Digital Technology and Solutions,

Research and Development

User Requirements Specification

|  |  |
| --- | --- |
| Accolade Number: | A-16528 |
| Project Name: | Trebuchet |
| Part Number(s) | N/A |

|  |  |
| --- | --- |
| Document Number: | RECIPE BUILDER URS\_DRAFT 5 |
| Revision Number: | 01 |
| Save Date: | 2 March 2021 |
| Author: | umesh\_rao@pall.com |

***CONFIDENTIAL****: This document is the confidential work product of Pall Corporation and no portion of this document may be copied, published, performed, or redistributed without the express written authority of   
a Pall corporate officer. This document is furnished for use by the assigned recipient only. This material is confidential and is not to be divulged to others without specific written permission from Pall Corporation.*

Table of Contents

[1 Approval Log 3](#_Toc40797351)

[2 Introduction 4](#_Toc40797352)

[2.1 Related Documents 4](#_Toc40797353)

[3 Overview 5](#_Toc40797354)

[3.1 Project Description 5](#_Toc40797355)

[3.2 Scope 5](#_Toc40797356)

[3.2.1 In Scope 5](#_Toc40797357)

[3.2.2 Out of Scope 6](#_Toc40797358)

[3.3 Key objective and Benefits 6](#_Toc40797359)

[4 Requirements 7](#_Toc40797360)

[4.1 Hardware & Infrastructure Requirements 9](#_Toc40797361)

[4.2 Recipe Builder Requirements 9](#_Toc40797362)

[4.3 Unit Procedure Requirements 10](#_Toc40797363)

[4.4 Operation, Phases and Transitions 11](#_Toc40797364)

[4.5 User Experience Requirements 11](#_Toc40797365)

[4.6 Alarm and Event Management 12](#_Toc40797366)

[4.7 Data Collection 12](#_Toc40797367)

[4.8 Error Management 13](#_Toc40797368)

[4.9 Historian 13](#_Toc40797369)

[4.10 Software Maintenance and Updates 13](#_Toc40797370)

[4.11 Report 13](#_Toc40797371)

[4.12 Audit Trail 14](#_Toc40797372)

[4.13 Access Control 14](#_Toc40797373)

[4.14 Disaster recovery and Back-up Requirements 14](#_Toc40797374)

[4.15 Documentation Requirements 15](#_Toc40797375)

[4.16 Regulations 15](#_Toc40797376)

[4.16.1 21 CFR Part 11 / Annex 11 requirements 15](#_Toc40797377)

[4.16.2 Data Integrity requirements 16](#_Toc40797378)

[4.17 Operational 16](#_Toc40797379)

[4.17.1 Servicing requirements 16](#_Toc40797380)

[4.17.2 System flexibility 16](#_Toc40797381)

[4.18 Ownership 16](#_Toc40797382)

[4.19 Product Privacy and Security policy requirements 17](#_Toc40797383)

[5 Glossary 18](#_Toc40797384)

[6 Revision Index 20](#_Toc40797385)

# Approval Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **name** | **title** | **signature** | **date** |
| Document originator | Umesh Rao | Project Manager |  |  |
| UX Approval | Krista Brown | UX Lead |  |  |
| technical approval | Jun Tao | Software Lead |  |  |
| VENDOR APPROVAL | Bindya Ganesh | TCS (Vendor) Lead |  |  |
| Automation approval | Daniel McCarthy | Automation Lead |  |  |
| PASS Approval | Andreas Flagmeyer | PASS Lead |  |  |
| Marketing Approval | Karl Rogler | Product Manager |  |  |
| project approval | Bryan Shepardson | Project Lead |  |  |
| quality | Sneha Vyas | Quality Lead |  |  |

\*Signatures will be obtained electronically via DocuSign

# Introduction

This document has been produced by Pall Corporation, to define the scope of work that will be applied to the Recipe Builder – code named Project Trebuchet.

Recipe Builder is a qualified suite of software applications meant as an add-on software for Pall automated products. This software (or a suite of software modules) are designed to allow for end user to create / modify / view automation recipes with parameters and to be passed onto ANSI/ISA–88 Batch compliant system – and Wonderware® Batch Management™ is proposed for this development.

Within the Recipe Builder, individual unit operation Editor (UoE) are used for creating / modifying / viewing unit operations, including phase definitions, transitions, flows and parameters for individual phases.

Customer copies of this document are not controlled documents and will not be up-issued if changes are made to the master document. Customer copies of this document show electronic duplicates of these signatures for presentation purposes only.

## Related Documents

|  |  |  |
| --- | --- | --- |
| **title** | **document number** | **issue** |
| Batch Control | ANSI/ISA-88 | Current |
| PALL Style Guide |  | Current |
| Single User Buffer Management Functional Specification | A-15650\_SUBM\_Auto\_FS | 2.0 |
| Single Use Buffer Management System Design Specification | A-15650\_SUBM\_Auto\_SDS | 1.47 |
| Danaher Product Privacy and Security By Design Policy | Danaher Policy | Current |

# Overview

## Project Description

In current environment, most commercially off the shelf automation system consists of Batch Engine with user interface to create, update, view, manage and execute Batch based recipes. A recipe is a necessary set of information that uniquely defines the production requirements for a specific product. Typically, end users require specialized engineers to be able to define and manage recipes and are specific to the underlying control system being used.

Pall Life Sciences intends to develop a software application that will:

* allow for users to create recipes and Unit Procedures – *online or offline* with synchronization capabilities to S88 based Batch systems – Wonderware® Batch Management™ will be used for development purposes.
* include graphical user Interface for creating the area layout, recipe formulas and unit procedures parameters
* include graphical user Interface for defining phase parameters and the phase structure
* include rules Engine for defining unit’s relationship relative to the train, enforcement of limits, engineering units, physical model, and parameters constraints.

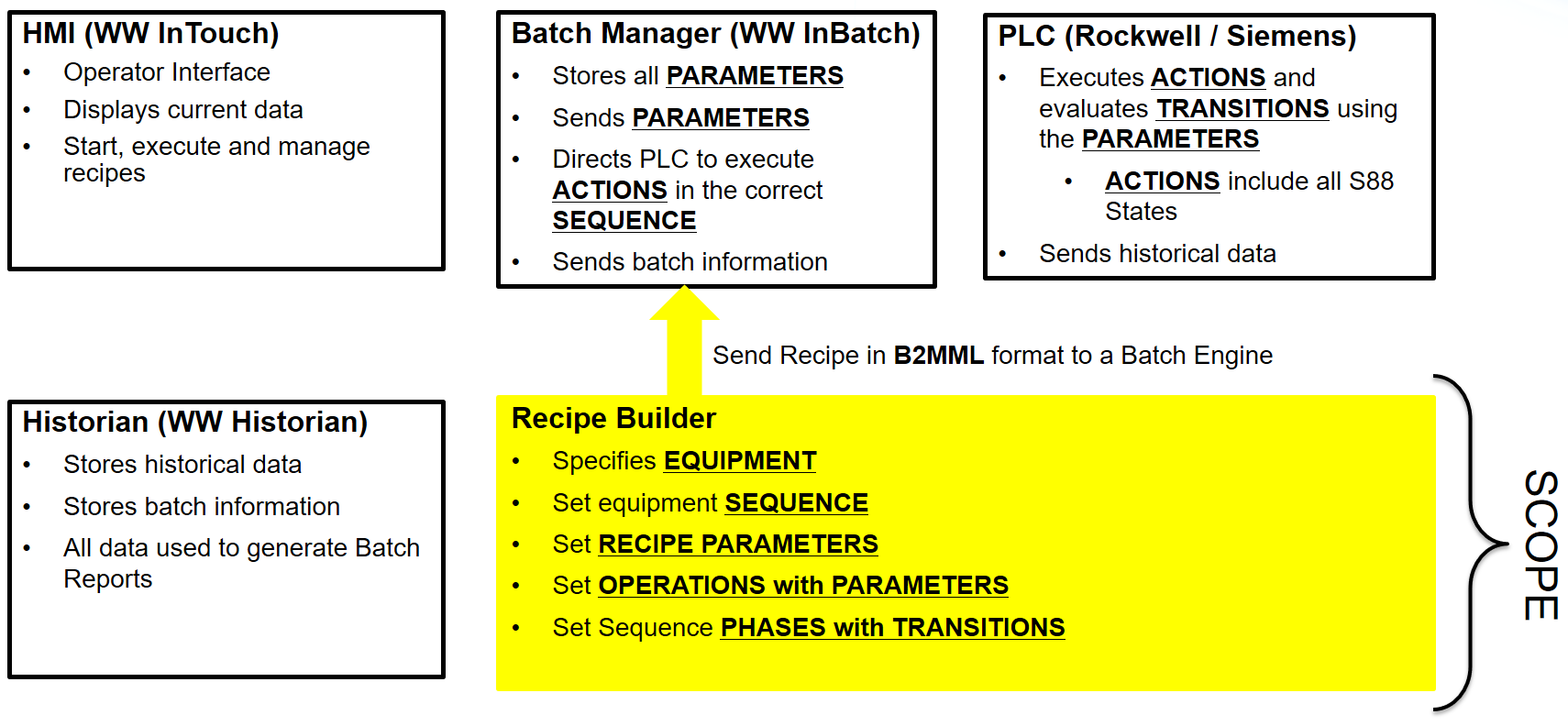
## Scope

### In Scope

Pall is developing a software with graphical user interface to allow for end users to build a recipe in accordance with S88 model.

Unit operations Editor (UoE) for Buffer Management and Bulk Fill is in the scope and is meant to allow for users to define the parameters for phases, allow for sequencing of operations and the phases for these Unit Procedures. The application is expected to be modular to allow for additional UoE as and when they are developed without impacting existing functionality. Buffer Management, Bulk Fill and Direct Flow Filtration UoE are in scope for the initial release.

From an integration standpoint, Wonderware Batch Management™ stateless APIs library will be used to synchronize the BatchML (B2MML) from Recipe Builder in online mode. In offline mode, this will only allow for creating / updating the recipe and would not interface with Batch Management application. The illustration indicated below provides a high-level representation of the Recipe Builder interfaces with Wonderware Batch Management and all the various modules involved.



### Out of Scope

* Recipe Builder is not a Batch Engine, Control system or SCADA system. Recipe execution and monitoring is outside the boundary of this application.
* Analytics, Graphing and mathematical modeling of the data is beyond the scope of the Recipe Builder.
* Future UoE development for other automated equipment will be handled via Change Control procedures.
* Interface with Manufacturing Execution Systems (MES) is out of scope.
* Scheduling of Batch, monitoring the batch being executed and creation of control recipe is not within the scope of this application.

## Key objective and Benefits

Today, process models are created using the control system vendor provided *Editor* or Batch client– if any. A typical batch processing plant is made up of units, process classes, connections, transfer classes, process, and transfer-phases.

* A unit is a piece of equipment that processes materials such as reactors, mixers and blenders. A unit can also hold materials such as holding tanks and bulk storage bags.
* Process classes are used to define process capabilities. Each unit in the class has the same processing capabilities and/or performs the same functions.
* Connections define equipment that transfers material from a source unit to a destination unit.
* Transfer classes are used to define transfer capabilities where all source units are in the same process class and all destination units are in the same process class.
* Process phases and transfer phases are used to describe capabilities of process and transfer classes respectively.

An integral part of process modelling involves defining specific tags for units, processes, connections, and transfers. Tags allow mapping data between Batch Engine and Controllers via the SCADA system. Building / Editing a recipe requires an automation engineer trained in developing and managing recipes. The current editors in market are complex and do not provide a user-friendly interface as acknowledged by key industry experts. While the market constraints are well understood, there are no incentives for the vendors to build a platform independent interface that utilizes today’s technology.

Hence, Pall is taking the initiative to develop a Recipe Builder that end users can use to build and maintain their own recipes.

# Requirements

The following words are used to signify the requirements in this specification:

* “must”, “shall” or “required” mean that the behaviour is an absolute requirement of the specification
* “must not”, “shall not” or “forbidden” mean that the behaviour is an absolute prohibition of the specification
* “should”, “recommended” or “should not” mean that there may exist valid reasons in particular circumstances where a deviation from the requirement or prohibition is acceptable or even useful. However, the full implications must be understood, carefully weighed and agreed upon before deviating from the specification
* “may”, “not required” or “optional” mean that an item is truly optional. However, an implementation that does not include a particular option must be prepared to interoperate with another implementation (or later version) that does include the option, though perhaps with reduced functionality. Similarly, an implementation that does include a particular option must be prepared to interoperate with another implementation that does not include the option (except, of course, for the feature the option provides)
* For Priority – HIGH means the functionality shall exist exactly as specified and fully tested. Medium priority indicates the functionality shall exist, but alternatives are acceptable and fully tested. LOW priority indicates the functionality may or may not exist with justification and testing is optional.



This figure illustrates the interaction between various components of a typical control system and how the Recipe Builder fits into the scheme. The proposal is to include a layer of application that sits on top of Wonderware Batch Management™ and gets data from Batch Management and WW ArchestrA™.

## Hardware & Infrastructure Requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Application *front end* shall run on a standard desktop class equipment. | HIGH |
|  | The application backend shall be capable of functioning in a virtual environment. | MEDIUM |
|  | Application front end shall be capable of running on Windows 10 or higher, and backend on Windows 2019 server or higher.  Note: *It is assumed that Central Historian shall be capable of hosting the Recipe Builder centralized database.* | LOW |
|  | This application front end shall be capable of running in Wonderware InTouch™ HMI. | MEDIUM |
|  | The application shall be adaptable to tablets / touchscreens in the future. Resolution independent! Just the overlay testing. | LOW |

## Recipe Builder Requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Recipe Builder front end shall either be a Web application with local and central data repository. | HIGH |
|  | Application shall interface with Wonderware Batch Management™. | HIGH |
|  | Application front end shall be capable of functioning on standard 4 GB RAM and 10 GB HDD computing platforms. | MEDIUM |
|  | Recipe Builder shall be capable of being used *offline* – wherein user shall have the ability to define / update recipes when not connected to the Recipe Builder server. Application capability shall not be degraded except for ability to synchronize, user roles/actions, rules management, look up operations and phase definitions and approval for transferring to Batch Management. Include the checkout feature for offline mode. | HIGH |
|  | Recipe Builder shall be capable of automatic versioning of recipes. Updating an existing recipe must create a new version of the recipe and each change shall include a log entry. | High |
|  | Recipe Builder shall have a viewer for text based comparison of ,merging, highlighting two recipes at a time. | low |
|  | Recipe shall include, at a minimum, following parameters:  Recipe ID, Version, Date, Description, Header, Equipment Requirement, Formula, Procedure Logic, Recipe element and associated author and recipe drafted / approved / released / obsoleted information.  **Note:** *Specific requirements such character length, data type and other structural requirements should align with Batch Management 2017.* | HIGH |
|  | The B2MML generated file shall be capable of being transferred with confirmation to Wonderware Batch Management™ system whilst in online mode. | HIGH |
|  | The Recipe Builder shall have a user management with different user levels. The rights associated with each role shall be configurable. Users from connected domain shall be capable of being added to the role. | HIGH |
|  | The Recipe Builder shall be capable of domain based user authentication or application level role based security. | Medium |
|  | The application shall allow for creating a new recipe from an existing recipe. | HIGH |
|  | Recipe Builder shall allow for transfer of recipes upon approval to Batch Management™ for execution. Approval shall be required in online mode by at least one authorized user that is NOT the author. Conversely, the offline version may not be exported unless approved.  Workflow needs to be defined in the UX. | HIGH |
|  | A separate user interface shall allow for users to create, add, update rules for parameters and layout being specified. For instance, upper and lower limits for temperature may not exceed 100 and 0 respectively. | Medium |
|  | Recipe Builder shall allow for creating a train layout using the available unit procedures. Adding a new unit procedure shall not impact the existing unit procedures. | HIGH |
|  | The Recipe Builder shall have the ability to define the train behaviour between various unit procedures with a minimum of following states – Start / Restart, Running, Aborted, Stopped, Held. Other states shall | HIGH |
|  | In online mode, for the parameters, application shall be capable of interfacing with the Historian to recommend parameters based on the previous batch runs – where this data is available (designated as golden Batch). | LOW |
|  | The Plant Model shall have the user interface for configuration of the train. The User will define:   * Units in the train with pre-configured Unique ID * The order in which the units are positioned physically * Physical unit associated with each logical position   Note: Allocation of equipment may be defined in the Recipe or utilize this as place holder for run time binding by the operator.  Recipe validation/approval not possible with place holder. Equipment instance (physical units) should be allocated when defining the recipe | HIGH |
|  | Recipes shall be approved prior to transfer to the Batch Management system. The user shall NOT be required to perform any task using Batch Client – other than to select and start the recipe from HMI. | HIGH |
|  | Each of the UoE for Unit operations shall be modular so as to be added to the Recipe Builder without impacting already existing Unit Operations Editor. | HIGH |
|  | The recipe shall be associated to a specific version of the Unit procedure(s). | HIGH |

## Unit Procedure Requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The Unit Operation Editor shall allow the creation of unit procedures by the addition of one or more operations. | HIGH |
|  | The following information shall be recorded for each unit procedure:   * unit procedure’s name * version number * status * author’s name * date and time of creation and last modification * Equipment class | HIGH |
|  | Each unit procedure shall have a unique name. Check for character limit on the WW side. | HIGH |
|  | Each unit procedure as a minimum shall have the following status:   * In development – when Unit procedure is being developed * Released – When sent to Wonderware Batch management * Archived – When no longer in use * Approved   Other status may be added as needed for development purposes.  Need to workflow this process. | HIGH |
|  | Operations within a unit procedure shall be executed in the order they are stored (from lowest to highest operation number). | HIGH |
|  | It shall be possible to insert, re-order, delete operations. Any delete operations will need user confirmation to prevent accidental deletions. | HIGH |
|  | Operations shall only be linear without any branching or looping. | HIGH |
|  | It shall be possible to alter the sequence of operations within a unit procedure. | HIGH |
|  | The Recipe Builder shall allow the user to enter up to four thousand (4000) parameters for each of the unit operations. | HIGH |
|  | At a minimum, each parameter shall include a default value, Actual value, Engineering Units and the alarm limits, if any. The application shall include the capability to enable acceptable ranges to be configured for all data input fields. | HIGH |
|  | Each Unit procedure shall have its own GUI for accepting parameters and arranging operations with phases with transitions. | HIGH |
|  | Only phases that are applicable for the unit procedure shall be available for use within the Unit Operations Editor, including user defined phase. | HIGH |
|  | Each of the unit procedure shall allow for the following parameters:   * Operational Setpoints * Any sequence choices that the unit requires * Declaration of its own operational states (e.g., upsets) * Define reactions to external operational states * Alarm Limits * Operator Prompts | HIGH |
|  | Alarms will be associated with each phase – in that only the alarms required for that phase will be activated while other alarms shall need to be de-activated. | HIGH |

## Operation, Phases and Transitions

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | An operation contained within a unit procedure shall contain a sequence of phase(s). | HIGH |
|  | Operations shall be developed from   * blank operation where phases can be added or, * user editable operations / templates where each phase within this template can be configured/ new phases added or, * Pre-populated operation with a fixed set of phases where parameters and setpoints can be set. | HIGH |
|  | Existing Unit procedure phases with its parameters must be available for selection to allow for creating recipes.  **Note:** *To allow for changes from automation / Batch Management to be carried over rather easily, an import script must allow for carrying over the phase names and the parameters with default values into Recipe Builder application. A template / Excel spreadsheet must exist for capturing phase parameters as defined within automation.* | HIGH |
|  | The application shall accept phase inputs for the following parameters:   * Mode of Operation * Controller status * Controller setpoints * Alarm status for each parameter (active/not active) * Alarm limits (HH, H, L, LL) * Parameters for the user defined phases, if any | HIGH |
|  | The following shall be selected as a transition condition:   * Each analogue input (<, <=, >=, > entered value) * Each digital input (external or internal) * User handshake * Timer/counter | HIGH |
|  | The GUI shall allow for the user to select phases and provide a mechanism to enter the values for each of the selected phases. | HIGH |
|  | The user can select as many phases (phase instances) as required from the available phases and sequence linearly.  **Note:** *Looping / branching / specifying transition conditions exists only for user defined phase.* | HIGH |
|  | The GUI must allow for phases to be added / removed / updated / re-ordered as needed. | HIGH |
|  | All the phases with its parameters are defined in Functional Specifications for Bulk Fill Rev x.x must be available for the Bulk Fill Unit Procedure. | HIGH |
|  | For each user definable / custom phase (or instance of custom phase), it shall be possible to define up to four (4) transition (T#) conditions for each phase using a graphical editor that allows to combine four different transition conditions using the following scheme: (T1 and/or T2) and/or (T3 and/or T4)  Automation for custom phase.  UX when saved as a template | HIGH |
|  | AND / OR logic shall be available per transition for custom phase. | HIGH |

## User Experience Requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Recipe Builder graphical user interface shall be compliant with Pall Style Guide.  Send it to tcs. | HIGH |
|  | Graphical User Interface shall be designed to be used with keyboard and mouse. | HIGH |
|  | GUI shall be aligned to the Bulk Fill UX wireframe. Where the alignment is not feasible, UX modifications are to be sought, update the wireframe and implemented. | HIGH |
|  | Standard workflows and tasks shall be intuitive, easy to use and not complicated. Not defined. | HIGH |
|  | The user specified parameters, sequences shall be verified against the rules specified via the application rules engine. In other words, the application shall provide guidance to ensure what the user specifies for each phase can be safely and accurately performed by the machine. | MEDIUM |
|  | Errors in the recipe development process shall be displayed at the time of input but shall allow for the user to move forward with development with errors present. However, the user shall NOT be able to send recipe to Batch Management with errors. | HIGH |
|  | The application shall be built to enable the creation of screens specific to each type of unit operation, but shall maintain the same look, feel and consistency across multiple unit operations. | MEDIUM |
|  | Graphical User Interface shall adhere to the prototype provided by UX. | HIGH |

## Alarm and Event Management

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Audit trail information (records / events stored in the centralized database) shall include recipes that were created, updated, old and new version for updated recipes, and details around synchronization with Batch Management™ with error logs. | HIGH |
|  | Recipe Builder shall check for exceptions / failures upon transferring a recipe from the Builder to Batch Management. | HIGH |
|  | Transferring recipe to Batch Management shall check for network connectivity and must complete the transfer as a single transaction to minimize risk of erratic data transmission. | HIGH |
|  | Prior to transferring the recipe, the Recipe Builder shall ensure all the unit Ids, specified parameters, volumetric balances and the recipe layout is aligned with plant layout structure.  Note: The algorithm to validate a recipe with regard to the plant layout has to be defined in detail. What happens with old approved recipes when removing for example, a unit from the plant layout? Is it allowed to modify the plant layout, or is this a static information? | MEDIUM |
|  | All the events, including creating a new recipe, updating the recipe, users logging in and out, recipe transfers to Batch Management™ shall be logged with the user Id, date time stamp and status. This shall be accessible with filters for user ID and date/time range. | LOW |

## Data Collection

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | All of the recipes defined shall be stored in a centralized database and checked out recipes shall be stored in local database. All the data transmission and storage shall be encrypted and access will be role based. | HIGH |
|  | The local recipe data shall be capable of synchronizing with the centralized database. In online mode, the application will automatically connect to the central database. There shall be a visual representation during such synchronization and conflicts, if any shall be logged as errors. | Medium |
|  | While synchronizing a recipe to the central Batch Management, conflict resolution mechanism shall exist so that multiple users with same recipe names do not overwrite each other’s recipes or existing recipes within Batch Management. | HIGH |

## Error Management

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Errors shall be displayed to the user if there is no connection to application database. | HIGH |
|  | Indication / status should be displayed to the user if there is no connection to Central Recipe server. | HIGH |
|  | Errors shall be displayed to the user if there are unresolved parameters or transmission of the recipe to Batch Management fails or results in exceptions. | HIGH |
|  | All the errors shall be logged to the application events log stored in the local secure database and synced up to the centralized database. | MEDIUM |
|  | The application shall be capable of exiting gracefully in the event of unhandled exceptions. | HIGH |
|  | The error level for logging should be “error” or “debug”. | LOW |

## Historian

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The Recipe builder shall be capable of interfacing with WW Historian and/or OSISOFT PI for recommending parameters based off designated golden batch. | LOW |

## Software Maintenance and Updates

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The suite of software applications shall allow for automatic updates upon user approval and authentication. Each software update shall be accompanied by Pall issued Software Release Notes. and published via secure cloud location. In the event of lack of network access, the application when connected to the Batch Management server should check for presence of any new updates. | HIGH |
|  | Pall R&D shall be responsible to keep the application up to date. Pall Technical Operations (PASS) / Service / Support shall be responsible for deployment. | MEDIUM |

## Report

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The Recipe Builder shall be capable of displaying the existing recipes in an easily readable and printable report format. | HIGH |
|  | Each of the unit procedure shall be capable of being displayed in an easily readable and printable report format. | HIGH |
|  | The plant layout shall be capable of being displayed in a report format with the Unit ID and description. | MEDIUM |
|  | Application Audit Trail per recipe of all actions / modifications and deletions shall be capable of being displayed in an easily readable and printable report format.  Audit trail relevant actions will not be changes in recipes. These changes can be tracked in the versioning comparison of recipes | MEDIUM |
|  | Materials List for each recipe shall be capable of being displayed in an easily readable and printable format. | HIGH |
|  | Application error logs shall be accessible and printable on demand. | LOW |

## Access Control

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The Recipe Builder shall have access controls based on roles – either as domain or application based. | HIGH |
|  | The Recipe Builder shall have the **Administrator**, **Engineer**, **Viewer** roles with ability to create custom roles. Administrator role shall have the ability to administrate users, review audit trail and error logs. | MEDIUM |
|  | Engineer role shall have the ability to create, update recipe, approve unit procedures and send the recipe to WW Batch Management™. | HIGH |
|  | Engineer role by default shall have the ability to update parameters and send the recipe to WW Batch Management™. | HIGH |

## Disaster recovery and Back-up Requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Recipe Builder is an application that can be restored from installation files for the client side and an image for the server setup. | LOW |
|  | Given that local database synchronizes with the centralized database, only the centralized database needs to be backed periodically in accordance with the end user IT policies. | MEDIUM |
|  | For restored systems, network connectivity must be established with Batch Management and the central database for restoring the data. | MEDIUM |
|  | The application shall be capable of checking for current version and initiating automatic updates in online mode. | LOW |

## Documentation Requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Application Installation Manual | MEDIUM |
|  | Application Administration Procedure | MEDIUM |
|  | Instructions for Use with example of creating a recipe using the standard work flow | HIGH |
|  | The application will have the full set of GAMP5 Documents:   * Validation Plan * User Requirement Specification (this document) * Functional Requirement Specification * System Design specification for all software components of the application * Risk Assessment * SFAT and Procedure that adequately challenge every aspect of the application * Traceability Matrix * IQ/OQ and associated reports * Software Build Records * Software release notes * Code review report * design review report(s) * change control documents | HIGH |
|  | Source code with appropriate release notes, executables and any referenced dynamic linked libraries shall form of part of the turnover package to Pall from the vendor. | HIGH |
|  | Standard Operating Procedures for application Administration and Maintenance, Backup, Restore and Disaster Recovery and User Access Management shall be created. | MEDIUM |
|  | License agreements for all the tools used for development purposes and runtime purposes shall be adhered to. Use of open source is subject to license agreements and approval. | HIGH |
|  | Work Instructions shall be developed to allow for creating new Unit operation editor and link it to the Recipe Builder. | MEDIUM |
|  | Deployment package for application build – client and server side with instructions shall be provided. | HIGH |

## Regulations

### 21 CFR Part 11 / Annex 11 requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The application shall utilize a secure database. | HIGH |
|  | Application shall generate reports from data extracted out of the database – utilizing local database in offline mode and centralized database in online mode. | HIGH |
|  | Access to application database shall be restricted by role based user access. | HIGH |
|  | Audit trail (events) shall be stored in the database. | HIGH |
|  | The application shall have a means for archiving the data from the centralized database based on a configurable retention period. | HIGH |
|  | All software objects shall be validated. | HIGH |
|  | User privileges and their implementation of them in any user-facing application shall be validated. | HIGH |
|  | Electronic Signature shall be based on requiring multiple users required to approve a recipe prior to sending to Batch Management. | LOW |

### Data Integrity requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Records transmitted to Batch management shall allow for reconstruction of the activity, including the traceability to the user ID with date /time. | HIGH |
|  | Data and metadata shall be stored in the local application database until it is successfully transferred to a central database. | MEDIUM |
|  | Data transmission to Batch management after ensuring the rules are followed shall be identifiable and traceable. | HIGH |
|  | Data shall be permanently stored in the centralized database for duration of the retention period as determined by the end user | HIGH |

## Operational

### Servicing requirements

|  |  |  |
| --- | --- | --- |
|  | |  |
| Preventative Maintenance Periods | Procedure driven based on end user requirements | |
| Tear and Wear / Spare Parts Availability | On Stock | |
| 24/7 Service Requirements | No | |
| Standard Service Contract | Yes | |
| Downtime acceptance | Procedure driven based on end user requirements | |

### System flexibility

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | Recipe Builder shall be scalable in terms of adding up to 25-unit procedures with its own unit operations editor and rules. Adding a new unit procedure shall not impact / alter existing unit procedure editor. | Medium |
|  | Recipe Builder shall be capable of multi-language support –at a minimum – English, German, French, Spanish, Mandarin and Japanese.  (How is this planned to be implemented? Left justification? Reports?) | Low |
|  | Recipe Builder shall be capable of functioning in native operating system language. | LOW |

## Ownership

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The application shall be designed and owned by Pall’s Digital Technology and Solutions group. The deployment shall be owned and maintained by Technical operations (PASS). Pall Corporation shall own the Intellectual property and any related software derivative works including the documentation and copyrights. | HIGH |

## Product Privacy and Security policy requirements

|  |  |  |
| --- | --- | --- |
| **REFERENCE** | **DESCRIPTION** | **PRIORITY** |
|  | The software application shall adhere to „Danaher Product Privacy and Security By Design Policy. “ | MEDIUM |
|  | Given the application integrates with Wonderware Batch Management™ – the application shall be capable of functioning without access to the Pall business network and internet connectivity. | MEDIUM |
|  | Source code / platform components delivered used shall be verified utilizing Code scanning tools and required to pass penetration testing. | HIGH |
|  | The application shall be designed for being capable of being licensed from an end user perspective. | MEDIUM |

# Glossary

|  |  |
| --- | --- |
| Term | Definition |
| ANSI | American National Standards Institute |
| CFR | Code of Federal Regulations |
| CM | Control Module - The lowest grouping of equipment in the physical model that can perform basic control.  A control module typically defines the control for a field device (i.e., transmitter or control valve); however, a control module can also be a lower level module such as an alarm handler or a higher-level collection of sensors, actuators, other control modules, and associated processing equipment that, from the standpoint of control, is operated as a single entity (i.e. PID control loop). |
| EM | Equipment Module - Functional grouping of control modules that in the physical model perform basic control and require coordination of multiple field devices. An equipment module typically defines the control for multiple field devices (i.e. transfer equipment, inlet valve header). |
| FDA | Food and Drug Administration |
| FS | Functional Specification |
| FRS | Functional Requirement Specification |
| GAMP | Good Automated Manufacturing Practices |
| GMP | Good Manufacturing Practices |
| I/O | Inputs/Outputs |
| ISA | International Society of Automation |
| ISPE | International Society for Pharmaceutical Engineering |
| HMI | Human Machine Interface - A graphic display allowing user interaction with the process. |
| MHRA | Medicines and Healthcare Regulatory Agency UK |
| PLC | Programmable Logic Controller |
| PM | Phase Module |
| RP | Recipe Procedure |
| RPE | Recipe Procedural Element |
| RPH | Recipe Phase - S88 logic called upon to carry out a control action connected to the system |
| RUP | Recipe Unit Procedure - A strategy for carrying out a contiguous process within a unit. It consists of contiguous operations and the algorithm necessary for the initiation, organization, and control of those operations. |
| SDS | Software Design Specification |
| Unit procedure | A strategy for carrying out a contiguous process within a unit. It consists  of contiguous operations and the algorithm necessary for the initiation, organization, and control  of those operations. |
| URS | User Requirements Specification |

# Revision Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Issue | Date | By | Change | Page |
|  | 5 March 2021 |  | Initial Issue | N/A |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
| Image result for pall corporation | | **Visit us on the Web at www.pall.com** |
| **Contact us at www.pall.com/contact** |
|  | |  |
|  | **Corporate Headquarters**  Port Washington, NY, USA  +1.800.717.7255 toll free (USA)  +1.516.484.5400 phone  **European Headquarters**  Fribourg, Switzerland  +41 (0)26 350 53 00 phone  **Asia-Pacific Headquarters**  Singapore  +65 6389 6500 phone | **International Offices**  Pall Corporation has offices and plants throughout the world in locations such as: Argentina, Australia, Austria,  Belgium, Brazil, Canada, China, France, Germany, India, Indonesia, Ireland, Italy, Japan, Korea, Malaysia, Mexico,  the Netherlands, New Zealand, Norway, Poland, Puerto Rico, Russia, Singapore, South Africa, Spain, Sweden, Switzerland, Taiwan, Thailand, the United Kingdom, the United States, and Venezuela. Distributors in all major industrial areas of the world. To locate the Pall office or distributor nearest you, visit www. Pall.com/contact.  The information provided in this literature was reviewed for accuracy at the time of publication. Product data may  be subject to change without notice. For current information consult your local Pall distributor or contact Pall directly. |
|  |
|  |
| © 2020 Pall Corporation. The Pall logo, Pall are trademarks of Pall Corporation.  ® indicates a trademark registered in the USA and TM indicates a common law trademark.  Filtration. Separation. Solution is a service mark of Pall Corporation. |
|  |