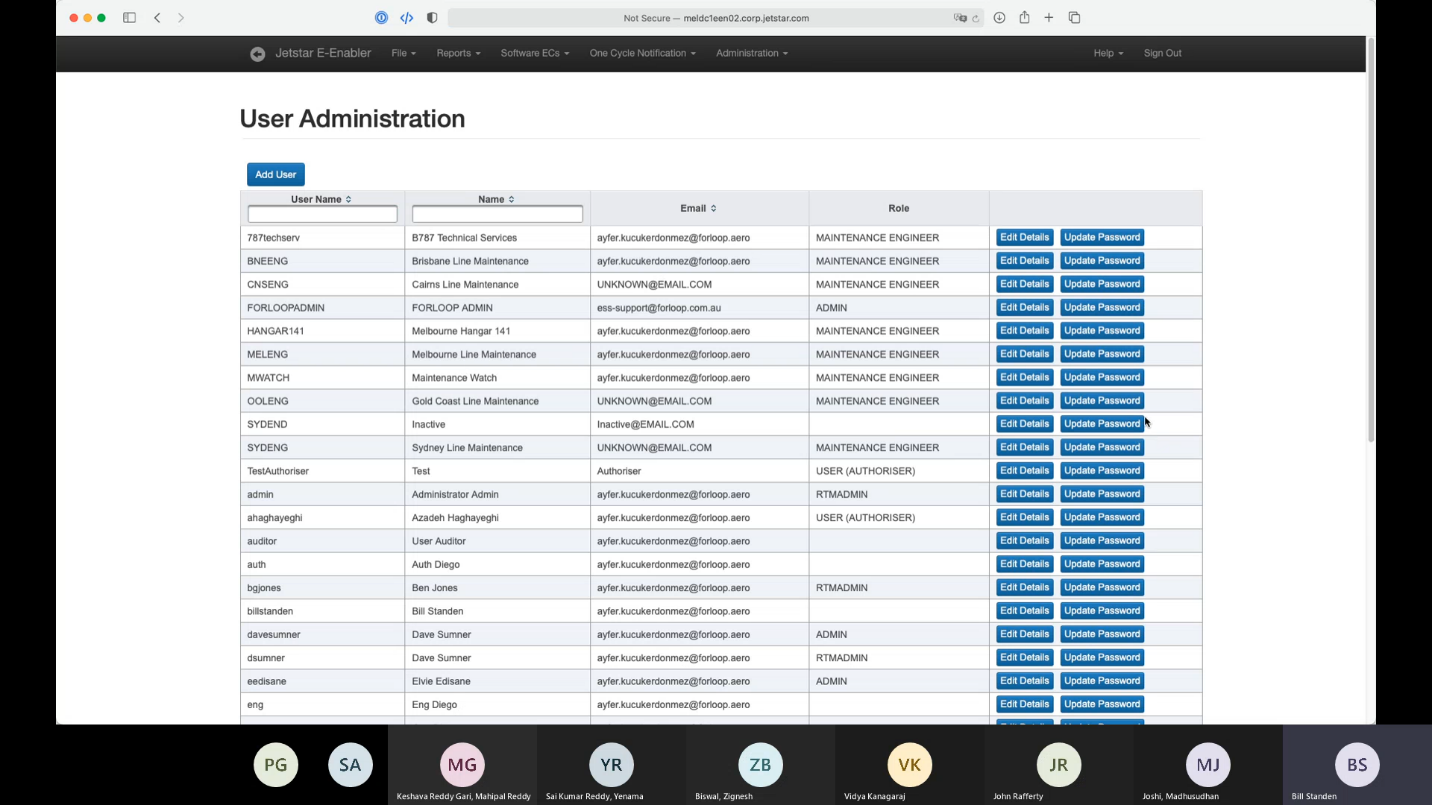
The NAS system folder contains the folders for sending and receiving the messages, those are:

* ENGINEERING/Forloop/ACARS\_SEND
* ENGINEERING/Forloop/ACARS\_SEND\_ARCHIVE
* ENGINEERING/Forloop/ACARS\_RECEIVE
* ENGINEERING/Forloop/ACARS\_RECEIVE\_ARCHIVE

## Creating New User Accounts

An existing user can add a new user inside the Administration under the User Administration and can assign the roles for that user.

Below Screenshot is taken from video recording, it will be replaced with clear picture once support team have access to the server



# **TECHNICAL DETAILS**

## Technical Overview

eEnabler application is built by using different technologies such as Java Spring, JPA, hibernate mapping and follows MVC architecture. (MVC is a model, view and controller).

eEnabler has its internal database AUTOEMN as well as TRAX database TRAXALL.

Internal database has read and write access and the data flows from trax database that contains all views.

* VIEW\_EEN\_EO\_COMPLIANCE
* VIEW\_EEN\_EO\_787
* **VW\_EEN\_PNMASTER\_SOFTWARE**
* **VW\_EEN\_AC**

## Building and Deployment Process

Required packages for deploying eEnabler :

* Java SE Development Kit 8,
* Apache Tomcat 8.5,
* eEnabler web archive (war)

eEnabler uses Maven and a set of plugins for building the application.

|  |  |
| --- | --- |
| Command | Description |
| mvn clean package | To build the code |
| Clean | clears already built packages and artifact ids |
| Package | takes the code and places it in a package called as JAR |

Note: - Run above commands from CMD (command prompt).

### General Folder Structure in IDE

|  |  |
| --- | --- |
| Module Name | Description |
| eEnabler-Web | It is a module that interacts with web services and each module has its own pom.xml files which consists of maven files and its dependencies of that module.  This module contains webapp folder that consists of jsf files which has the combination of Java and HTML files.  Resources folder keeps necessary configuration files. |
| eEnabler-Services | This module manages all business logics and interacts with UI and DB |
| eEnabler-Model | Used for data abstraction layer that uses hibernate which maps to DB relational interaction. |
| eEnabler-db | Manages all DB changes, generates and migrates sql files. Any DB updates can be done through db\_changelog.xml file that is resided inside src folder under ChangeLogs folder. |

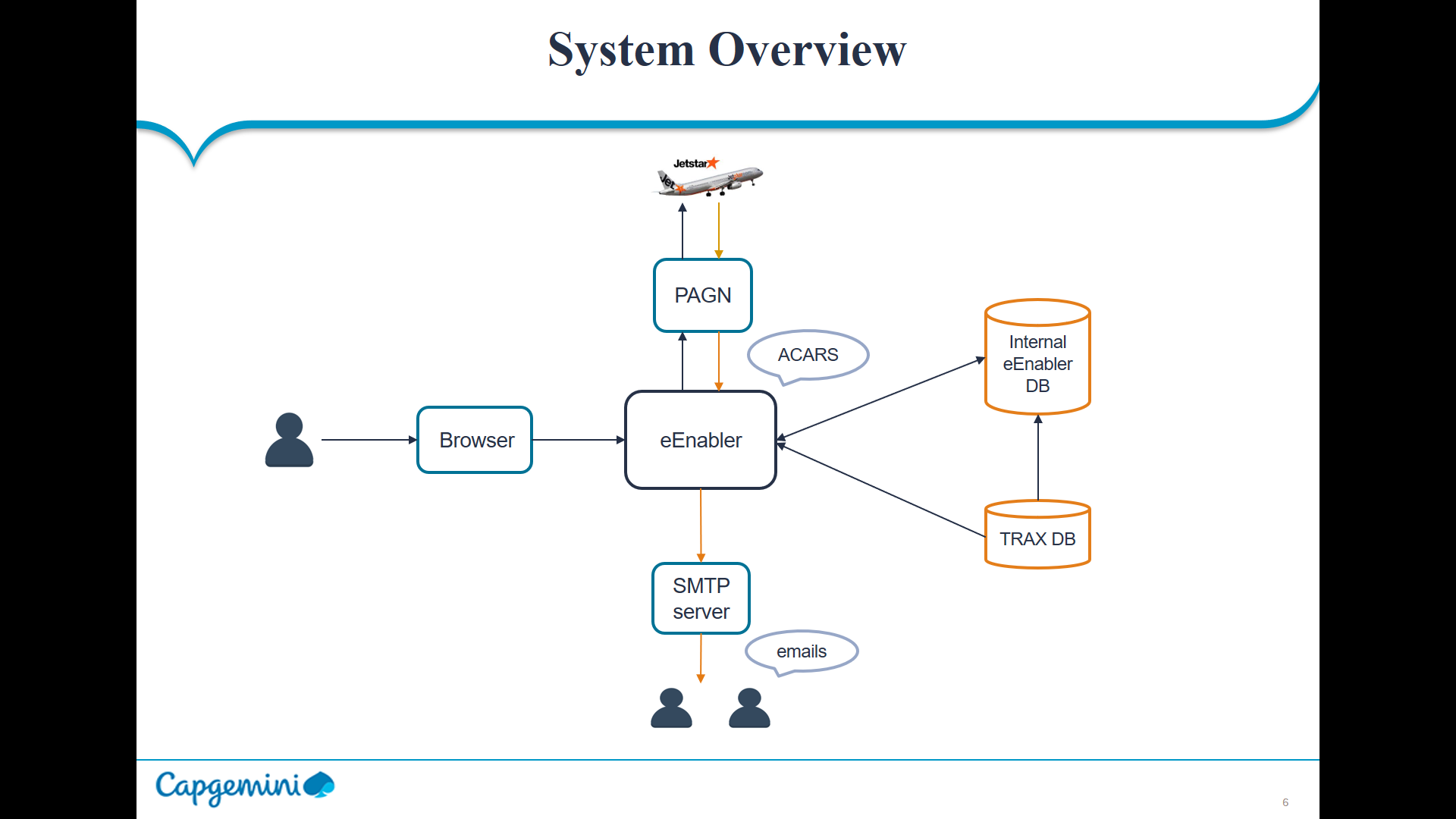
### eEnabler Released Versions

eEnabler has released four versions as shown below :

|  |  |
| --- | --- |
| Version Number | Details |
| 1.1 | it has basic software parts and workflow of system also called as clean release |
| 1.2 | this has added additional functionalities for software configuration approval process and also supports for aircraft lockdown process which is around restriction of software configuration. |
| 1.3 | added a set of new reports and historical process information, known exceptions and user management functionality |
| 1.4 | significant added integration with Jetstar PAGN system and ACARS messaging that flows through PAGN. |

## Application Architecture

Below diagram shows the interaction of eEnabler with database and the user.



eEnabler uses MVC architecture and when a user makes web service request using URL instance the server gives response with the GUI. User can login into account by providing the login credentials.

After validating the credentials eEnabler Home page will be displayed and that user now has an access to the application. When user needs information from aircraft he can send ACAR message using PAGN system and when the aircraft gives the response, user gets email notification from eEnabler SMTP server.

eEnabler system uses two databases for its operational purpose i.e., EENABLEP which is internal database of eEnabler and TRAX database, however it will not directly interact with two databases, the data of TRAX is sent to internal database which has read and write access and from internal db we will get data to eEnabler.

## Source Code Control

|  |
| --- |
| GitHub access links |
| <https://github.com/jetstar-engineering/jq-eng-eenab> |
| <https://github.com/jetstar-engineering/jq-eng-eenab/blob/head-jenkins/pom.xml> |
| <https://github.com/jetstar-engineering/jq-eng-eenab/tree/head-jenkins/eenabler-web> |

## URL’s

|  |  |
| --- | --- |
| Environment | URL |
| Production | <http://meldc1een01.corp.jetstar.com:8080/e-enabler/> |
| Development | <http://meldc1een02.corp.jetstar.com:8080/e-enabler/> |
| UAT | [http://meldc1aem03.corp.jetstar.com:8080/e-enabler/](http://meldc1aem03.corp.jetstar.com:8080/e-enabler/%20%20) |

## Technical Environments

Development or testing environment for this system is existing on meldc1een02 server which has Linux operating system.

**Instances and Application servers**

|  |  |
| --- | --- |
| Instance | Application Server |
| PROD | MELDC1EEN01 |
| DEV | MELDC1EEN02 |
| UAT | MELDC1AEM03 |

## **Hardware and Software Pre-requisites**

It requires an operating system installed with Java IDE

|  |  |
| --- | --- |
| Software | Version |
| Java | 8.0 |
| Oracle | 11g |
| Apache Tomcat | 8.0 & above |

## System Interface

N/A

# DATASETS & DATABASES

## Database Design / Database Dictionary

The database and its servers are :

|  |  |  |
| --- | --- | --- |
| Instance | Database Name | Database Server |
| PROD | EENABLEP | MELDC1TRXH01 |
| DEV | EENABLED | MELDC1TRXH02 |
| UAT | EENABLET | MELDC1TRXH02 |

TRAXALL database contains views that is used for reading the data and synchronizes the data with internal database of eEnabler.

* VIEW\_EEN\_EO\_COMPLIANCE
* VIEW\_EEN\_EO\_787
* **VW\_EEN\_PNMASTER\_SOFTWARE**
* **VW\_EEN\_AC**

# PROCESS RELATED

## Major Design Assumptions

N/A

## System Access and Security

Before accessing the eEnabler application it has a set of users in its database who has the permissions according to their roles specified. Every user has to access the application by the password.

After the user gets created one or more roles can be assigned and the access levels can be determined for the system functions.

|  |  |  |
| --- | --- | --- |
| Role | Role Description | Permissions |
| User | A standard user who involves in the usual SEC lifecycle | Can Create/Modify/Delete SECs  Can be an authorizer  Cannot change system settings  Cannot import mask or baseline |
| Admin | An administrative user who performs initial and ongoing setup and configuration of  eEnabler as well as SEC maintenance activities | Access to all functions including system settings, except for vendor specific  settings |

## Log Files

* **Log rotation** :/etc/logrotate.d/tomcat8
* **Service to start/stop:**  /etc/init.d/tomcat8
* $TOMCAT\_HOME/conf/logging.properties set level of logging

## Size /Complexity/Criticality

|  |  |
| --- | --- |
| Priority | Description |
| Severity Level 1  (Critical) | Total service failure or loss of critical business functionality resulting in significant in loss of revenue opportunity or legal/privacy exposure and/or Security Breaches (i.e. Virus or security exposures). |
| Severity Level 2  (High) | Serious failure or inhibited performance of a system function affecting multiple/critical users, where no acceptable work-around is available or impact results in loss of revenue. |
| Severity Level 3  (Medium) | Failure or inhibited performance of an individual function or component, affecting one or more users, for which an acceptable workaround is available. |
| Severity Level 4  (Low) | Low or non-impact issue, or cosmetic problem, where a fix or attendance can be scheduled within an agreed time. |

## Communication Protocols

eEnabler uses SMTP (simple mail transfer protocol) for communication between aircrafts, eEnabler and users via the PAGN system.

## Test Environment

eEnabler has its own Test UAT environment that is deployed in meldc1aemo3 server. If any changes need to be done to eEnabler then we can test it on UAT server after successful execution, we can move it to production environment.

## Scope of Application Team

* Check if server is available
* Check if application and databases are up and working
* We are responsible if the user is not able to create any software configuration parts in application
* Source Code Management
* High Disk Utilization and generating Reports.
* False signalling folders of Inbound and Outbound ACARS messages
* If application performance is slow then we have to perform health check of database and the application connections.
* When Incident, Request, Change and problem tickets are encountered we have to respond to them according to the SLAs.

# SOFTWARE PRODUCTION ENVIRONMENT

The following are the files required for building application :

|  |  |
| --- | --- |
| Files | Description |
| TOMCAT\_HOME/webapps/e-enabler.war | Copy eEnabler web archive (war) into tomcat webapps directory |
| TOMCAT\_HOME/bin/startup.sh | To restart Tomcat Server |
| /etc/eenabler/application12.properties | eEnabler reads its configuration from the properties file |

This properties file contains the connection parameters between eEnabler internal database and TRAX database connection. And also has SMTP notification sending functionality parameters.

# 7. CONTACT DETAILS



**Capgemini Support Team**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Email Id | Contact |
| Pankaj Nanda | Engagement Manager (Onsite) | [Pankaj.Nanda@jetstar.com](mailto:Pankaj.Nanda@jetstar.com) | +61 431 026 492 |
| Sunil Mehta | Delivery Manager (Offshore) | [Sunil.a.Mehta@capgemini.com](mailto:Manojmanikanta.kothamasu@bluescopesteel.com) | +91 9886330813 |

**Customer Representative**

|  |  |  |
| --- | --- | --- |
| Name | Title | Email ID |
| John Rafferty | Business Owner | <John.Rafferty@jetstar.com> |
| Benjamin G. Jones | eEnabler owner | <benjamin.jones@jetstar.com> |
| Dave Sumner | eEnabler owner | <Dave.Sumner@jetstar.com> |

## Other References

Applications support documents