

---

## INTRODUCTION

The purpose of this document is to define scope and requirements of a web-based Regular Expression Editor for the developers of a software services company. The proposed system will aid the developers in rapid development of regular expressions. Currently, they test the regular expression as part of the module thereby making the development & testing more complex.

This document is the primary input to the development team to architect a solution for this project.

### System Users

The entire team of developers of the software services company is expected to benefit from the regular expression editor, REX.

### Assumptions

1. REX will be integrated with the existing company's intranet and therefore it will leverage the existing Intranet's authentication mechanism.
2. It is assumed that the standard `java.util.regex` package will be used for the development.

## REQUIREMENTS

REX will allow users to easily create, edit and test regular expressions. The user interface for the same is outlined below.

The screenshot shows a web-based interface titled "REX - Regular Expression Editor". It features a text input field at the top for entering a regular expression. Below this are three checkboxes: "Case Insensitive" (checked), "Dot Matches Newlines" (unchecked), and "Ignore White-spaces" (unchecked). To the left is a larger text area for entering a test string. On the right, there are two output fields: "Match Result" and "Match Groups". At the bottom, there are two buttons: "Clear" and "Run".

The top most field will accept the regular expression from the user. The check boxes will allow selection of various options for the regex entered. The user will be able to enter the test string on the left hand side text box. Clicking "Run" button, will execute the regex with selected options on the test string and show the results in the text boxes on the right hand side. The matched results are shown highlighted in a light blue color. The matched groups are shown in multiple lines – one group per line. A sample screen of such a run is shown

below.

The screenshot shows the 'REX - Regular Expression Editor' web application. At the top, the title 'REX - Regular Expression Editor' is displayed in blue. Below the title is a text input field containing the regular expression `(?<month>\d{1,2})\V(?<day>\d{1,2})\V(?<year>\d{4})`. Underneath the input field are three checkboxes: 'Case Insensitive' (checked), 'Dot Matches Newlines' (unchecked), and 'Ignore White-spaces' (unchecked). The main area is divided into two columns. The left column contains a text area with the text 'Today's date is: 5/28/2012.'. The right column contains two sections: 'Match Result' and 'Match Groups'. The 'Match Result' section shows the text 'Today's date is: 5/28/2012 .' with the date '5/28/2012' highlighted in blue. The 'Match Groups' section displays the following information: month 5, day 28, and year 2012. At the bottom of the interface are two buttons: 'Clear' and 'Run'.

Clicking “Clear” Button will clear the form and it will be ready for another regex editing and/or testing.

### Optional Requirements

It is highly desirable to provide a capability to save tested regex in to a library, so that they may be reused in future. In this case, user will be able to save each regex with a “title” and a “description outlining its purpose”. A list view of all saved regex will be available for browsing the library.

Since REX is expected to use Intranet’s authentication, for the purpose of this project, entering user name will take you to the user’s REX screen. You may create sample users directly from the backend database.

## DEVELOPMENT ENVIRONMENT

REX will be developed as a web application using Java/JSP and DB2 database. Eclipse will be used as the IDE for the same. You may consider using a JavaScript framework like jQuery/Prototype/ Scriptaculous.