8/28/2018

Chandru k

Gramener IT SERVices

Instruction Document - Task Management & Status Reporting

**Instruction Document History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Role | Name | Comments |
| 28.08.2018 | 1.0 | Final | Chandru | Creation v1.0 |
|  |  |  |  |  |

**Table of Contents**

Contents

[1. Introduction 3](#_Toc523216270)

[1.1 Purpose of the document 3](#_Toc523216271)

[1.2 Objectives 3](#_Toc523216272)

[2. AS IS Process Description 4](#_Toc523216273)

[2.1 Process Overview 4](#_Toc523216274)

[2.1.1 Inputs needed for execution 4](#_Toc523216275)

[2.1.2 Output 5](#_Toc523216276)

[2.2 Detailed Functional Diagram 6](#_Toc523216277)

[2.3 Flow Description 7](#_Toc523216278)

[3 Implemented Functionalities: 7](#_Toc523216279)

[Send Task: 7](#_Toc523216280)

[Ad-hoc Requirement: 7](#_Toc523216281)

[Status Reporting: 8](#_Toc523216282)

[Exception Handling: 8](#_Toc523216283)

[4 Extended Functionalities: 8](#_Toc523216284)

[Data Source: 8](#_Toc523216285)

[File Location: 8](#_Toc523216286)

[Metrics Collection: 8](#_Toc523216287)

# Introduction

## 1.1 Purpose of the document

The Instruction Document describes the Project Objective, Functional Design, Component details, Use cases and Implementation of Task Management and Status Reporting using UiPath Robotic Process Automation (RPA).

## Objectives

The Project & Process Management activities are an integral part of any project and there are a lot of activities done on a daily basis, by Project Teams which are repetitive. There are tools available in the market for project management, but with limited customizations.

I believe UiPath as RPA tool can fulfill the automation of Project management activities with essential customization, according to any custom requirement. In this initial Phase, I have automated the Task Management and Status Reporting activities. Functionalities can be extended to Metrics collection, Resource Management, Defect Tracking and many more, which is not in scope for this phase.

Automated Task Management & Status Reporting includes the below features:

* Email resources based on their roles, with the planned task list for the complete project.
* Daily Tasks for the respective users will be sent in the morning
* Follow up emails will be sent to the users, until the task completion
* Notify Project team about the ad-hoc requirements added(Unplanned task)
* Send Status report to Client at end of the business day.

By using UiPath as Project Management tool, below are the following key benefits for the Project Managers:

* *Automated email to resources about the planned task and daily follow up*
* *Can customize subject and mail content in email notifications*
* *Automated timely reporting, helps to build trust and transparency*
* *Can define the process initially, which has to be followed throughout the project*
* *Never miss any planned/unplanned task out of tracking*
* *As an extended functionality, Manager can build project/resource specific actions to run at any specific time*

# AS IS Process Description

## 2.1 Process Overview

General information about the process selected for RPA implementation, prior to its automation:

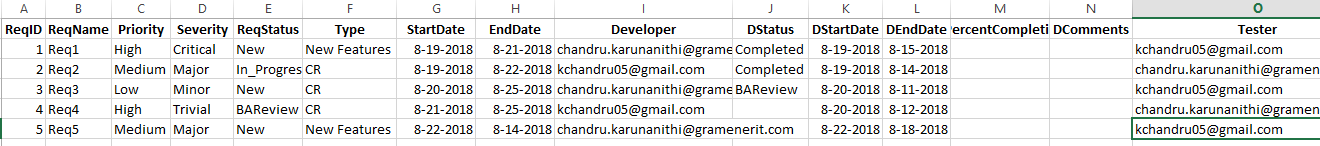
|  |  |
| --- | --- |
| AS IS Process details | |
| Process Full Name | Task Management and Status Reporting |
| Function | Process automation |
| Department | Project Management |
| Short description | Process automation of the day to day tasks of the project manager and team members |
| The role required for running the process | Once published, can use the scheduler to run the project in orchestrator. Anyone with respective orchestrator login can modify/run the project |
| Process Schedule | 2 schedulers running at the specified time during all working days |
| No. of Items process per day | Depends on Task List |
| Average execution time | Less than a minute for avg of 50 tasks at any given time. |
| Peak period | No Peak period |
| Level of Exception rate | Input Files, not present/kept open during the execution |
| Input Data | Config, Task List, Status List Work Sheet, Ad-hoc Requirement document (if any) and notepad file  More details in 2.1.1 |
| Output | Adhocrequirement Sheet and Mail notification to project team members.  More details in 2.1.2 |

### 2.1.1 Input Data:

Below are the inputs files required for the system to run the Task Management and Status reporting bots. This Pre-condition is must to be included in Data Folder.

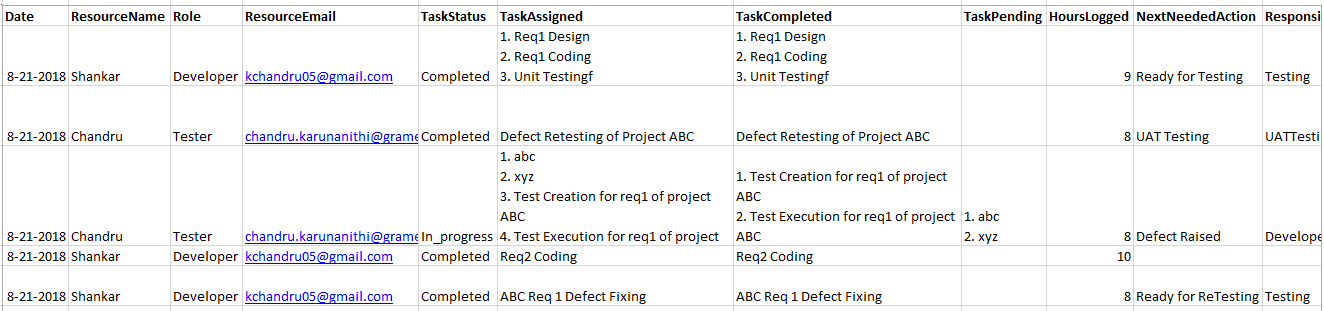
**Config File** – Consists of Settings and Constant used in UiPath xaml files. Initially, System looks out for the config file to read all the Settings(File Paths)and Constant (Variables)

**TaskList Workbook**– From the requirement document of the project, the Project Manager/Head identifies the requirements and create the task list excel and assign resources, define schedule and effort for each task. Sample Task List file has to be added to Data\Input Folder. The file will look something like below:



**StatusList Workbook**– This workbook should be created and placed in Data\Input Folder. It is expected that the resource should fill in the status of the activities they perform on a daily basis. This workbook will be read the bot to send the status report to the client.

\*Date Fields in the Task List should be in Date Format.



**ExtractColumnData** – Create Blank Notepad file to store the extracted column value temporarily when reading the excel file.

**AdhocRequirement** – Other than planned task defined in the task list, if there is any unplanned task identified during project development, Ad-hoc requirement should be created with all details in PDF and place it in data\Input Folder. The naming convention as AdhocRequirement\_[ID].pdf

During Execution, PDF File is read and details are stored in AdhocRequirement workbook [Data\Output Folder] and Send mail to Project Team.

### 2.1.2 Output

**Mail Notification:** As per defined condition, mail notification will be sent across all project team members based on Task list and assignee.

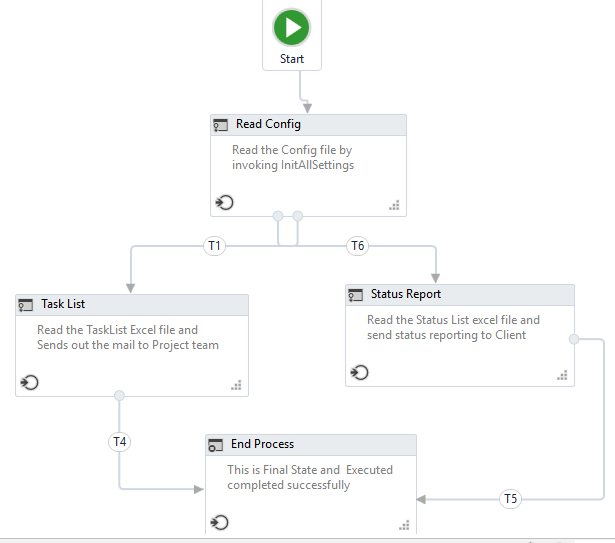
**Status Reporting:** Based on Status report details entered by the resources, Status reports will be sent to stakeholders/Client included with necessary details.

**AdhocRequirement**: Ad-hoc Requirement PDF File is read and details are stored in AdhocRequirement workbook [Data\Output Folder]

### 2.1.3 Assumptions

* + Task list should follow pre-defined template and details have to be filled in with proper data and placed in Data\Input Folder
  + Status List should follow a pre-defined template and details have to be filled in with proper data by the resources and placed in Data\Input Folder
  + Task and Status List should be updated on a daily basis if and only if there are any updates
  + Date Fields in the Task List and Status List should be in Date Format (mm-dd-yyyy)
  + Currently, only 1 developer & tester can be assigned to a task

## 2.2 Detailed Functional Diagram



## Flow Description

The short description of the component used in the project as below:

|  |  |
| --- | --- |
| **Step** | **Short Description** |
| 1 | Configure the scheduler in orchestrator to run at 8 am and 9 pm for every working day using unattended Bot. |
| 2 | Read Config file for all the inputs like source data path(Tasklist), Settings and Constant |
| 3 | Check the current hour and assign a value to variable Trigger |
| 4 | Based on the trigger value, Condition is to execute the State accordingly |
| 4.1 | If current hours is 8 am, Task list should be executed. |
| 4.1.1 | Data Loading – Read & load the Task List source data |
| 4.1.2 | Call Dataload Workflow – It sends Task details to Developer, Tester, Client, and Manager based on a defined condition |
| 4.1.2 | Call Adhoc Requirement workflow – Checks if there is any ad-hoc requirement document added by Client and send the same to Project team if added |
| 4.1.3 | Upon completion, it executes the end process and complete. |
| 4.2 | If the current hour is 9 pm, Status Report Data should be executed |
| 4.2.1 | Call Status Report Data – Checks status report excel input file and filter today’s status and send mail to Client. |
| 4.2.2 | Upon completion, it executes the end process and complete |

# Implemented Functionalities:

## Send Task:

As a daily task, reads the task list placed at Data\Input Folder and load it as Data table. Based on a condition defined in DataLoad.xaml it sends out the mail to Project team includes Developer, Tester, Support(if any), Project Manager, Client, and BA

## Ad-hoc Requirement:

Other the planned task defined in Requirement Source, if there are any ad-hoc requirement added, then the system will read the file (Data\Input\ Folder) and send out the mail to Project manager. Ad-hoc requirement details are extracted from PDF and added to excel sheet placed in (Data\Output Folder) and then file moved to Ad-hoc Requirement Folder placed in the Input.

## Status Reporting:

As a daily task, team member updates the status of the day in the excel file. StatusReportData.xaml will read this file placed in Data\Input Folder and read the data based on a condition defined and send out the status mail to Project Client.

## Exception Handling:

All the Exceptions are handled across application and a valid log message is printed. This prevents the application to stop in-between the execution and avoid manual intervention.

# Extended Functionalities:

## Data Source:

As per current implementation, the data source of the application is considered as MS Excel. It can be replaced by any web-based/Desktop tool as per need.

## File Location:

Project files are currently been placed in Data Folder of Application and read/write from the application. It could be difficult to project team to read/write in UiPath Folder, so this location can be changed to any Online File Repository where UiPath can read data file for processing.

## Metrics Collection:

In Requirement and Status report data source, Schedule and Effort is identified and it can be used to calculate the metrics calculation according to Project, Task, and Resource. Tracking defect details help to build metrics related to defect.