Analytics & Controls, Research and Development

Software Design Specification

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

Document Number:	Recipe Builder Design Spec.DOCX
Revision Number:	01
Save Date:	8 November 2021
Author:	Tata Consultancy Services

Contents

1 Approval Log

	NAME	TITLE	SIGNATURE	DATE
DOCUMENT ORIGINATOR	Bindya Bysemane Ganesh	Project Manager [TCS]	DocuSigned by: Bindya Bysemane Ganesh 2E2176288BD4442...	08-Nov-21 | 06:35 PST
TECHNICAL APPROVAL	Umesh Rao	Project Manager	DocuSigned by: 1DCB4DB487DF49D...	11-Nov-21 | 05:37 PST
TECHNICAL APPROVAL	Jun Tao	Software Lead	DocuSigned by: Jun Tao 586CAE7438024E0...	08-Nov-21 | 07:00 PST

2 Introduction

Guidelines

The introduction should provide information on:

- *Ownership of the document: Who created the document, under what authority and for what purpose*
- *Contractual Status of the document [if applicable] like custom development and outsourcing*
- *Relationship to other documents like User Requirement Specification and Functional Specifications*

This document has been produced by TCS, to document the software design specifications for Recipe Builder (Trebuchet) Project, undertaken by the Pall Corporation. The software design specifications for the project Trebuchet are developed in accordance with the procedure/work instructions R&D AT QP006 “Automation Software Standards”.

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.

2.1 Related Documents

TITLE	DOCUMENT NUMBER	ISSUE
A-16532_Auto_FS 1v5 release	A-16532_Auto_FS (1v5)	20 July 2021
RecipeBuilder_FS	A-16528	
BatchManagement_StatelessAPI.chm	-	Nov 2018
SDS_V1_36_signed_EAG_Pall	A-16646_SUBM_Auto_SDS	19 July 2021

3 Overview

Guidelines

An overview should be provided, describing the configuration and/or design. The overview should not contain detailed design information.

3.1 Project Overview

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) is 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 – Wonderware® Batch Management™.

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

3.2 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 developed software application that:

- allow for users to create recipes and Unit Procedures – *online or offline* with synchronization capabilities to S88 based Batch systems – Wonderware® Batch Management™
- Graphical user Interface for defining phase parameters and the phase structure

3.3 Scope and Key Objectives

The scope of the document is to describe the design specification involved in implementing functionality of the Recipe Builder software tool. The main purpose of the Recipe Builder is to provide an intuitive and user-friendly graphical interface to model and parametrize the production workflows in a batch processing plant.

The resulting recipes must be converted and transferred to a Batch Management System, which is responsible for controlling the production processes.

3.4 Terminology

TERM	DEFINITION
Unit	A piece of equipment that processes or holds materials in a batch production plant; several units of same type might be available
Unit procedure	Defines a processing sequence of operations to be executed on the unit
Phase	The lowest level of processing step executed on the equipment; the execution of the phase can be customized through a fixed set of specific parameters
Operation	Logical set of phases
Train	Several physical units working together to manufacture a product
Recipe	Definition of the order of execution for the units defined in the train, to obtain the desired product
Batch Management System	Central control system for recipe-based batch processes, responsible for the manufacturing of a product according to a production order
Modular Bulk Fill	The Modular Bulk Fill Control System provides automated sequencing and equipment control of the Modular Bulk Fill skid activities. The Modular Bulk Fill supports Recipe control or manual control of the process. The recipes will be controlled by the batch engine

3.5 System Configuration

The Recipe Builder is implemented as a desktop app created using web technologies, it implements client-server architecture. Recipe Builder Server and recipe builder clients constitute the two main software modules.

The Recipe Builder application will be integrated into the existing production automation system of the plant. The different functions of the application are activated by the user, who interacts with the application by operating its graphical interface.

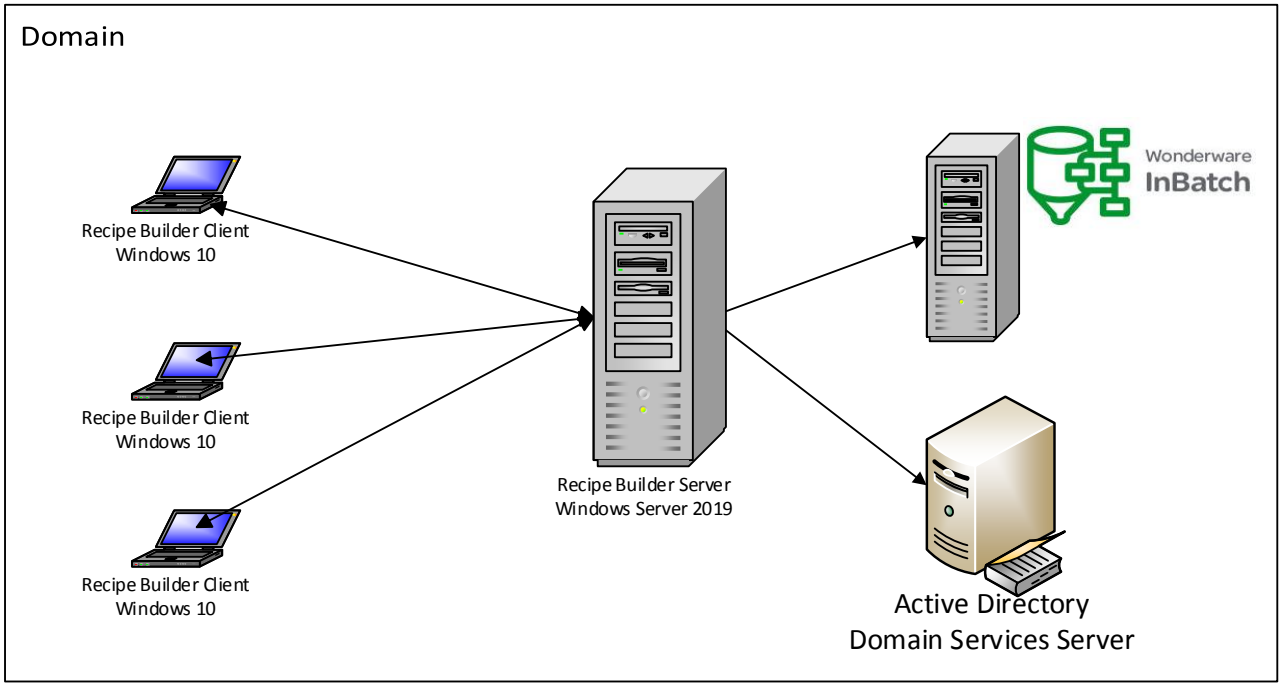


Figure 1 System Architecture

3.5.1 Recipe Builder Server

Recipe Builder server is supported in Windows Server 2019 operating system. It requires IIS and .net framework to be installed to host REST API services. It consists of components implemented using the.Net Web API technology

The Recipe Builder Server can handle several Recipe Builder Client instances. Client and Server communication is realized with Transport Layer Security (TLS), TLS is a cryptographic protocol, it provides end-to-end communications security over networks

Recipe Builder Server Application Technology Stack

- Framework: .Net Framework (version 4.8)
- IDE: Visual Studio Professional 2019
- Version Control: Git
- Web Infrastructure: ASP.Net Web API
- Programming language: C#
- Debug Logging: .NET logging with Seri Log extension
- ORM: Entity Framework 6
- Installer: WIX 3.11
- Database: PostgreSQL 13

3.5.1.1 System Requirement

Operating System	Windows Server 2019
.Net Framework	4.8
IIS services	enabled
Domain Services and Active Directory	Configured

3.5.1.2 Development Tools

- Visual Studio 2019
- .Net framework 4.8
- WiX Tool set 3.11
- AWS code commit
- Postgres SQL 13
- Git 2.31
- Source tree 3.4.2
- IIS (Internet Information Services)

3.5.2 Recipe Builder Client

Recipe Builder client is an electron app. It is installed in Windows 10.

The Recipe Builder Client will be able to work offline with limited functionality, without the Recipe Builder Server. The frontend consists of screens guiding the user through the process of building a procedure to run on the equipment available on the plant. Several equipment units that must run together to manufacture a recipe are configured as a train. The user builds the recipe by assigning the unit procedures that have to be executed on each physical unit from the train. The access to the different screens and the actions triggered through the user interaction is allowed only for authorized users with the necessary rights

Recipe Builder Client Application Technology Stack

- Framework: Electron with Angular (version 11)
- Database: PostgreSQL 13
- Client backend: Node JS 14.6
- IDE: Visual Studio Code
- UI components: Angular Material
- Programming language: Typescript 4.1.2
- Installer: WIX 3.11

3.5.2.1 System Requirement

Operating System	Windows 10

3.5.2.2 Development Tools

- Visual Code 1.59
- Angular 11
- Electron Framework 13.1.7
- WiX Tool set 3.11
- Visual Studio 2019
- Node JS 14.6
- Angular material
- Git 2.31
- Source tree 3.4.2
- PostgreSQL 13

3.5.3 Batch management Server

The only interface to other systems is with the Batch Management System.

Batch Management System. Is Central control system for recipe-based batch processes, responsible for the manufacturing of a product according to a production order

the Recipe Builder Server interacts with the Batch Management System. The recipes uploaded to the Wonderware Batch Management System are compliant with the BSMML (XML) format, defined in the BatchML V0401 specification (https://services.mesa.org/ResourceLibrary/ShowResource/daa0f443-f958-4e46-b6a2-76223207f7b7)

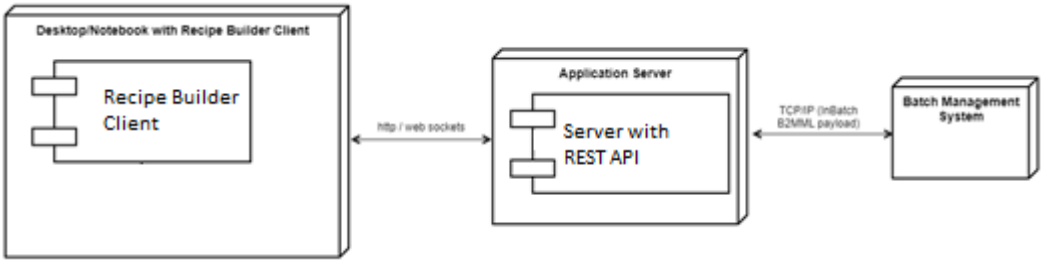
The Wonderware Batch Management™ stateless APIs library will be used for instance to interact with the Wonderware Batch Management System.

4 Description

Guidelines

The high-level description should be broken down to the level of the individual software modules, briefly stating the purpose of each. This should describe the functions of each module and interfaces between the modules as well as the interfaces to the external system[if applicable].
A system diagram is recommended.

4.1 Software Modules



There are 2 main Modules

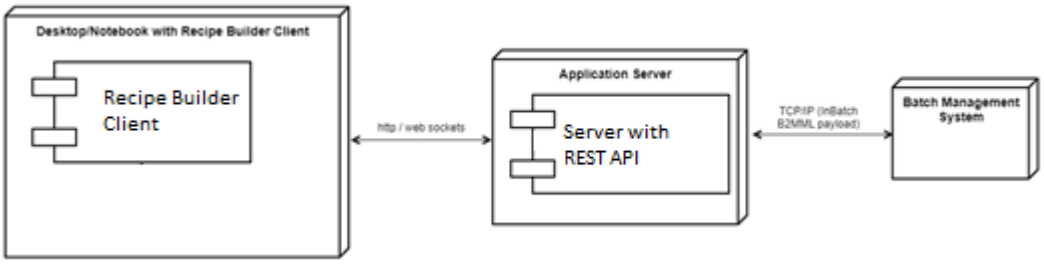
1. Recipe Builder Client
2. Recipe Builder Server

The Recipe Builder Client instances do not have a direct connection to the Batch Management System. They can interact with the Batch Management System (for the upload of a recipe) only over the Recipe Builder Server, in online mode.

4.2 Interface

There are 3 main Interfaces

1. Recipe Builder Client UI interface
2. Recipe Builder Server REST API interface
3. Batch management System's batch API interface



4.2.1 Recipe Builder Client UI Interface

Recipe Builder Client offers the graphical interfaces that lead the user through the selection and parametrization of the involved physical units as well as the product workflow to obtain different recipes. For each type of physical Unit, a corresponding unit editor is available.

The frontend part is handling all the user interaction and is the only way that a user has to access the functionality of the Recipe Builder.

The Recipe Builder Client is built modularly, with screens available in the frontend for each functionality:

- Unit Editors
- Dashboard
- Unit Procedure tab Screen
- Recipe Audit Screen
- Audit Trail System Log Screen
- User Audit trails
- Manage users
- Account Page
- About Screen

1.1.1 Recipe Builder API Interface

Recipe Builder Server API (application programming interface) are set of rules that define how applications or devices can connect to and communicate with each other. Recipe Builder uses REST API representational state transfer architectural style.

REST APIs communicate via HTTP requests to perform standard database functions like creating, reading, updating, and deleting records (also known as CRUD) within a resource to access and use data. This data can be used to GET, PUT, POST and DELETE data types, which refers to the reading, updating, creating, and deleting of operations concerning resources. REST APIs are stateless, meaning that each request needs to include all the information necessary for processing it

Request headers and parameters are also important in REST API calls because they include important identifier information such as metadata, authorizations, uniform resource identifiers (URIs), caching, cookies and more. Request headers and response headers, along with conventional HTTP status codes, are used within REST APIs.

Controllers are responsible for responding to requests made against an ASP.NET MVC website. Each browser request is mapped to a particular controller. It exposes methods that will be called when the API receives requests through the route

Below are the Recipe Builders Controller and API governed by its implemented http routing

4.2.1.1 Account Controller

Account controller is responsible for the actions specific to login , log out and managing users in the recipe Builder application

ACTION	URI	DESCRIPTION
POST	/api/AddNewUser	Add a new user by administrator.
POST	/api/ChangeUserRole	Change the user role for a particular user by administrator.
DELETE	/api/DeleteUser	Perform delete user by administrator.
GET	/api/GetAllUser	Get all the available usernames.
Get	/api/GetUserData	Get user details like, role , first name and last name etc.
POST	/api/LoginUser	Allow user to login.
POST	/api/Logout	Capture user logged out event.

4.2.1.2 Audit Trail Controller

This controller implements API to fetch audit trail of recipe , system error log and user audit trails .It also filters data as per the Request criteria

ACTION	URI	DESCRIPTION
GET	/api/GetFilteredUserAuditTrail	Get filtered Audit events for specific users.
GET	/api/GetFilteredUsersAuditTrail	Get filtered Audit events for all the users.
GET	/api/GetRecipeAuditTrail	Get Audit events of a recipe.
GET	/api/GetSystemErrorLog	Get System error Log.
GET	/api/GetUserAuditTrail	Get Audit events of a user.
DELETE	/api/PurgeSystemErrorLog	Delete Old System Error Log.

4.2.1.3 Batch Recipe Controller

This control implements API related to conversion of B2MMl xml and also API to release recipe to batch management.

ACTION	URI	DESCRIPTION
POST	/api/ExportRecipeAsXML	Export recipe as a b2mml file.
POST	/api/ReleaseRecipe	Release the recipe for testing or production.

4.2.1.4 General

This controller implements API related to getting the server status and server version

ACTION	URI	DESCRIPTION
GET	/api/GetServerStatus	Get if the server call is successful.
GET	/api/GetServerVersion	Server version is fetched successful.

4.2.1.5 Offline Synch

This controller specified the API used for synchronizing client error log and also the online offline event saved in Recipe Builder client.

ACTION	URI	DESCRIPTION
POST	/api/SynchClientErrorLog	Log all the client error logs generated.
POST	/api/SynchOfflineOnlinetEror Events	Log all the online / offline events to audit logs.

4.2.1.6 Operation

Controller specific to operations workflow

ACTION	URI	DESCRIPTION
POST	/api/AddOperationAsTemplate	Save operation as a template.
GET	/api/GetAllOperationTemplate	Fetch the operations which are made as templates.
GET	/api/GetOperationTemplate	Fetch the operation template.

4.2.1.7 Phase

Controller specific to phase operations

ACTION	URI	DESCRIPTION
Get	GET /api/GetAllPhaseTemplate	Fetch all the custom phase which are made as template corresponding to unit procedure.
GET	GET /api/GetPhaseTemplate	Fetch phase which are made as template.
POST	POST /api/SavePhaseAsTemplate	Save custom phase as a template.

4.2.1.8 Recipe State

This controller helps to manage recipe State APIs

ACTION	URI	DESCRIPTION
POST	/api/CheckInRecipe	Perform Check-in recipe.
POST	/api/CheckoutSubmittedRecip e	Chekout the recipe while reviewing.
POST	/api/IgnoreChangesAndCheck In	Perform Check-in without updating the recipe.
POST	/api/RejectRecipeForApproval	Reject the recipe approval while reviewing.

POST	/api/SubmitRecipeForProduction	Submit recipe for approval.
POST	/api/SubmitRecipeForTest	Submit recipe for approval.
POST	/api/UndoCheckout	Perform Undo Checkout and discard the recent updates.

4.2.1.9 Unit Procedure

Controller to manage the recipe creation, fetch recipe details etc.

ACTION	URI	DESCRIPTION
POST	/api/CreateRecipeFromTemplate	Create recipe from template.
POST	/api/CreateRecipe	Create new recipe.
POST	/api/DuplicateRecipe	Duplicate and create new recipe.
GET	/api/GetAllRecipeListView	Get all the recipes.
GET	/api/GetCheckOutRecipe	Get details of checked out recipe.
GET	/api/GetConfiguredUnitProcedure	Checks if Unit procedure is configured.
GET	/api/GetFilteredRecipes	Get all the recipes based on the filter criteria.
GET	/api/GetRecentUnitProcedureRecipe	Gets the recently updated unit procedure.
GET	/api/GetRecipeDetails	Get recipe details
POST	/api/RecipeAsTemplate	Save a recipe as template.
POST	/api/RenameRecipe	Rename recipe.

4.2.2 Batch System Interface

The Wonderware Batch Management™ stateless APIs library will be used for instance to interact with the Wonderware Batch Management System. Below API are used Recipe Builder

#	INBATCH API	DESCRIPTION
1	`ImportRecipeFromXml`	Imports a recipe into the Recipe Database from an B2MML Xml string.
2	`ValidateRecipe`	Validate a specified recipe with the Recipe Database of Batch management system and returns true or false
3	`SynchronizeRecipe`	Performs a synchronize of the recipe against the InBatch Process Model.
4	`ApproveRecipeForTest`	Approves the specified recipe for test.

5	`ApproveRecipeForProduction`	Approves the specified recipe for production.

5 Software Module Description

Guidelines

A detailed description of each software module should be provided including the following:
• Module operation: the description may take the form of pseudo code or a flow chart
• Interfaces to other modules: these may refer to the system diagram, if one is produced.
• Error handling and data checking
• Data mapping to each module
• Software module data

For each sub-program in the software module, the following should be described:
• sub-program Operation
• the steps involved in each process to be performed and the inputs to and outputs from each step
• parameters: each parameter should be identified as either input parameter, output parameter or input & output parameter. The parameters should also be identified as either pass by value or pass by reference.
• algorithms
• language, including version as well as reference to any programming standards
• HMI screens
• sub-program data
• report generation

5.1 Recipe Builder Client

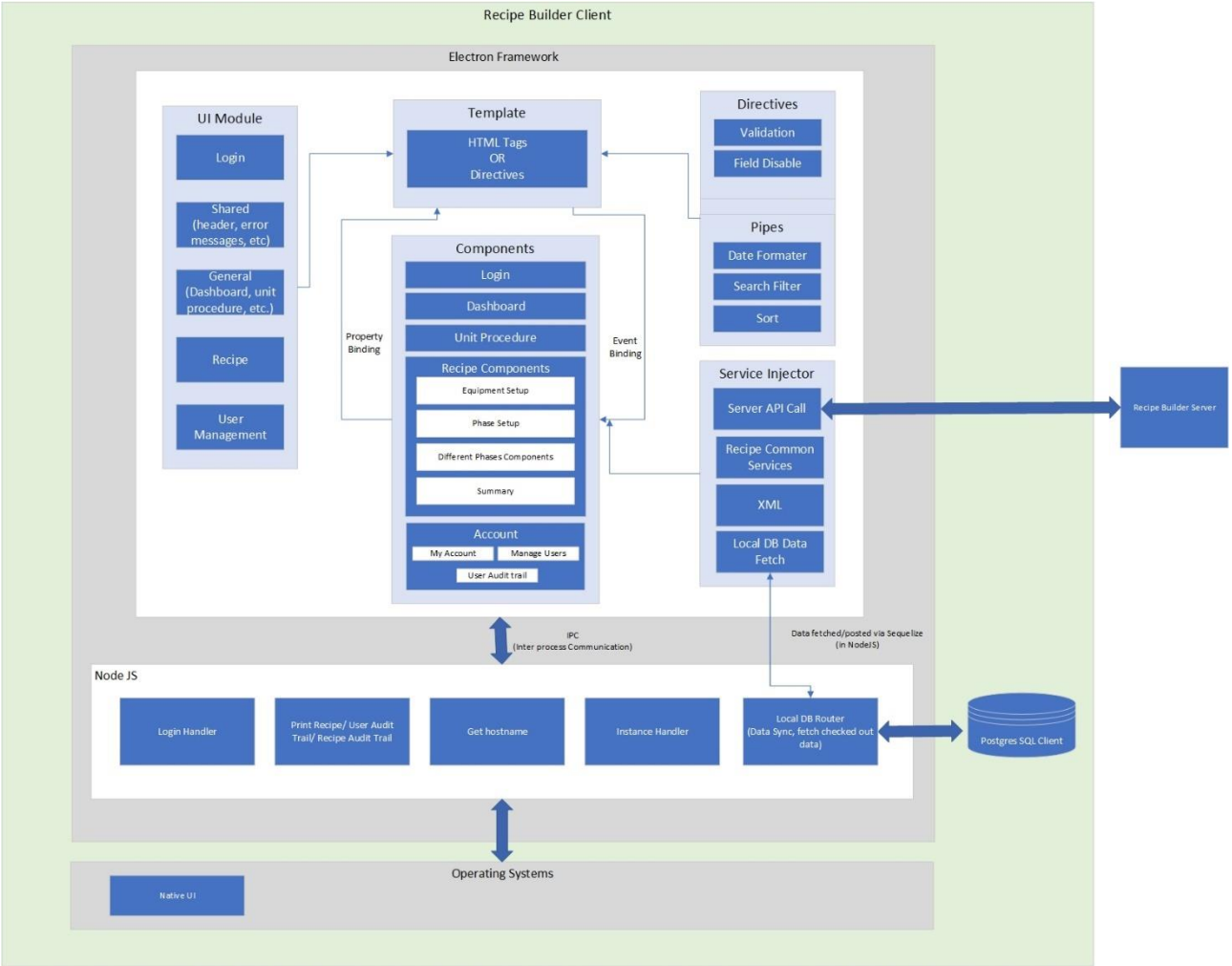
The Recipe Builder Client allows the configuration of the technical processes on a plant, necessary to manufacture the planned products. The frontend part offers the graphical interfaces that lead the user through the selection and parametrization of the involved physical units as well as the product workflow to obtain different recipes. For each type of physical unit a corresponding unit editor is available

Electron framework uses chromium inbuilt browser for displaying the web content and node JS for working with the local filesystem and operating system. Electron has two types of processes: Main and Renderer. Asynchronously communication between theses process is possible via inter process communication (IPC) modules

The frontend part is handling all the user interaction and is the only way that a user must access the functionality of the Recipe Builder.

The Recipe Builder Client instances do not have a direct connection to the Batch Management System. They can interact with the Batch Management System (for the upload of a recipe) only over the Recipe Builder Server, in online mode.

The Recipe Builder Client support stand-alone offline mode if the central Recipe Builder Server component is not available. The offline mode allows only user interactions regarding the recipe editing. No rules modification is allowed in local mode.



5.1.1 Main Process (Nodejs) Module

The communication with the native OS happens via Main process. Main process code is written in Nodejs. Here saving file to the file system, printing recipes are done in Main process via NodeJS. Communication with the local database happens via a middleware ExpressJS which will respond to HTTP Requests. This will listen to the port which is dynamically generated on start of the application. Generated port will be the any available port which will be assigned to application. To work with relational database sequelize entity framework is used..

5.1.2 UI Modules

UI modules are separated on basis of the functionality and reusability of the components. Below are the Modules list and the components related to that module.

5.1.2.1 Login Module

#	COMPONENTS	DESCRIPTION
1	Login	Used in login page

5.1.2.2 Shared Module

#	COMPONENTS/MODULES	DESCRIPTION
1	Header	Header or navbar of the application
2	Doughnut-Chart	
3	Success-banner	This component is used to display generic success message as banner
4	About-dialog	This component is used to display about section. Here in this page server version, Client version will be display.
5	Angular-Material-Module	This module has reference to all Angular material components.

5.1.2.3 General Module

#	COMPONENTS/MODULES	DESCRIPTION
1	Dashboard	Here in this component Recent 6 recipes are displayed along with count of recipes that has been updated since last login.
2	Recipe-Landing (Unit procedure)	He all the recipes are displayed (All UOE Type recipes)
3	Recipe-Preview	This component is used to display the preview of recipe. Here version history, operation, phases and parameter related to phases are displayed.
4	Rename-Recipe	This component is used as dialog (Pop-up) for renaming of recipes
5	Promote to template	This component is used as confirmation window for promoting recipe as template.
6	Print Recipe	This component is used to print recipe. This contains all the data related to recipes, including operation phase and phase parameters
7	Custom recipe (Create Recipe)	This component is used as dialog to create custom recipe.

5.1.2.4 Recipe Module

#	COMPONENTS/MODULES	DESCRIPTION
1	equipment-setup-configuration	This component is used to display equipment setup phases.
2	phase-setup-configuration	This component is used to add operation and phases to the recipe.
3	add-operation	This component is used in phase-setup-configuration to add operation to the recipe
4	common-equipment-flow	This is the generic component used in all phases to display flow diagram

5	generic-phase-display	This is the generic component used for the detailed view of phases
6	custom-phase	This component is used to display custom phase in detailed view
7	manifold-section(manifold-phase)	This component is used to display manifold phase.
8	operation-menu-dialog	This component is used as dialog for rename of operation, save operation as template and delete operation
9	phase-edit-dialog	This is the generic component used to edit name and description of phases
10	phase-menu-dialog	This component is used as dialog for save phase as template, delete phase.
11	phase-setup-add-phase-dialog	This component is used to add phase to the
12	Print	To print Bill of materials in summary page
13	recipe-audit-trail	This is component is used as dialog to view recipe audit trail in summary page
14	recipe-audit-print	This component is used to print recipe audit trail.
15	recipe-audit-filter	This component is used as custom filter for recipe audit trail
16	recipe-notes	This component is used to add and display recipe notes.
17	side-navbar	Side navbar component is used for navigation inside recipe. Here we can navigate to landing page or to any of the phases or to the summary page.
18	submit-release-dialog	This component is used as dialog for submit recipe in summary page.
19	summary	This is used to display the detailed summary of the recipe. this component will be shown on navigating to summary page.
20	transition-custom	This component is used for custom transition. This is used in custom phase page as a dialog for custom transition.

5.1.2.5 User Management Module

#	COMPONENTS/MODULES	DESCRIPTION
1	my-account	This component is used for my account page
2	manage-users	This component will be shown on redirecting to manage user's page. This will be shown only for the admin.
3	delete-user-dialog	This is used as confirmation dialog to delete user.
4	print-user-audit-trail	This component is used to print audit trail.
5	user-audit-trail	This page is used to display system audit tail

6	custom-filter-dialog	This component is used as filter for my account page
7	audit-filter	This Component is used as filter for user audit trail and system audit trail page
8	purge-dialog	This component is used to purge data dialog
9	error-log-dialog	This component is used for exporting of error log to csv file

5.1.3 UI Services

UI Services are commonly used or module wise used functions across the application. This contains the common API call function, calculations and validation used by the application.

#	SERVICES	DESCRIPTION
1	common Service	This has common function used across the application eg: Access check, Get state name, Loader, Export to xml etc
2	main-api Service	This service contains common API function call used by the entire application. Here authorization and headers for the API call is handled
3	addUser Service	This service contains all the api call functions related to User management components.
4	banner Service	This is used to set success or error message in the application
5	create-recipe Service	This contains the common function like Create recipe, Create recipe from template, Promote recipe as template.
6	header Service	This contains function call to get unit procedure from local database
7	local-api Service	This service contains the common function to connect to the local database and to get/post data.
8	login Service	This service has login/logout API call and auto login functionality.
9	operations Service	This contains the function related to operation like add operation, get operation,
10	phase-name Service	This service contains the list of phase name and the Routing URL for the phases.
11	phase-params Service	This service contains the functionality to get and set phase parameters, save phase as template functions
12	print Service	This contains the common function for printing of recipe across the application.
13	recent-unit-procedures Service	This service contains call to get recent unit procedure and check out recipe
14	recipe-check-in Service	This service has a function call for checkin recipe, reject recipe and checkout submitted recipe.

15	recipe-list-view Service	This contains get recipe based on Unit procedure and based on filters
16	rename-recipe Service	This contains renaming of recipe function across the application.
17	server-status Service	This contains the function to check if the application status in online or offline.
18	server-version Service	This service contains the function to get the server version
19	Summary Service	This contains the API call for release recipe, submit recipe for test, ignore changes and check in, submit recipe for production, Undo checkout, get recipe audit trail.
20	Upload Log Service	This contains service to upload offline/ online event and error logs
21	XML Generator Service	This service is used to generate recipe XML.

5.1.4 UI Pipes

Pipes are simple functions to use in template expressions to accept an input value and return a transformed value. Pipes are useful because you can use them throughout application, while only declaring each pipe once. For example, you would use a pipe to show a date as April 15, 1988 rather than the raw string format.

Below are the pipes used in the application

#	PIPES	DESCRIPTION
1	Date Formatter Pipe	This pipe is used across the application to format date in **MMM D, YYYY [at] hh:mm A**
2	Filter Valves Pipe	This pipe is used for searching of valves in the custom phase
3	Sort Pipe	This pipe is used for sorting of the table and data across the application.

5.1.5 Directives

Directives are classes that add additional behaviour to elements in applications. There are two directives used in the application.

#	DIRECTIVES	DESCRIPTION
1	Custom Validation Directive	This contains the common validation for the form field in the application
2	Disable Fields Directive	This will disable the field based on the condition.

5.2 Recipe Builder Server

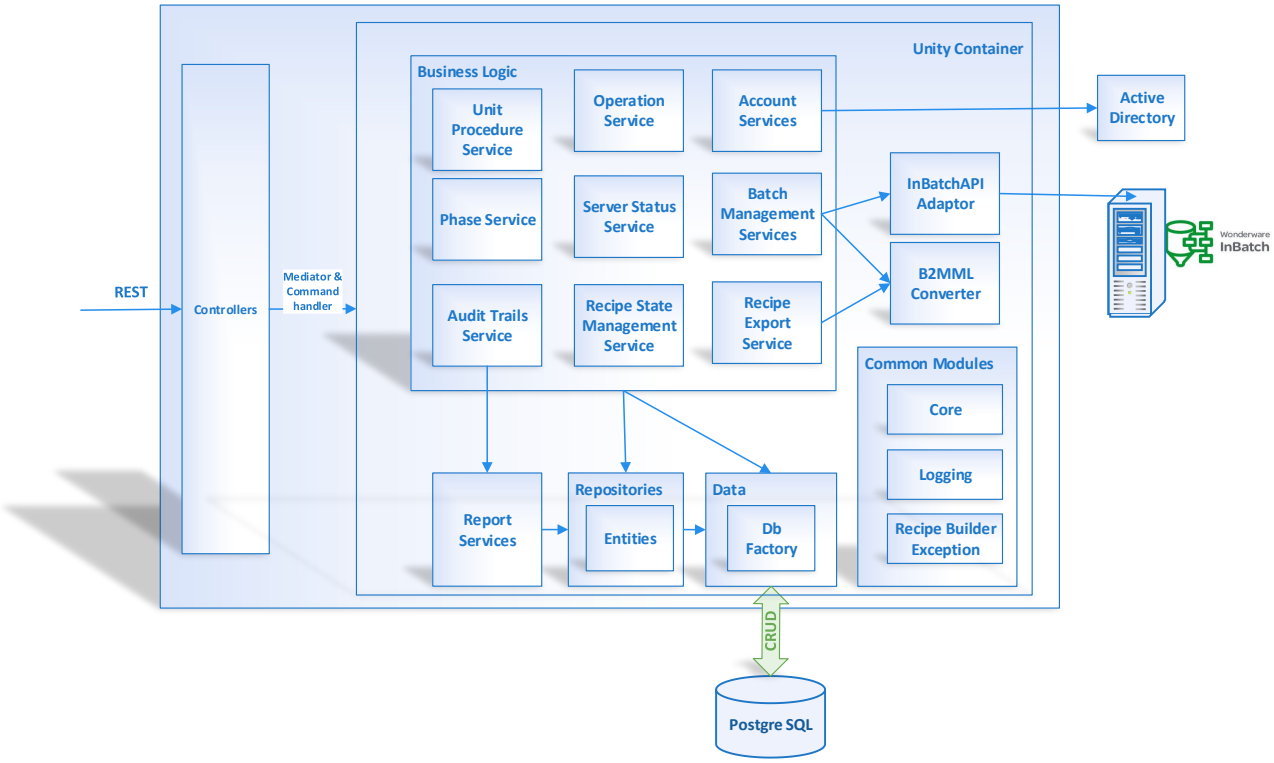


Figure 2 Recipe Builder Server Modules

The Recipe Builder Server implements the backend services handling the functionality required by the different application contexts.

The Recipe Builder Server manages the central resources (infrastructure, storage, external communication,) needed by the Recipe Builder Client instances to create or edit recipes. It is built modularly, containing different components responsible for the access to every resource type. Each of these components contains the detailed information about the structure of the data repository for the managed resource and about how to access it. The components have well-defined interfaces that allow to fetch or modify the data from the repositories.

The Recipe Builder Server backend services responsible for the different application contexts use internally these components to perform the actions triggered by the recipe Builder Client through the call of Rest APIs. This allows the control and protection of the access to the server resources, as each Rest API can be executed only if the user has a role with the necessary permission rights.

The Recipe Build Server handles the logic to solve eventual conflicts which might occur if some Recipe Builder Clients performed changes on the data repository while in local mode. Basically, the server will increment the version of a recipe after each recipe update received from a client.

The Recipe Builder Server is responsible for the storage of the recipes configured in the Recipe Builder Clients into the central data repository. An authorized user can validate and approve single recipes from the central repository. The approved recipes can be loaded into the Batch Management System.

Another functionality handled by the server is the storage of the phases and their parameters, defined for each unit type in the control system.

Different events documenting the handling of recipes are stored by the Recipe Builder Server for the audit trail.

Moules in recipe builders are

5.2.1 Controllers

Controllers are responsible for responding to requests made against an ASP.NET MVC website. Each browser request is mapped to a particular controller. It exposes methods that will be called when the API receives requests through the route

5.2.2 Services

This module contains the Business Logic layer and handles all the business rules, calculations, and actual logic within your application it will use data models retrieved from data-access layer.

TABLE	DESCRIPTION
Operation Services	Contains logic for recipe operations related functionality
Account Service	Contains logic for login, logout, and user management related functionality
Phase	Contains logic for recipe phase related functionality
Server Status	Sends server status as true
Recipe state Management services	Handles business logic for recipe workflow and logs recipe audit log
Batch management Services	Handles business logic for releasing recipe and any communication with batch service.
Audit Trail services	Logs audit log , synchronizes audit log from client

5.2.3 Report Services

This module is responsible for logging and generating system error log and audit log reports

5.2.4 Repositories

Repository is an object that is meant to fetch and save entities from/to storage

5.2.5 Entities

Entity Framework is object-database mapper. Entity is an object representing (usually) a row in a database. Entity Framework 6 is used to connect to PostgreSQL database and to create entities class. Every database table is mapped to one entities class. Refer 5.1.2 section for database tables

5.2.6 Recipe Builder Data

This module is responsible for connecting to the PostgreSQL database and responsible for saving and updating the “RecipeBuilderDB” database

5.2.7 InBatchAPI Adaptor

This module provides interface to interact with the batch management through InBatch API

5.2.8 B2MML Converter

This module is responsible to covert Recipe .net object to Batch XML. The batch xml is in format of B2MML-V0401. Customized version of the of B2MML-V0401 schema obtained from Wonder Ware, theses schema are serialized to .net objects. the recipe data is populated to created recipe batch XML

5.2.9 Common Modules

These modules implement the common functionality which are used by all other modules

5.2.9.1 Core

This module implements string constants, enumerated values, and Helper functions

5.2.9.2 Logging

This module creates instance application logging. Serilog library provides diagnostic logging to files. Recipe Builder has several levels of log events, Debug, Information, Error and Fatal. The logging level can be configured in the server configuration file

5.2.9.3 Recipe Builder Exception

This is a custom exception created to handle all known errors. All the error caught by recipe builder server is constructed into error codes. Error code is sent to client to display appropriate message on the UI interface

Error code	Error Message
1000	Recipe name already exists
1001	Unknown error
1002	Error creating recipe
1003	Error rename Recipe
1004	Error adding operations to Recipe
1005	Error rename operation
1006	Error Deleting operation
1007	operations name is empty.
1008	Error Moving operation sequence in recipe
1009	Operation Name Already Exists
1010	User role is empty
1011	Username already Exists
1012	User must be Administrator
1013	User does not present in Active Directory
1014	Incorrect password
1015	Error adding user
1016	Error getting checked-out Recipe
1017	Error exporting Recipe
1018	User not present
1019	Error duplicating recipe
1020	Checked-out recipe cannot be checked Out Again
1021	Error Checking in Recipe
1022	Checked-out Recipe can be checked-in
1023	Error getting Recipe
1024	Error getting all recipes
1025	Error getting recently updated recipe
1026	Error archiving Recipe
1027	Checked-out Recipe cannot be released
1028	Error connecting to Batch Management Server
1029	Error doing Recipe Release
1030	Unauthorized access
1031	Error adding recipe note
1032	Empty recipes note

1033	Error getting user Audit Trail
1034	User Does Not Exists
1035	Error Recipe as template
1036	Error already promoted
1037	Error changing user role
1038	User must be Administrator to change user role
1039	User cannot change their role
1040	Error adding operation as template
1041	Operation is already a template
1042	Operation does not exist
1043	Error getting All Operation template
1044	Error recipe does not exist
1045	Error creating Recipe from template
1046	Error in Submitting Recipe for Test
1047	Error in Submitting Recipe for Production
1048	Error getting Username
1049	Operation template name already exists
1050	Recipe cannot be rejected if not submitted for approval
1051	User who created recipe cannot release recipe
1052	Error releasing Recipe to Batch Management
1053	Recipe cannot be Released If Not Submitted for Approval
1054	Recipe should be submitted For Test or Production
1055	Error In checking-out submitted Recipe
1056	Error In rejecting Recipe
1057	Released for production Recipe cannot be checked-out
1058	Error getting operation template
1059	Error operation template does not exist
1060	Error getting Recipe Audit Trail
1061	Error max Audit Log Limit reached
1062	Error fetching System Error Log
1063	Error getting Recipes
1064	Error max recipe limit
1065	Phase template name already exists
1066	Error saving phase as template
1067	Error getting all Phase template
1068	Error getting phase template
1069	Error phase template does not exist
1070	Error getting all user Audit Trail
1071	Error synchronizing recipe in Batch Management server
1072	Error user permission for Batch Management server
1073	Error undo checkout
1074	Error getting number of Recipes updated

5.2.10 Interface with modules

The module of the server receives HTTP request from the client. Each client request is mapped to a particular controller.

Command and query responsibility segregation is used to separate read and write operation to database.

The separation occurs based upon whether the methods are a command or a query. This pattern can be applied in other cases whereby the API is split into two separate microservices, one for commands and the other for reads.

A mediator object is used where other command and query objects communicate with it rather than each other. the Mediator recognizes the "Request" and delegates it to the respective "Handler" that returns the data accordingly.

Command and query handler interacts with the services class. Service class has necessary business logic to process HTTP requests.

DTO are used as a container to encapsulate request data and pass it from one layer of the application to another and return data back to the controller.

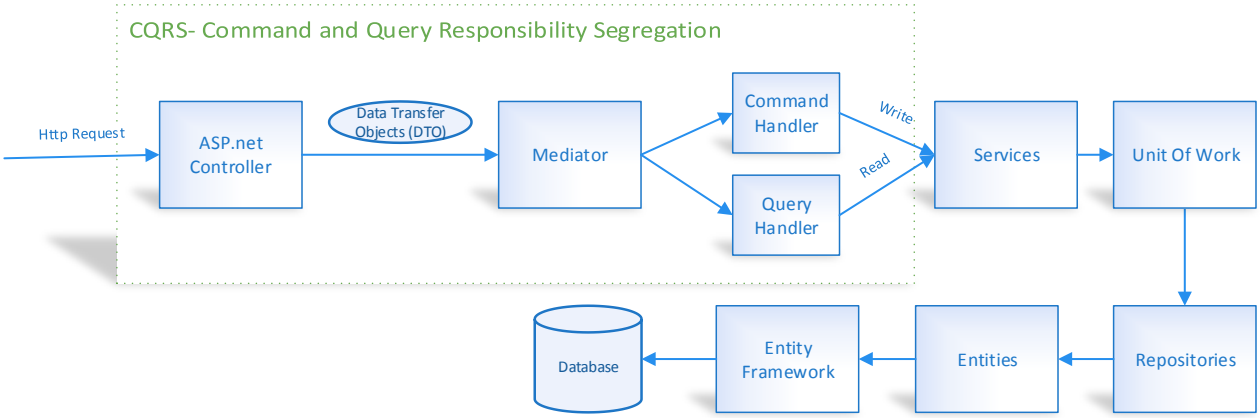


Figure 3 Operation Flow Direction

6 Data

Guidelines	*System data and the major data objects should be defined. The data should be characterized in a hierarchical manner with complex objects being built up of simpler objects. A description of the data objects should be provided including the data types, data format, data precision and data accuracy.*

6.1 Database Design

Recipe builder uses PostgreSQL as database engine. There are two databases created

6.1.1 Recipe Builder Client Database

Client database is used to store the data of checked out recipes. It is also storing necessary data required to edit a recipe is in checkout state.

6.1.2 Recipe Builder Server Database

Server database stores all the information related to recipe creation and managing the recipe workflow states.

Below is the database design for both server and client database

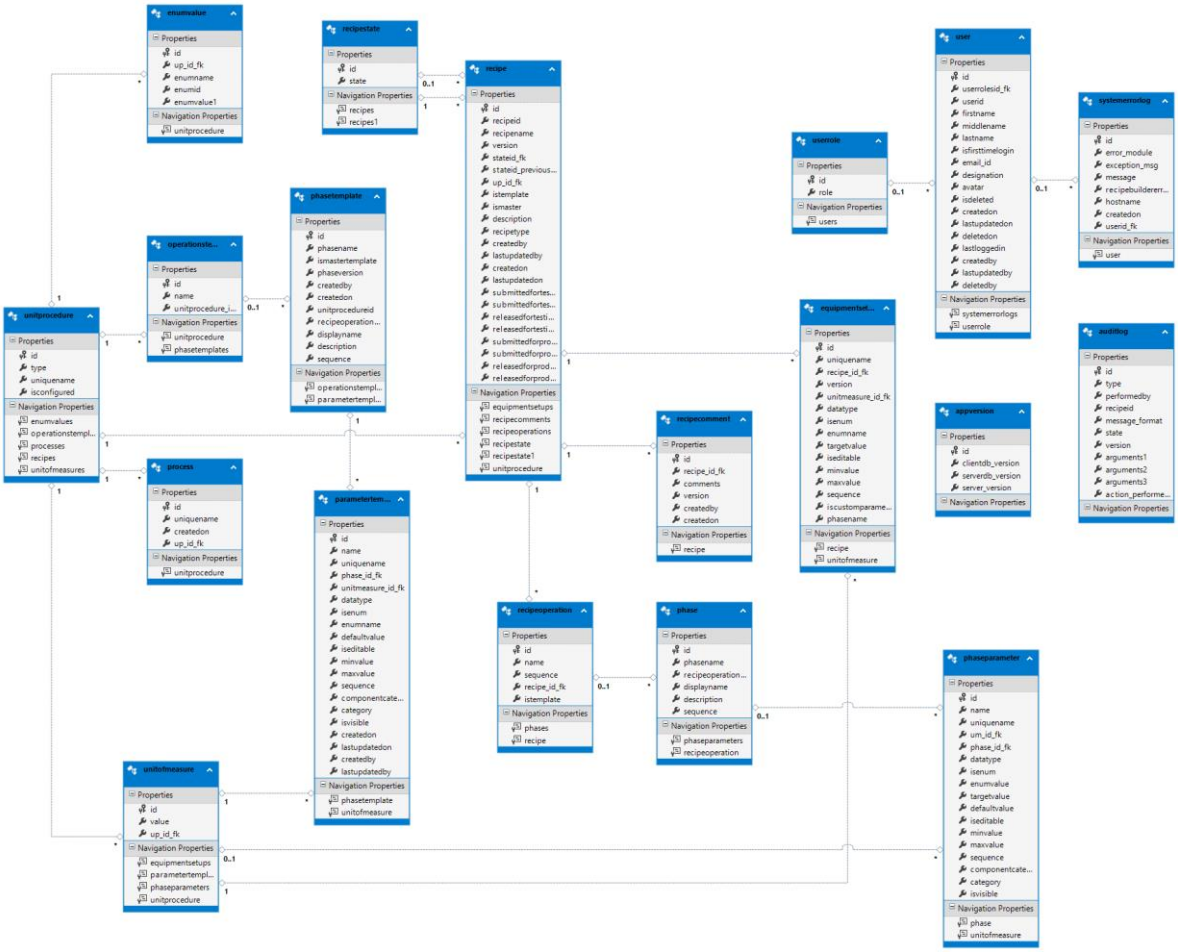


Figure 4 Database Design

#	TABLE	DESCRIPTION
1	UnitProcedure	Stores information of the configured unit operation equipments
2	Process	Stores process instance name of the each of the unit procedure
3	PhaseTemplate	Stores the phase details from master data
4	OperationTemplate	Stores the operations which were made as template
5	ParameterTemplate	Stores the parameters from master data
6	EnumValue	Enumerations values available for the parameters are stored here
7	RecipeState	Different states of the recipes are specified in this table
8	recipe	Recipes are stored in this table

9	Recipe operation	Operations related to recipe are stored in this table
10	Phase	Phases added to the recipes are stored in this table
11	Recipe Comments	Phases added to the recipes are stored in this table
12	EquipmentSetupParameter	Equipment set up parameters of the recipe stored in this table
13	Phaseparameter	parameter added to the recipes are stored in this table
14	appVersion	Database versions and supported server and client version stored in this table
15	Audit Log	Recipe and user audit logs are stored here
16	User	Users added to the applications are stored here
17	userRole	User roles available in the recipe builder are stored
18	systemerrorlog	System errors are stored in this table

6.2 Input Data

Input for creating recipe template tables is master data XML. The xml is exported from the Wonder ware batch systems Unit Export feature of Batch Automation System. This explains contains details of the enumeration and enumeration type used , units used within the unit operation , process instance details and phases and its parameters details .

The master data is extracted from unit export xml and the entry is made in the recipe templates table of recipe builder database. Below mapping is used to extract data

XML TAG	DESCRIPTION
EnumSets	Holds Enums and Enum values used in the recipe
ENUMVALUES\Name	Name of Enum value
ENUMVALUES\ EnumValue	Enums items belonging to Enum values
Process\Phases\Name	Holds process instance name
PROCESSPHASES	This xml element holds the phase collection and it detail
PROCESSPHASES\Phase	Recipe phase details
PROCESSPHASES\Phase\Name	Name of the phase
PROCESSPHASES\Phase\ PHASEPARAMS\parameter	Parameter details of a phase
PROCESSPHASES\Phase\ PHASEPARAMS\parameter\Name	Unique name of the parameter

PROCESSPHASES\Phase\ PHASEPARAMS\parameter\ PARAMPROCVARS\ ProcVarExt\ UnitMeasureId	Units of the parameter
PROCESSPHASES\Phase\ PHASEPARAMS\parameter\ PARAMPROCVARS\ ProcVarExt\ TargetDfltVal	Default value of the parameter
PROCESSPHASES\Phase\ PHASEPARAMS\parameter\ PARAMPROCVARS\ ProcVarExt\ HighLimitDfltVal	Max value of the parameter
PROCESSPHASES\Phase\ PHASEPARAMS\parameter\ PARAMPROCVARS\ ProcVarExt\ LowDevDfltVal	Min value of the parameter
PROCESSPHASES\Phase\ PHASEPARAMS\parameter\ PARAMPROCVARS\ ProcVarExt\ TargetDataClass \	Data type of the parameter

Version of UOE supported.

UOE	VERSION	REFERENCE
Bulk Fill	MBF_V1_06	A-16532_Auto_FS 1v5 release, A-16646_SUBM_Auto_SDS

6.3 Validation Data

NAME	NUMBER OF CHARACTER	VALID CHARACTER
Description	120	Alphanumeric
Material ID	16	Alphanumeric
Material Name	40	Alphanumeric
Recipe ID	16	Alphanumeric
Recipe Name	128	Alphanumeric
Operation Name	16	Alphanumeric

6.4 Data type Mapping

Data type mapping of the parameters from Batch engine to recipe builder application

VALUE	DATA CLASS	DATA TYPE
0	Analog	Float
1	Discrete	Bool
2	String	String

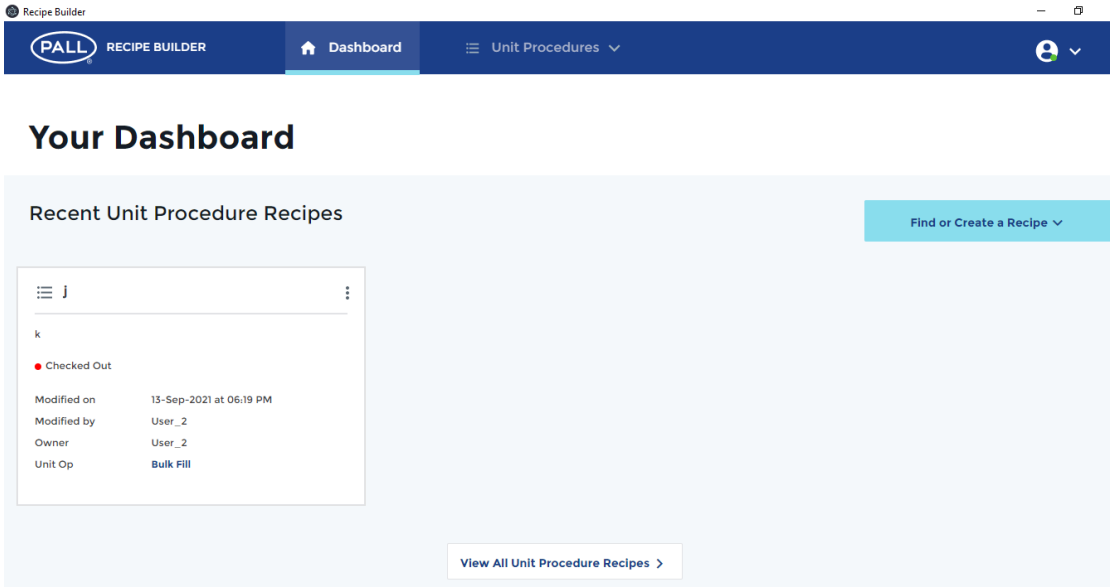
3	Enumeration	Integer

6.5 User Interface

6.5.1 Screens

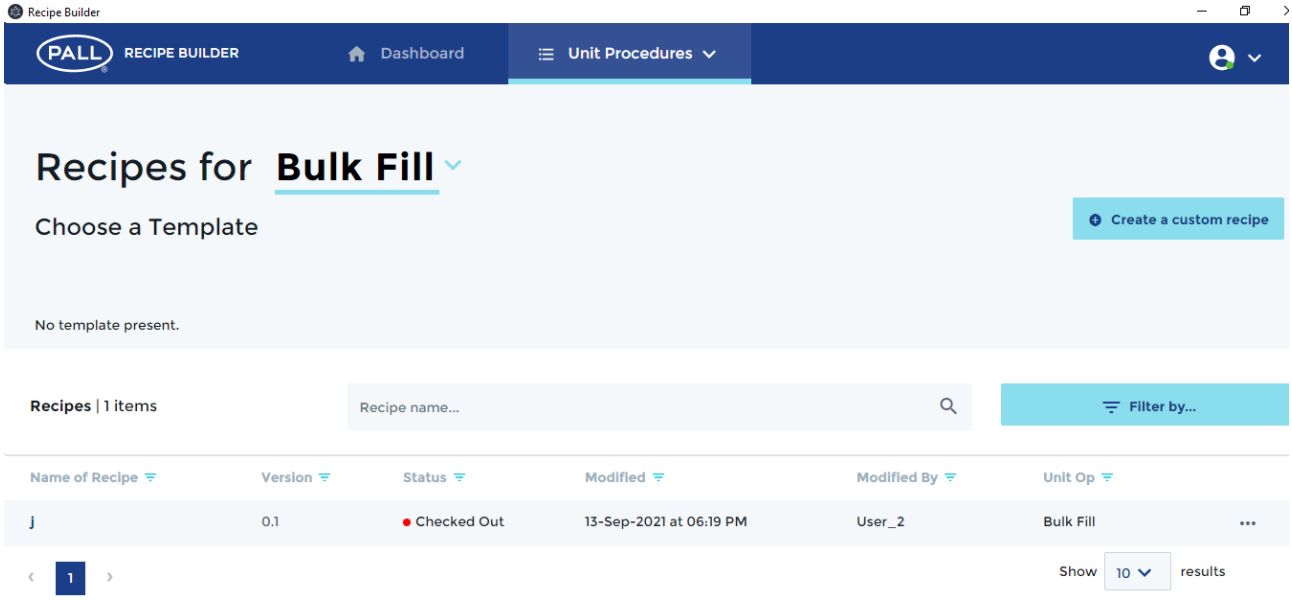
6.5.1.1 Dashboard

Logged in user can view recently modification to the recipes



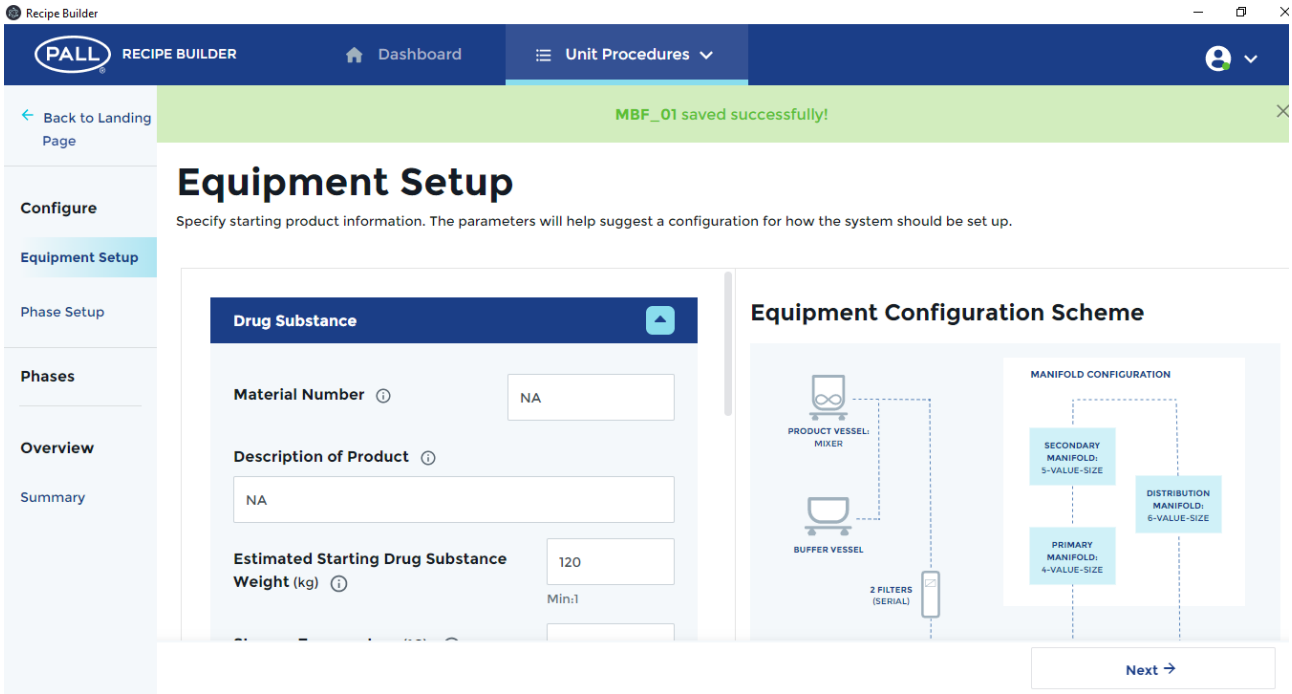
6.5.1.2 Unit Procedure

User can access recipes in the application to view there details and also create recipes in this screen.



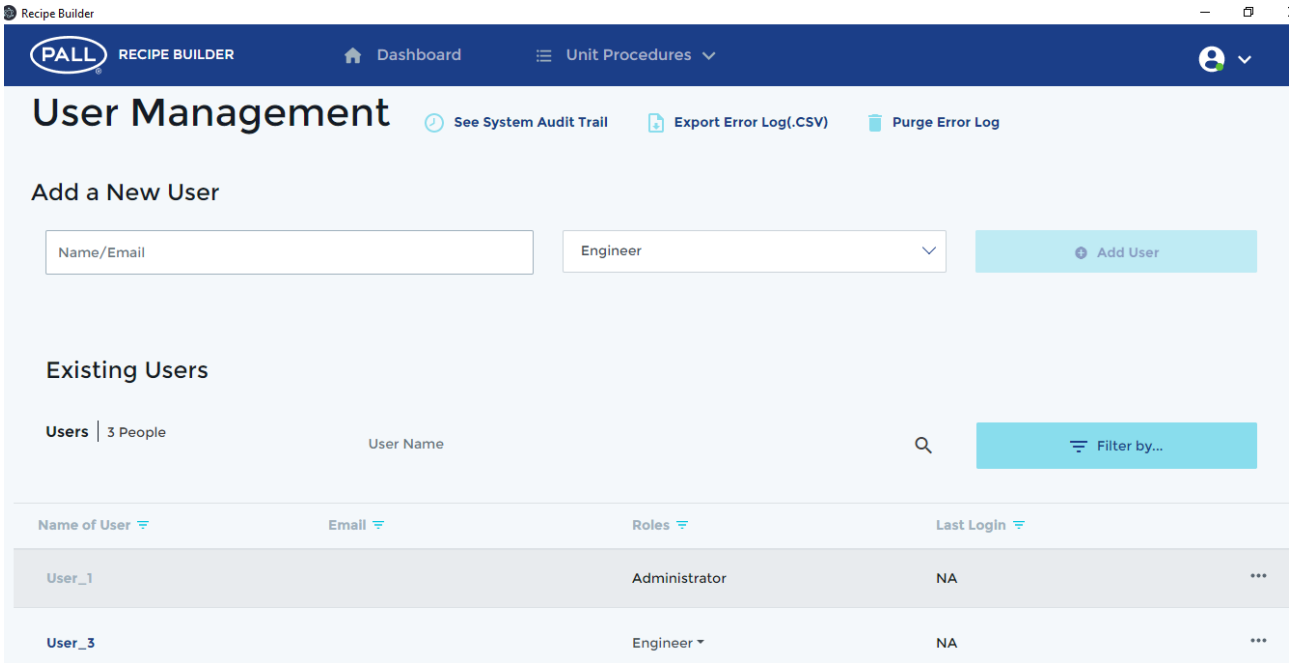
6.5.1.3 Recipe Edit Screen

Recipes are edited in this screen



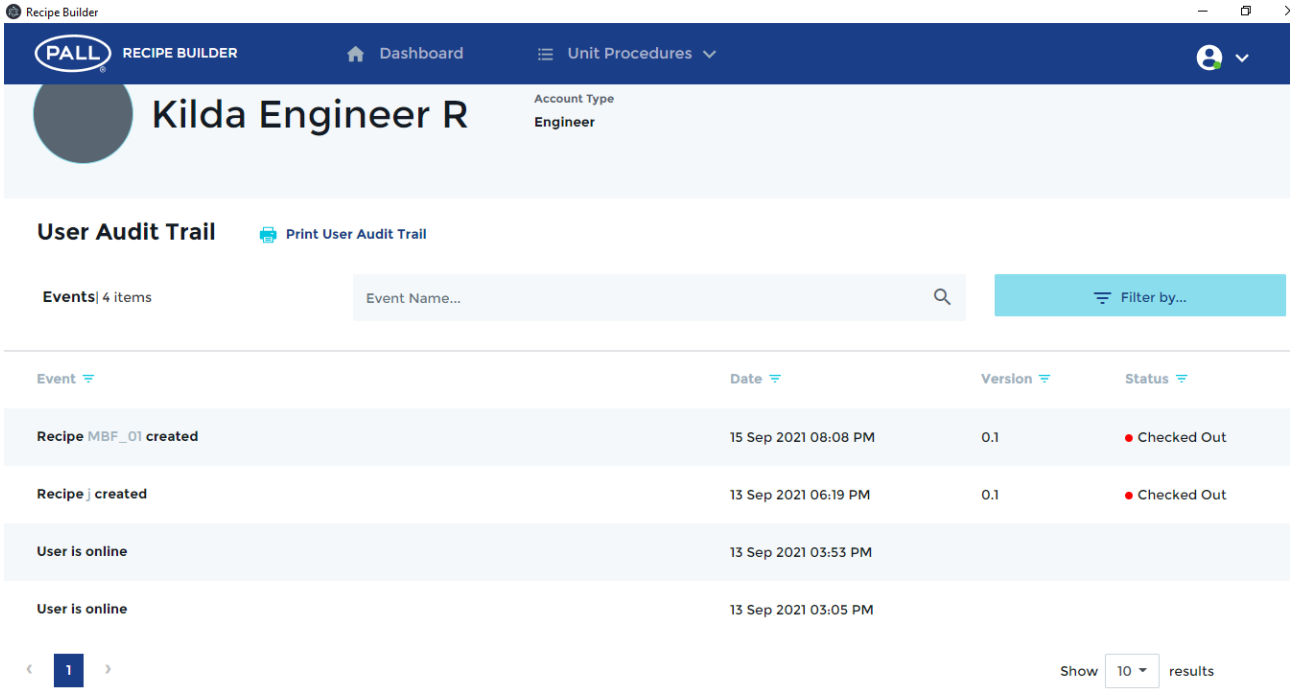
6.5.1.4 User management Screen

Administrator can manage users and user role from this screen. Administrator can also view system error log and system audit log of all users events



6.5.1.5 Audit Trails Screen

This screen displays the logged in users role and the users audit events generated by him/her

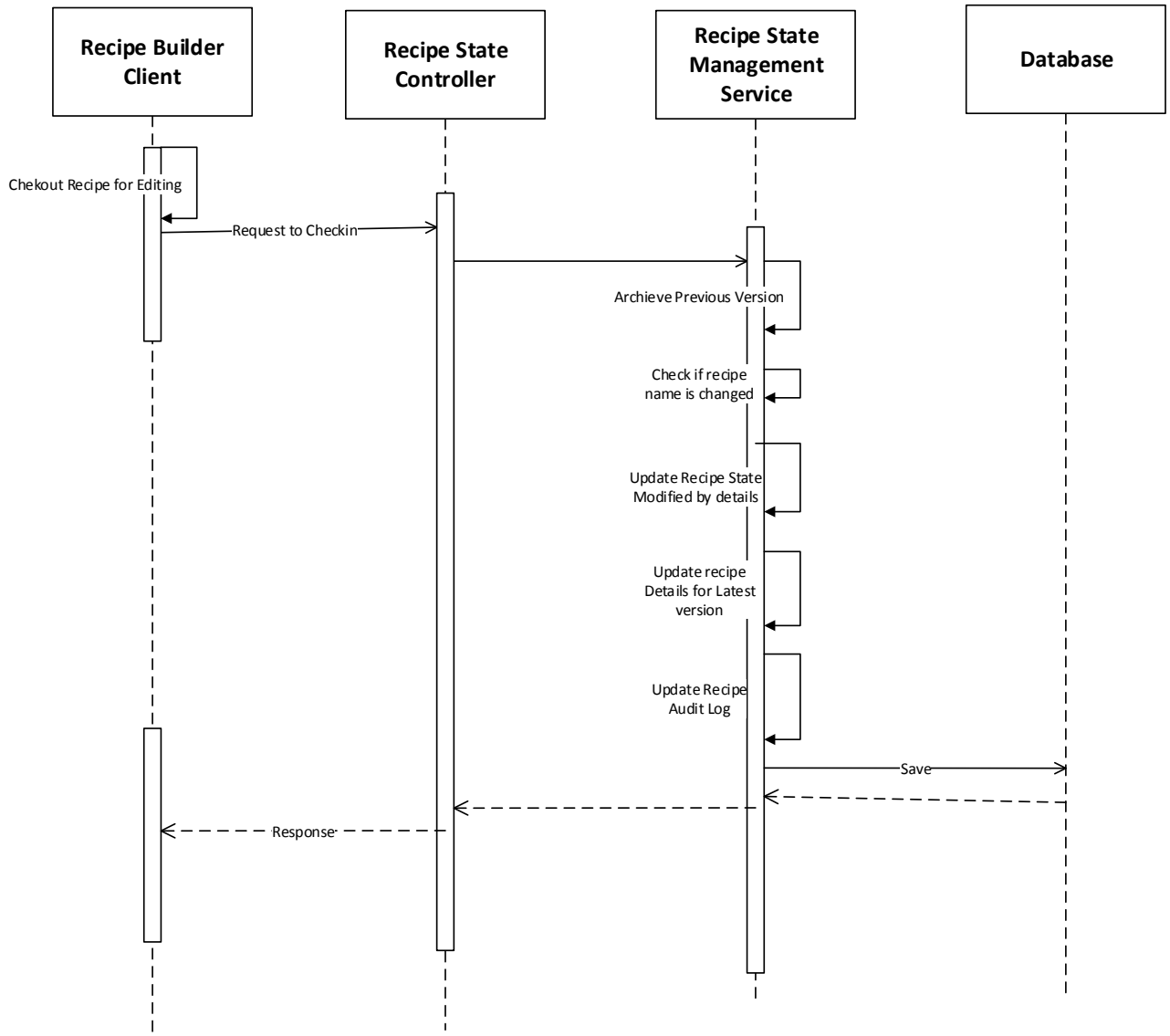


6.6 Use Case diagram

6.6.1 Check-in recipe

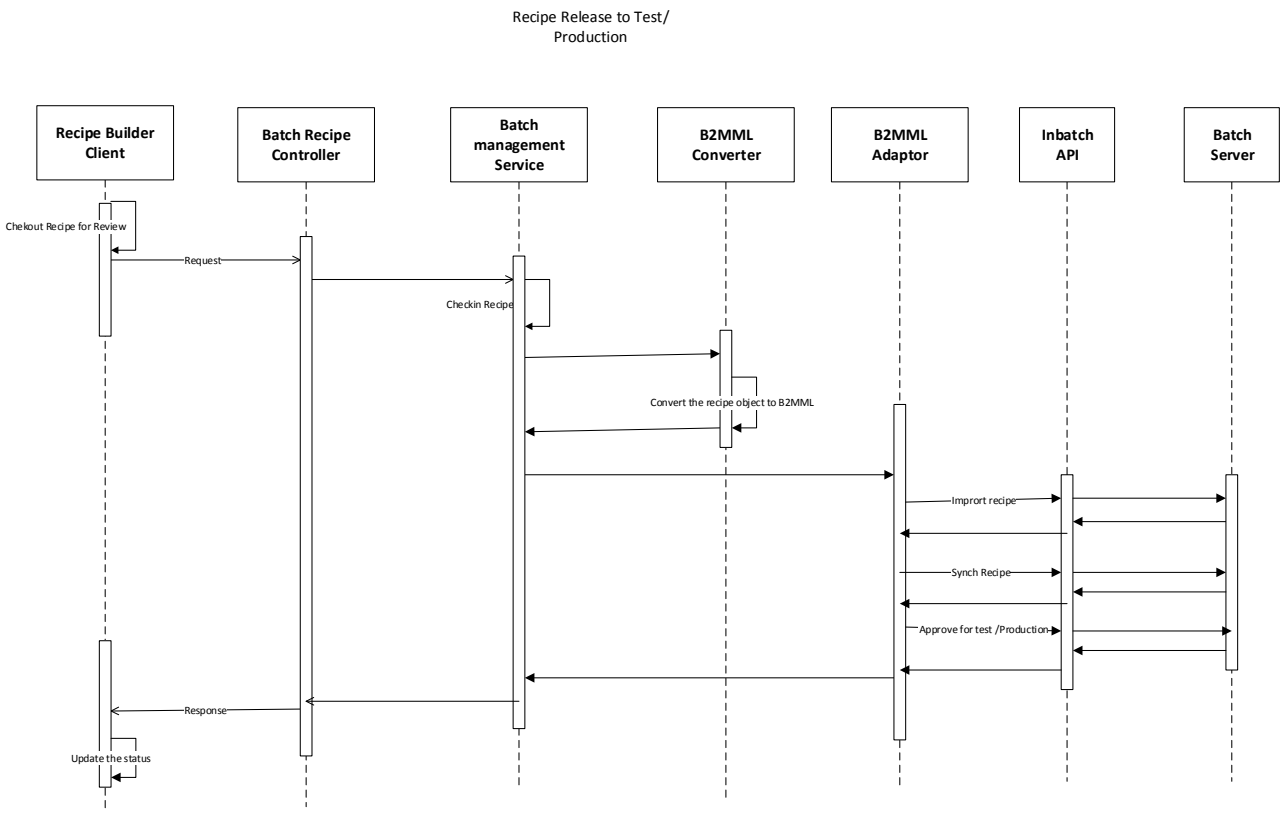
The user edits a recipe and stores it locally and performs checkin.

Recipe Check-in



6.6.2 Release recipe

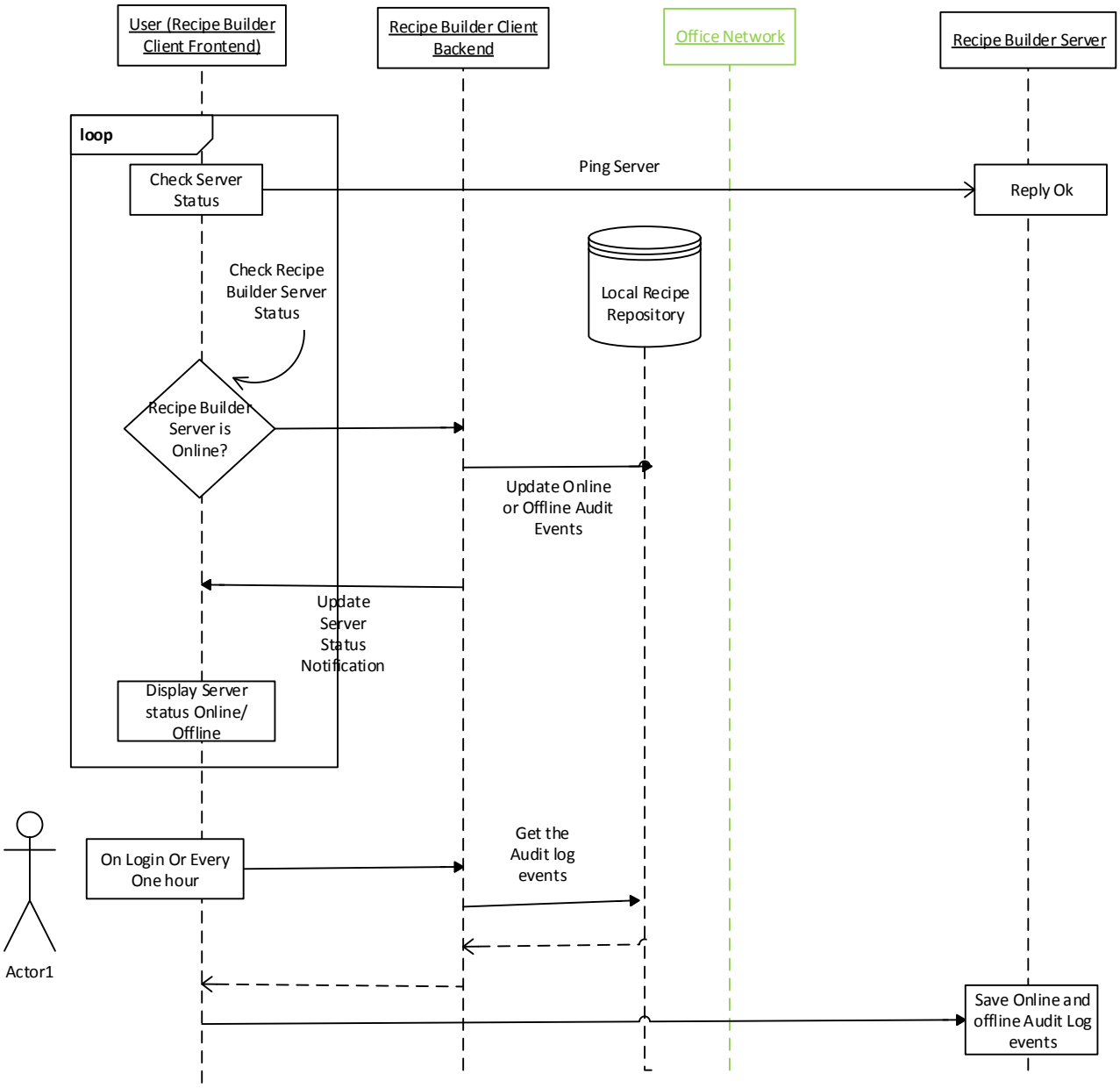
An approved recipe is transferred to the Batch Management System, which is reachable by network. The transfer is completed successfully after getting positive feedback from the Batch Management System.



6.6.3 Server Online/Offline Status

The Recipe Builder Server is offline. The Recipe Builder Client backend checks cyclically if the Recipe Builder Server becomes available. If the Recipe Builder Server gets online, the user if informed and can decide to switch to the online mode.

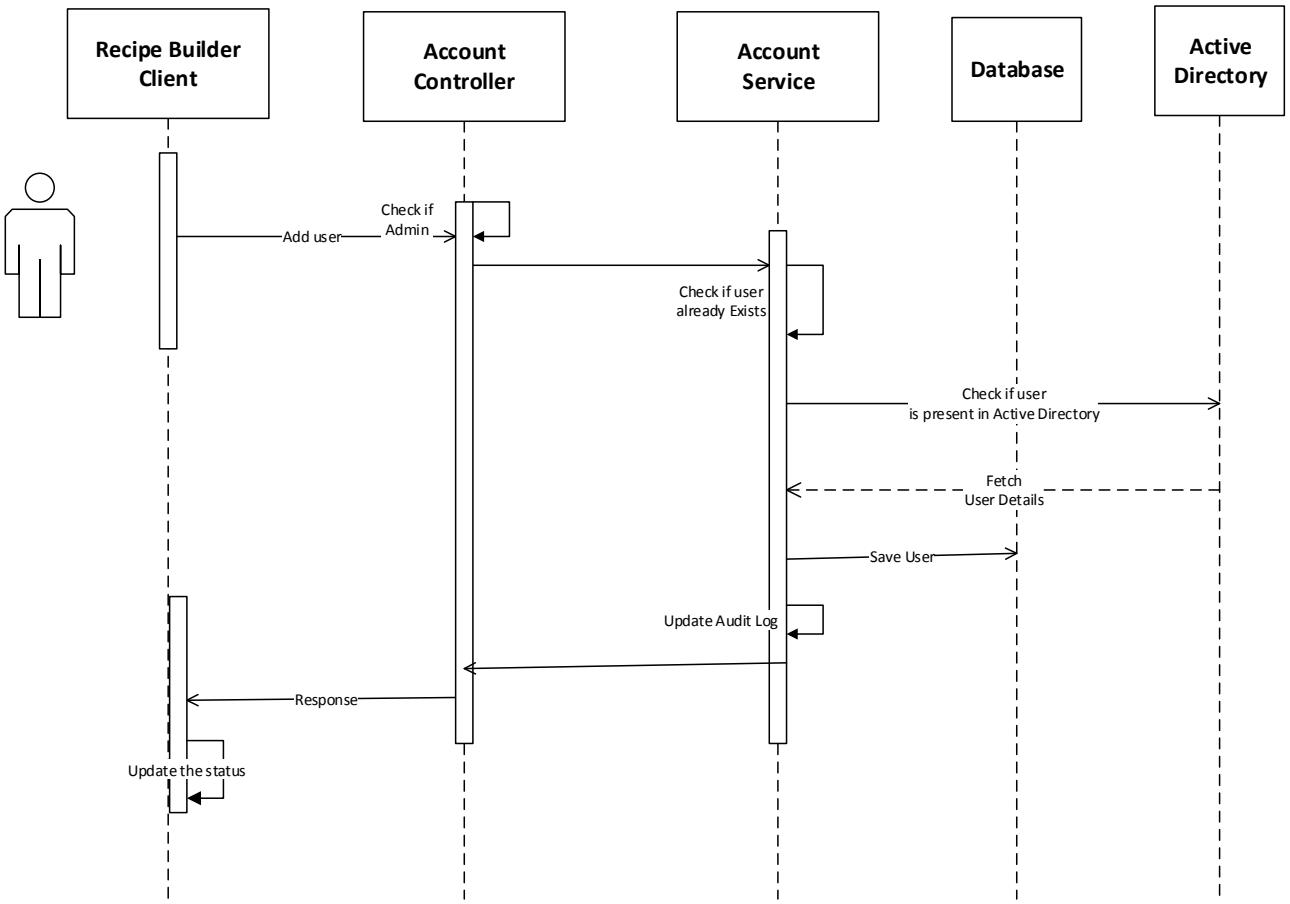
Server Online/Offline Status



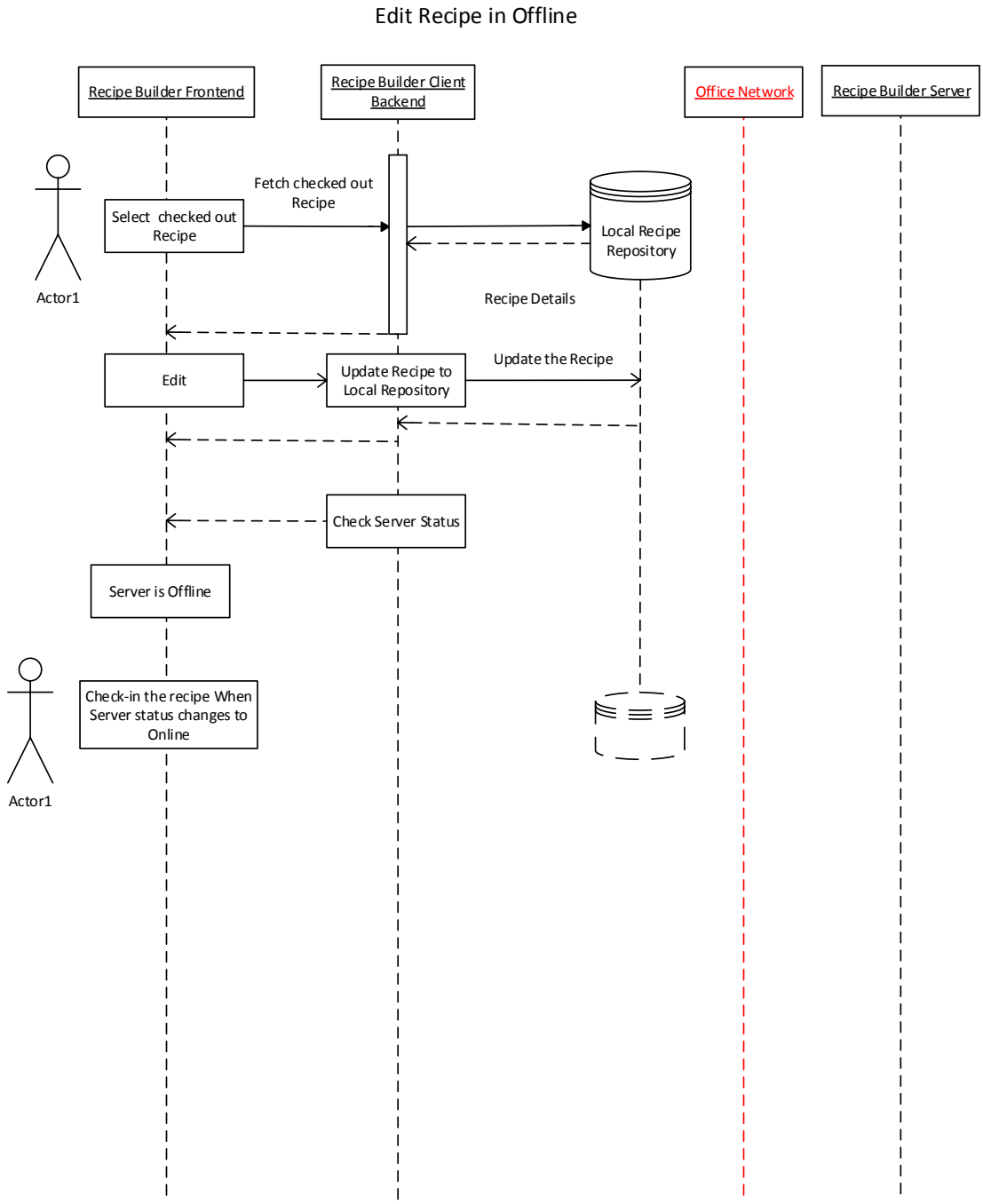
6.6.4 Add New User

Administrators adds user to recipe builder application

Add User


Recipe Builder Client
Account Controller
Account Service
Database
Active Directory
Add user
Check if Admin
Check if user already Exists
Check if user is present in Active Directory
Fetch User Details
Save User
Update Audit Log
Response
Update the status

6.6.5 Edit recipe Offline



6.7 Data Records

6.7.1 User Access

Only authorized users can access Recipe builder Client. Users present in the domain active directory only those users can be added as recipe builder users. Users are assigned roles by the administrator. Application supports roles Administrator, Viewer and Engineer.

Below is the matrix of features available based on the roles.

Roles	Administrator	Engineer	Viewer
Functions			
Create recipes	-	X	-
Edit Recipes	-	X	-
Approve Recipes	-	X	-
Release Recipes	-	X	-
View Recipe Audit Trail	X	X	X
View User Audit Trail	X	Logged-in user Audit Trail	Logged-in user Audit Trail
View system event log	X	-	-
Add / remove users	X	-	-
View Recipes	X	X	X
Promote Recipes as templates	-	X	-

"X" – Allowed , "-" – not allowed action

6.7.2 Audit Trail

6.7.2.1 Recipe Audit trail

Modification to the specific recipe is captured in "auditlog" table along with the recipe version, recipe state, recipe ID and user who performed the action

Below events are captured as part of recipe operations

- Recipe created,
- Recipe created from template,
- Recipe checked out,
- Recipe checked-in,
- Recipe duplicated,
- Recipe exported,
- Recipe state changed,
- Recipe released for test,
- Recipe released for production,
- Recipe archived,
- Recipe submitted for test,
- Recipe submitted for production,
- Recipe printed,
- Error releasing recipe,
- Recipe rejected for test,
- Recipe rejected for production,
- Recipe name changed,
- Performed Undo Checkout

6.7.2.2 User Audit trail

User audit trail is stored in "auditlog" table of database. Along with recipe audit log events below events are also captured for every user

- Log in,
- Log out,
- New user added,
- User deleted,
- Role updated,
- Recipe created,
- Recipe created from template,
- User role changed,
- Operation added as template,
- User is offline
- User is online
- Save phase as template

6.7.2.3 System Error Log Audit trail

Any exception caught in the server and client the errors are logged in “SystemErrorLog” table of database. This report can be fetched by the admin from Manage user’s screen

6.8 System Security

6.8.1 Client Authentication

Only authorized users can access Recipe builder Client.

Recipe Builder provides access only to user that has been added by the recipe builder Administrator. Those users which are part of the configured active directory of that domain can be added to the system. Users’ login credentials are validated by the active directory and recipe builder do not store User password

6.8.2 HTTPS Configuration

Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between Recipe Builder Client and Server. HTTPS is encrypted to increase security of data transfer. This protects users from “man-in-the-middle” attacks, where someone may steal the information being sent.

HTTPS uses an encryption protocol to encrypt communications. The protocol is called Transport Layer Security (TLS). This protocol secures communications by using an asymmetric public key infrastructure.

TLS certificate used, works by storing generated keys (public and private) in server. The public key is verified with the client and the private key used in the decryption process.

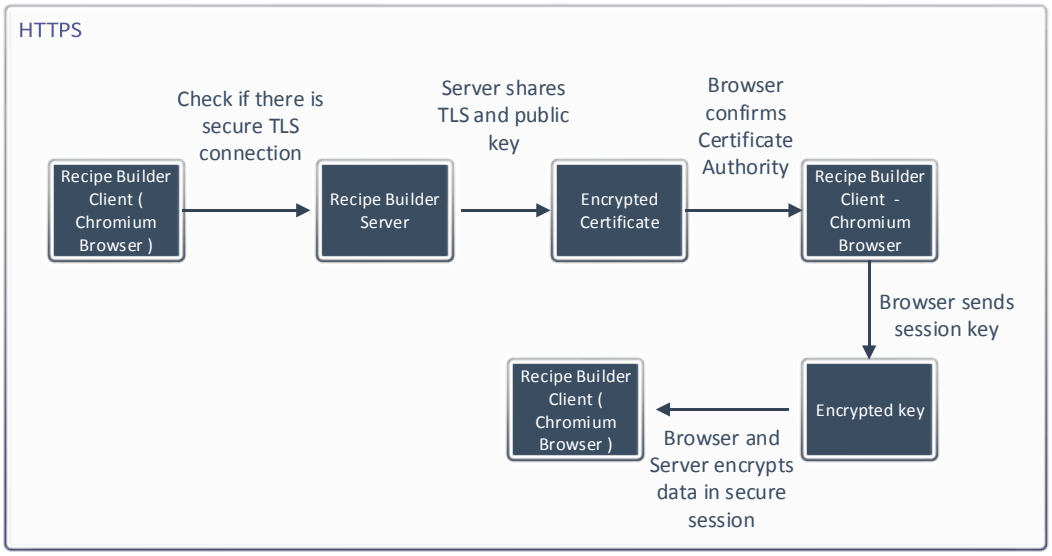


Figure 5 HTTPS Communication

6.8.3 REST API Basic Authentication and Authorization

The client sends HTTPS requests with the Authorization header that contains the word Basic followed by a space and a base64-encoded(non-encrypted) string username. The Server API can be accessed by only relevant users having a designated recipe builder role.

6.8.4 Online/Offline Authentication in client

Authorization and authentication are handled by login module of Recipe builder client.

When User is offline, one who has last logged in and the current windows logged in user is matched to proceed with the recipe builder authentication

6.8.1 Password Encryption

The database password in client and server is encrypted using AES (Advanced Encryption Standard) encryption algorithm.

.NET provides the algorithm through “AesCryptoServiceProvider” class. The key to encrypt password is stored in EncryptionHelper class of RecipeBuilder.Core module.

7 Open Source and third-party components

7.1 Recipe Builder Client

Thrid party	Version	License Type	Purpose	URL
angular-devkit/build-angular	0.1101.3	MIT License	This package contains Architect builders used to build and test Angular applications and libraries.	https://www.npmjs.com/package/@angular-devkit/build-angular
angular	11	MIT-style License	Angular is an application design framework and development platform for	https://angular.io/

			creating efficient and sophisticated single-page apps	
@types/node	12.11.1	MIT License	This package contains type definitions for Node.js (https://nodejs.org/).	https://www.npmjs.com/package/@types/node
codelyzer	6.0.0	MIT License	A set of tslint rules for static code analysis of Angular TypeScript projects.	https://www.npmjs.com/package/codelyzer
electron	11.2.2	MIT License	Electron is a free and open-source software framework developed and maintained by GitHub. It allows for the development of desktop GUI applications using web technologies	https://www.electronjs.org/
electron-packager	15.2.0	BSD 2-Clause "Simplified" License	Electron Packager is a command line tool and Node.js library that bundles Electron-based application source code with a renamed Electron executable and supporting files into folders ready for distribution.	https://www.npmjs.com/package/electron-packager
electron-reloader	1.2.1	MIT License	Simple auto-reloading for Electron apps during development	https://www.npmjs.com/package/electron-reloader
jasmine-core	3.6.0	MIT License	Jasmine is a Behavior Driven Development testing framework for JavaScript	https://www.npmjs.com/package/jasmine-core
jasmine-spec-reporter	5.0.0	Apache-2.0 License	Real time console spec reporter for jasmine testing framework.	https://www.npmjs.com/package/jasmine-spec-reporter
karma	5.2.0	MIT License	A simple tool that allows you to execute JavaScript code in multiple real browsers.	https://www.npmjs.com/package/karma
karma-chrome-launcher	3.1.0	MIT License	Launcher for Google Chrome, Google Chrome Canary and Google Chromium.	https://www.npmjs.com/package/karma-chrome-launcher
karma-coverage	2.0.3	MIT License	Generate code coverage using Istanbul	https://www.npmjs.com/package/karma-coverage
karma-jasmine	4.0.0	MIT License	Adapter for the Jasmine testing framework.	https://www.npmjs.com/package/karma-jasmine
karma-jasmine-html-reporter	1.5.0	MIT License	Reporter that dynamically shows tests results at debug.html page.	https://www.npmjs.com/package/karma-jasmine-html-reporter
protractor	7.0.0	MIT License	Protractor is an end-to-end test framework for Angular and AngularJS applications. Protractor is a Node.js program built on top of WebDriverJS	https://www.npmjs.com/package/protractor

ts-node	8.3.0	MIT License	ts-node is a TypeScript execution engine and REPL for Node.js. It JIT transforms TypeScript into JavaScript, enabling you to directly execute TypeScript on Node.js without precompiling.	https://www.npmjs.com/package/ts-node
tslint	6.1.0	Apache-2.0 License	TSLint is an extensible static analysis tool that checks TypeScript code for readability, maintainability, and functionality errors.	https://www.npmjs.com/package/tslint
typescript	4.1.2	Apache-2.0 License	TypeScript is a language for application-scale JavaScript. TypeScript adds optional types to JavaScript that support tools for large-scale JavaScript applications for any browser, for any host, on any OS	https://www.npmjs.com/package/typescript
@angular/material	11.2.4	MIT License	Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps	https://angular.io/
@ng-bootstrap/ng-bootstrap	9.1.2	MIT License	Angular widgets built from the ground up using only Bootstrap CSS with APIs designed for the Angular ecosystem	https://www.npmjs.com/package/@ng-bootstrap/ng-bootstrap
@ngx-translate/core	13.0.0	MIT License	The internationalization (i18n) library for Angular	https://www.npmjs.com/package/@ngx-translate/core
@ngx-translate/http-loader	6.0.0	MIT License	A loader for ngx-translate that loads translations using http.	https://www.npmjs.com/package/@ngx-translate/http-loader
body-parse	0.1.0	MIT License	Parse incoming request bodies in a middleware before your handlers, available under the req.body property.	https://www.npmjs.com/package/body-parser
body-parser	1.19.0	MIT License	Parse incoming request bodies in a middleware before your handlers, available under the req.body property.	https://www.npmjs.com/package/body-parser
bootstrap	4.6.0	MIT License	Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.	https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)
chart.js	2.9.4	MIT License	Chart.js is a free open-source JavaScript library for data visualization, which supports 8 chart types	https://www.chartjs.org/docs/latest/

cors	2.8.5	MIT License	Cross-origin resource sharing is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served.	https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
express	4.17.1	MIT License	Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software designed for building web applications and APIs.	https://expressjs.com/
install	0.13.0	MIT License	This command installs a package and any packages that it depends on	https://docs.npmjs.com/cli/v7/commands/npm-install
jquery	3.6.0	MIT License	jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.	https://jquery.com/
jslinq	1.0.22	MIT License	LINQ provider for Javascript	https://www.npmjs.com/package/jslinq
material-design-icons	3.0.1	Apache-2.0 License	Material Design Icons' growing icon collection allows designers and developers targeting various platforms to download icons in the format, color and size	https://materialdesignicons.com/
moment	2.29.1	MIT License	MomentJS is a JavaScript library which helps is parsing, validating, manipulating and displaying date/time in JavaScript in a very easy way .	https://momentjs.com/
mxgraph	4.2.2	Apache License 2.0	mxGraph is a JavaScript diagramming library that enables interactive graph and charting applications to be quickly created that run natively in any major browser that is supported by its vendor.	http://jgraph.github.io/mxgraph/
ng2-charts	2.4.2	MIT License	Beautiful charts for Angular2 based on Chart.js	https://www.npmjs.com/package/ng2-charts
ng2-search-filter	0.5.1	MIT License	Angular 2 filter to make custom search.	https://www.npmjs.com/package/ng2-search-filter
ngx-moment	5.0.0	MIT License	moment.js pipes for Angular	https://www.npmjs.com/package/ngx-moment
ngx-owl-carousel	2.0.7	MIT License	ngx-owl-carousel-o is built for Angular >=6.0.0.	https://www.npmjs.com/package/ngx-owl-carousel-o
ngx-ui-loader	11.0.0	MIT License	An all-in-one and fully customizable loader/spinner for	https://www.npmjs.com/package/ngx-ui-loader

			Angular applications. It supports foreground, background spinner/loader, indicative progress bar and multiple loaders	
node-cryptico	1.0.1	MIT License	Generating an RSA key pair & public key string	https://www.npmjs.com/package/cryptico
optional-require	1.0.3	Apache-2.0 License	node.js require that let you handle module not found error without try/catch. Allows you to gracefully require a module only if it exists and contains no error.	https://www.npmjs.com/package/optional-require
owl.carousel	2.3.4	MIT License	ngx-owl-carousel-o is built for Angular >=6.0.0.	https://owlcarousel2.github.io/OwlCarousel2/
pg	8.6.0	MIT License	Non-blocking PostgreSQL client for Node.js. Pure JavaScript and optional native libpq bindings	https://www.npmjs.com/package/pg
pg-hstore	2.3.3	MIT License	node package for serializing and deserializing JSON data to hstore format	https://www.npmjs.com/package/pg-hstore
postgres	1.0.2	PostgreSQL License	PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance	https://www.postgresql.org/
rxjs	6.6.0	Apache-2.0 License	RxJS is a library for reactive programming using Observables, to make it easier to compose asynchronous or callback-based code.	https://rxjs.dev/
script-loader	0.72	MIT License	A library for loading JavaScript files asynchronously. It loads script files by injecting script tags into DOM during runtime.	https://www.npmjs.com/package/script-loader
sequelize	6.6.2	MIT License	Sequelize is a promise based Node.js ORM for Postgres.	https://sequelize.org/master/index.html
sequelize-auto	0.82	MIT License	Automatically generate models for SequelizeJS via the command line.	https://www.npmjs.com/package/sequelize-auto
tslib	2.0.0	BSD Zero Clause license	This is a runtime library for TypeScript that contains all of the TypeScript helper functions.	https://www.npmjs.com/package/tslib
zone.js	0.11.3	MIT License	A zone is an execution context that persists across async tasks	https://www.npmjs.com/package/zone.js?activeTab=readme

7.2 Recipe Builder Server

Dependency	Version	License Type	Purpose	URL
Auto Mapper	10.1.1	MIT	AutoMapper is a simple little library built to solve a deceptively complex problem - getting rid of code that mapped one object to another.	https://automapper.org
Entity Framework	6.4.4	Apache License 2.0	Entity Framework 6 (EF6) is a tried and tested object-relational mapper for .NET with many years of feature development and stabilization.	NuGet Gallery | EntityFramework 6.4.4
EntityFramework6.Npgsql	6.4.1	PostgreSQL License	PostgreSQL provider for Entity Framework 6	NuGet Gallery | EntityFramework6.Npgsql 6.4.3
MediatR	9.0.0	Apache License 2.0	imple, unambitious mediator implementation in .NET	NuGet Gallery | MediatR 9.0.0
Microsoft.AspNet.Mvc	5.2.7	MICROSOFT SOFTWARE LICENS - .net Framework	This package contains the runtime assemblies for ASP.NET MVC. ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives you full control over markup.	NuGet Gallery | Report Abuse by Microsoft.AspNet.Mvc 5.2.7
Moq	4.16.1	BSD 3-Clause License https://raw.githubusercontent.com/moq/moq4/master/License.txt	Moq is the most popular and friendly mocking framework for .NET.	https://www.nuget.org/packages/MSTest.TestAdapter/1.2.0/
Newtonsoft.Json.	13.0.1	MIT - https://licenses.nuget.org/MIT	Json.NET is a popular high-performance JSON framework for .NET	NuGet Gallery | Newtonsoft.Json 13.0.1
Npgsql	5.0.7	PostgreSQL License	Npgsql is the open-source .NET data provider for PostgreSQL.	https://www.nuget.org/packages/Npgsql/5.0.7
Serilog	2.10	Apache License 2.0	Simple .NET logging with fully structured events	https://www.nuget.org/packages/serilog/
Swashbuckle	5.6.0	The 3-Clause BSD License | Open Source Initiative	Seamlessly adds a Swagger to Web Api projects!	NuGet Gallery | Swashbuckle 5.6.0
Unity	5.11.1	Apache License 2.0	his package contains Unity Container and Abstractions libraries as a single package.	NuGet Gallery | Unity 5.11.10
WebActivatorEx	2	Microsoft Public license (Microsoft Public	A package that allows other packages to execute some start-up code in web apps. This package should be used over	https://www.nuget.org/packages/WebActivatorEx/2.0.0

		License (MS-PL) | Open Source Initiative)	the older Web Activator, which was not strong named.	
Castle.Core	4.1	Apache License 2.0	Related to mock object creation. Used for unit test.	Castle Project

8 Version Control

Version control, also known as source control, is used for tracking and managing modification to software code. The source code for Recipe Builder application is maintained in AWS cloud Server.

AWS Code Commit provided by Amazon is a secure, highly scalable, managed source control service that hosts private Git repositories. It makes it easy for teams to securely collaborate on code with contributions encrypted in transit and at rest. It supports the standard functionality of Git; it works seamlessly with existing Git-based tools.

The AWS code commits URL: https://dhr-lsc-portal.awsapps.com/start/#/.

8.1 Code Branch

Below are the code branches and its details

8.1.1 Main branch:

Release candidates are be merged to this branch

8.1.2 Development branch:

All sprint end code is merged to this branch. Each UOE development is in separate development branch

8.1.3 Sprint Branches:

Sprint related Feature or task is merged here

8.1.4 User Story or Defect:

Developers use this branch to implement user stories and fix defects

8.2 Branching Rules

1. Every end of Agile Sprint after verification and validation code is merged to “Development” branch
2. If there are any hot fix to be given with respect to previous sprint activity. Hot fix is merged back to Development branch. Same hot fix will be merged to current sprint Branch also
3. Git Tags will be created when significant merges are made
4. All release candidates will be made from master branch.
5. Code review is performed whenever code is merged to any of the branches

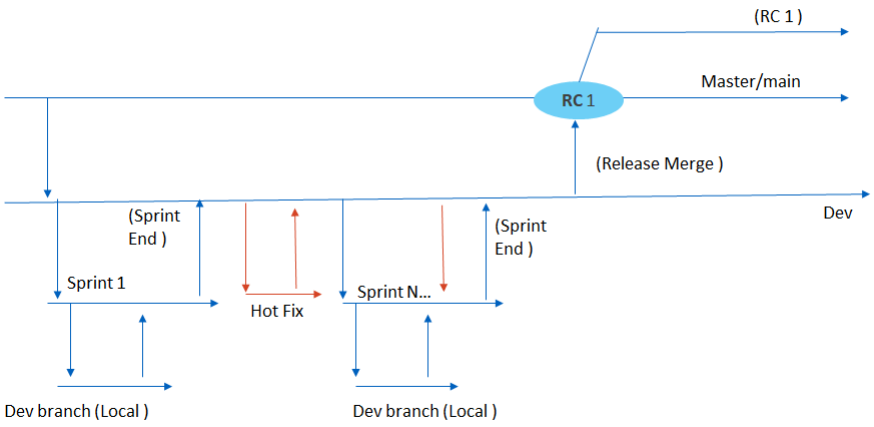


Figure 6 Branching Strategy

9 Glossary

Guidelines

Definitions of any terms and abbreviations that may be unfamiliar to the readership of the document should be included.

TERM	DEFINITION
UOE	Unit Operation Equipment
P and I D	Process and instrumentation Diagram
IIS	Internet Information Services
HTTPS	Hypertext transfer protocol secure
API	Application Programming Interface
AWS	Amazon Web Services

10 Revision Index

ISSUE	DATE	BY	CHANGE	PAGE
01	11-10-2021	Bindya B	Initial version	All

Pall Corporation

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

Visit us on the Web at www.pall.com

Contact us at www.pall.com/contact

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.

Filtration. Separation. Solution.SM