Software Requirement Specification

for

**JCNEx**

**Java Contract Number Extractor**

Version 1.0

Prepared by team “Bration”

15-08-2015

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Changes | Version |
| First release | 15-08-2015 | Initial release | 1.00 |
|  |  |  |  |

Table of Contents

[Revision History 1](#_Toc427499126)

[Introduction 3](#_Toc427499127)

[Purpose 3](#_Toc427499128)

[Document Conventions 3](#_Toc427499129)

[Project Scope 3](#_Toc427499130)

[References 3](#_Toc427499131)

[Overall Description 4](#_Toc427499132)

[Product Perspective 4](#_Toc427499133)

[Product Features 4](#_Toc427499134)

[User Classes and Characteristics 4](#_Toc427499135)

[Operating Environment 4](#_Toc427499136)

[Specific Requirements 5](#_Toc427499137)

[Input file requirements 5](#_Toc427499138)

[Output file requirements 5](#_Toc427499139)

[User interface requirements 6](#_Toc427499140)

[Start process screen 6](#_Toc427499141)

[Prompt messages requirements 6](#_Toc427499142)

[Functional Requirements 7](#_Toc427499143)

[Use case related to select an input file: 7](#_Toc427499144)

# Introduction

## Purpose

In this document are specified the software requirements for JCNEx, Java based system that extract contract number from XML file. This documents includes functional, non-functional requirements and use cases for all features..

## Document Conventions

This document was created when the JCNEx was in final development process, so all requirments described in the document are already satisfied. However, it is important to update this document with every future requirement to stay informed. This document is intended for both the stakeholders and the developers of the system

## Project Scope

JCNEx is a JAVA based application for parsing, analyzing and extract Contract number form each company invoice received using XML file format. The JCNEx read the XML input file (**invoice.xml**), and to process it to look for specific tags POnumber which contain specific Contract number which need to be extracted in a new tag, for better used. This approach gives an output file with POnumber and Contract number which can be used to create the report document with full information about clients and companies.

## References

IEEE 830-1998 standard for writing SRS document.

# Overall Description

## Product Perspective

JCNEx is specifically designed for the use of company or institutions which use digital document exchange.

The JCNEx application is composed of user friendly interface which consist a login screen and executing element which process the input file and present a result in a file structure.

User interface

Analyzing, parsing and extracting

Output Files

Input Files

## Product Features

The following list offers a brief outline and description of the main features and functionalities of the JCNEx.

Main Features:

* Extract Contract number from PO number of invoice xml file
* Analyze xml document for data which are not meet the specification
* Create an input file with splitted Convert and PO numbers in different tags.

## User Classes and Characteristics

JCNEx can be used from developers that want to incorporate it to their applications or to used it as a base to build an application. The simplicity of using JCNEX makes it easy to use to anyone that wants to take advantage of the XML file parser or extractor.

It can be used also from companies and institutions which already have a digital document exchange process using XML files.

## Operating Environment

JCNEx has written in Java so is compatible with all the operating systems i.e. is OS platform independent. It only need jvm. (java virtual machine).

# Specific Requirements

## Input file requirements

For input file, JCNEx will use a single XML file, invoices.xml. This file consists of many similar XML trees. It has a document root, <invoices>, which can contain one or more instances of the <invoice> element. One of the internel elemnt need to be PO number <PONumber> which contain a Contract number which is mark with “-” and can be extract.

Sample input XML file structure is:

<Invoice>

<InvoiceNumber>INV001</InvoiceNumber>

<Date>2015/08/08</Date>

<PONumber>123456-CT123</PONumber>

<TotalNet>100.00</TotalNet>

<TotalVAT>20.00</TotalVAT>

<TotalGross>120.00</TotalGross>

</Invoice>

If some of these requirements are not exist in input file it should be rejected.

## Output file requirements

The Output file should be contain the same structure of Input file with additional Contract number which extract from PONumber and add into tag Contract (<Contract>).

Samle output XML file structure is:

<Invoice>

<InvoiceNumber>INV001</InvoiceNumber>

<Date>2015/08/08</Date>

<PONumber>123456</PONumber>

<Contract>CT123</Contract><TotalNet>100.00</TotalNet>

<TotalVAT>20.00</TotalVAT>

<TotalGross>120.00</TotalGross>

</Invoice>

## User interface requirements

### Start process screen

JCNEx hava a sample graphical interface wich help user to select the input file and folder where output file shoulde be saved.

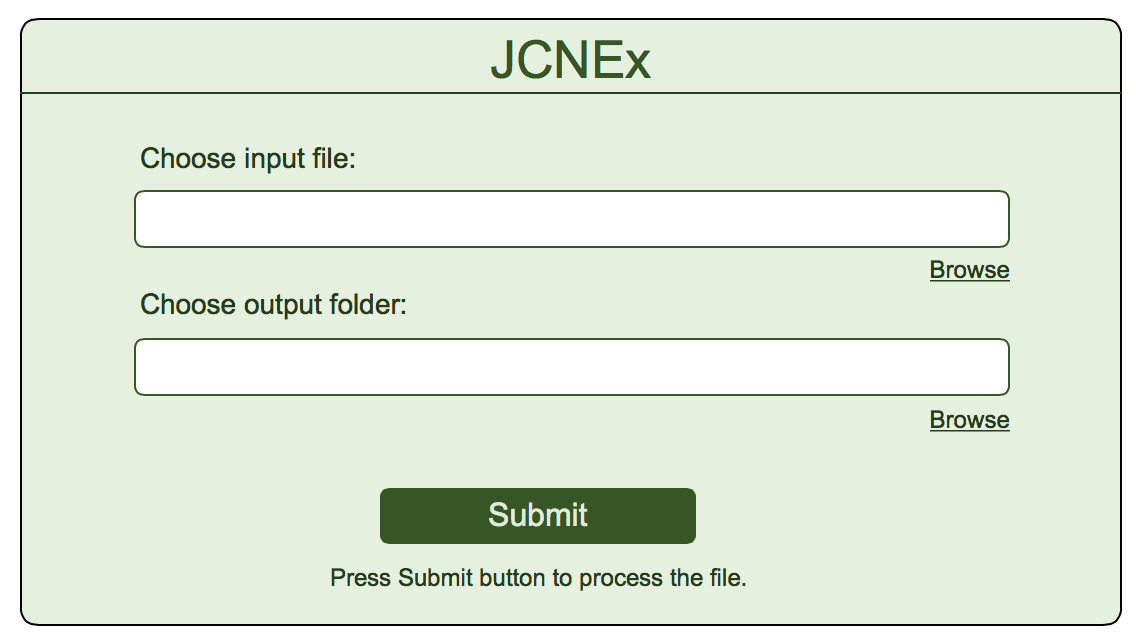
It contains:

Input file field and browse button to choose the file.

Output file field and browse button to choose the location folder of output file.

Submit button to start processing.

In next screen is present a mockups of the interface:

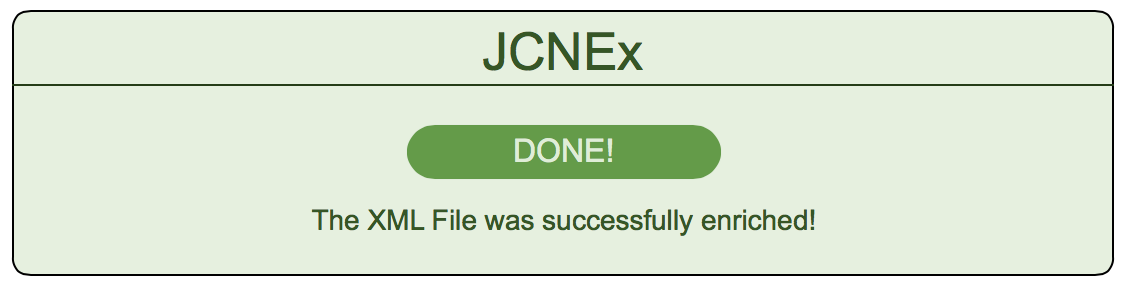


### Prompt messages requirements

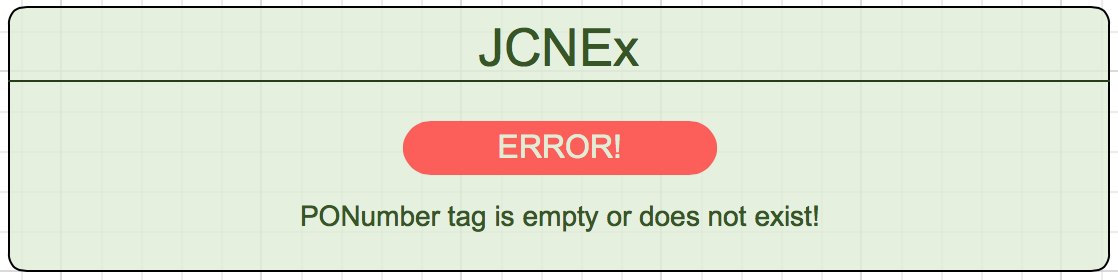
Ever process of selected file complete with the promt message which present if the process is complete successfully or the file is incorrect and hava some Error.

The prompt messages are:

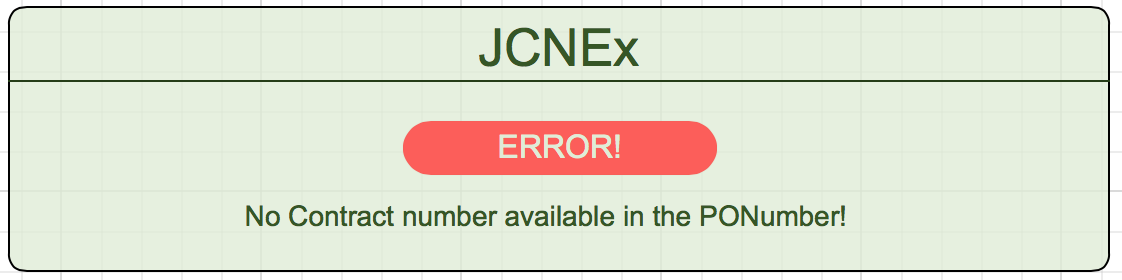
* Complete process successfully:



* Input file is incorrect:
  + If the PONumber tag is not exist in input file.



* + If the “-” is not exits in POnumber



## Functional Requirements

We describe the functional requirements by giving various use cases.

### Use case related to select an input file:

**Use Case 1:** Select an input file

Primary Actor: User

Pre Condition: Application is start and prompt window to choose file is released

Main Scenario:

1. Press the Browse button.
2. Standard Open screen is displayed.
3. User choose the input file with is in .xml format
4. Press OK

Result: The name and location path is displayed in input file field on Choose file application screen.

Alternative Scenario:

3a) Choose the input file which is not .xml file

3a) 1. Press OK

Result: Prompt message displayed that file format is invalid.

**Use Case 2:** Select a correct input file

Primary Actor: User

Pre Condition: Application is start and prompt window to choose file is released

Main Scenario:

1. Press the Browse button.
2. Standard Open screen is displayed.
3. User choose the input file with correct .xml structure
4. Press Submit button

Result: The process is complete successfully the Prompt message appeared and display “The XML file is successfully enriched.”

**Use Case 3:** Select an incorrect input file

Primary Actor: User

Pre Condition: Application is start and prompt window to choose file is released

Main Scenario:

1. Press the Browse button.
2. Standard Open screen is displayed.
3. User choose the input file which PONumber tag is missing;
4. Press Submit button

Result: The process is complete and Prompt message appeared with Error message “PONumber tag is empty or not exist”

**Use Case 4:** Select an incorrect input file

Primary Actor: User

Pre Condition: Application is start and prompt window to choose file is released

Main Scenario:

1. Press the Browse button.
2. Standard Open screen is displayed.
3. User choose the input file which PONumber not include a “-” delimiter;
4. Press Submit button

Result: The process is complete and Prompt message appeared with Error message “No Contract number available in PONumber”

### 