

SMS search based on content or sender Ver-1 : Web Runtime

Introduction

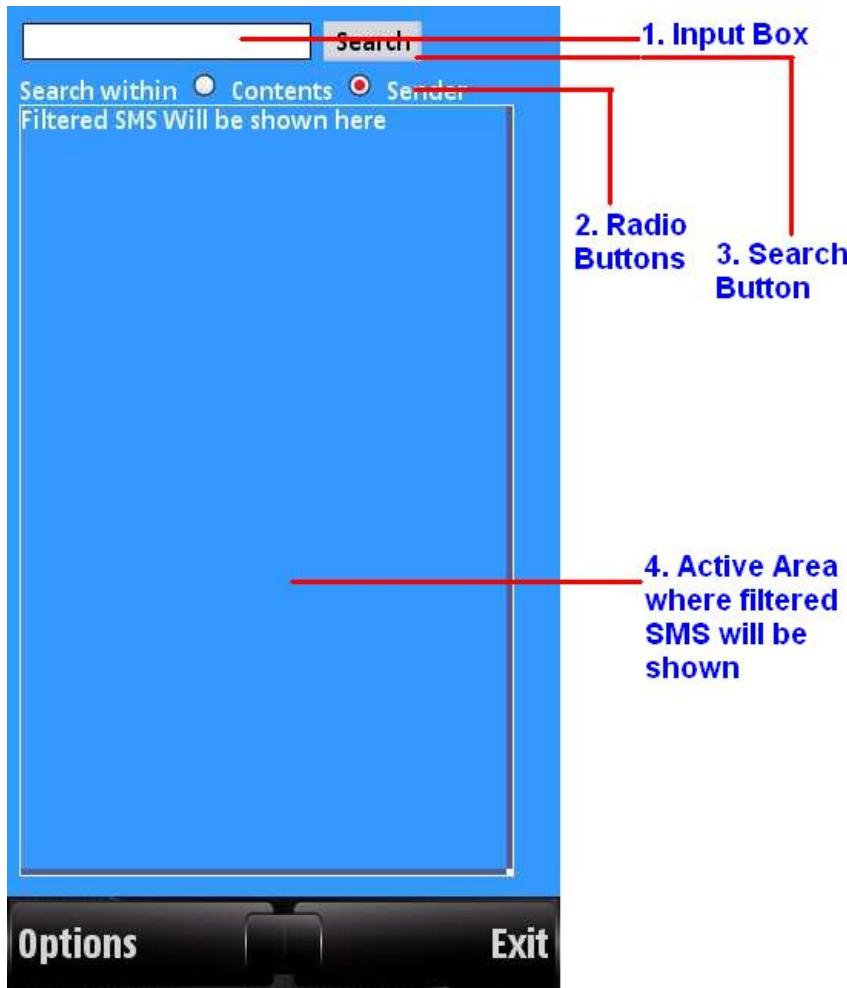
This application/article is useful to those who are frequent user of SMS. The application is capable of following:

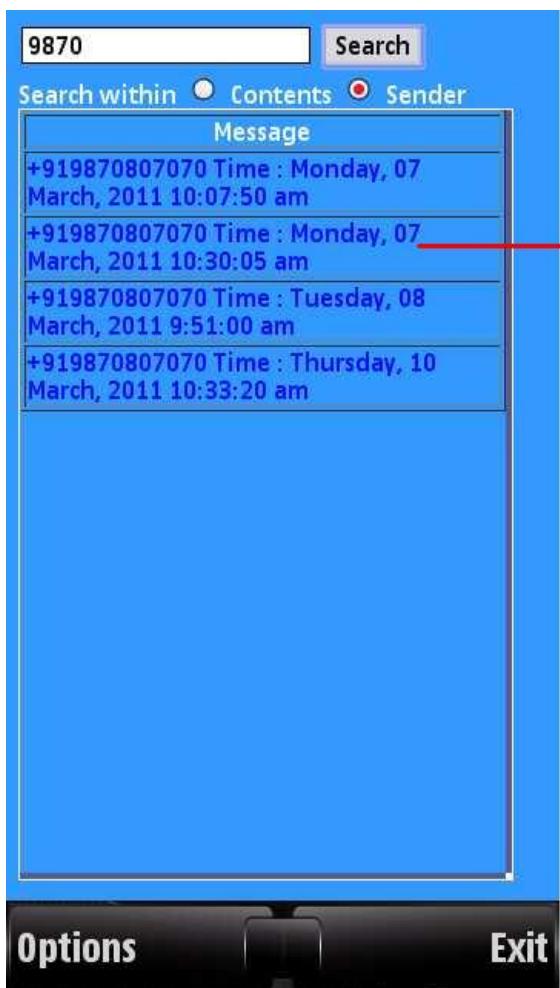
- You can search SMSes based on the name/number of sender
- You can search SMSes based on the contents/word within the SMS

How to use ?

As shown in the application figure-1 below, the application is divided into the following parts.

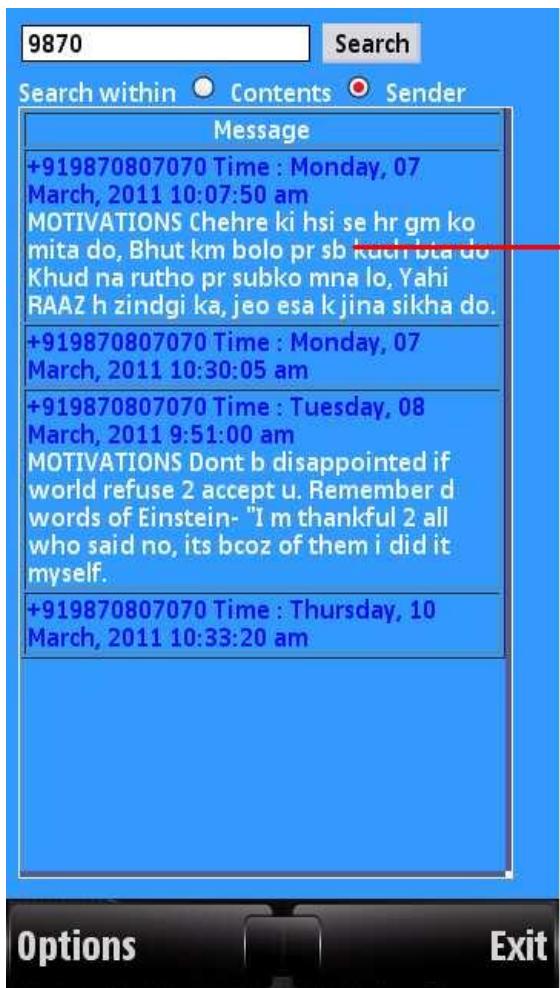
- **(1) An Input box to search string :** In this box, you can give a sender's name / number. The string provided within the input box will be used to search into SMS
- **(2) A radio button to select search criteria:** You can select either **Contents** or **Sender** radio button. By default the "Sender" radio button is selected. The application will search for the sender if **Sender** radio button is selected. The application will search within the SMS contents if the **Contents** Radio button is selected.
- **(3) Search Button :** As soon as you click on search button, the application will search the SMSes based on the inputs provided. (i.e. search string & search within) It will show name/number of sender with date and time of SMS (say title of SMS) into the active area of application. You can see this in the figure-2
- **(4) Active Area :** This is the area where filtered SMS will be shown. By default you can see name/number along with SMS date and time of sender as a title of SMS. You can click on the title and the application will show full SMS just below it. Clicking back to the same title will hide the SMS. Figure-3 illustrates this.





SMSe are
filtered based on
'sender' whose
number is like
'9870'

Figure-2



You can see the
SMS text
just below
the title once you
touch(click) on
the SMS title.

Figure-3

How to create project

Please download and install the Nokia Web SDK. Create a new project named SmsNameFilter in it. Replace the files, SmsNameFilter.html, SmsNameFilter.css and SmsNameFilter.js with the code below.

SmsNameFilter.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>SMS Name Filter Ver 1</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<script language="javascript" type="text/javascript" src="SmsNameFilter.js"></script>
<link rel="stylesheet" href="SmsNameFilter.css" type="text/css" />
<meta name="Generator" content="Nokia WDE 3.0.0" />
</head>
<body>

<div id="inputpanel">

<input type="text" name="search" id="search" />

<input type="button" id="button" onclick="filterSMS();" value="Search"/>
<br />
Search within
<input type="radio" id="radio1" name="radio" value="contents" /> Contents
<input type="radio" id="radio2" name="radio" value="sender" checked="checked" />
Sender
</div>
<div id="SMSlog">Filtered SMS Will be shown here</div>
</body>
</html>
```

SmsNameFilter.css

```
/* applies to the id 'mainpanel' */
#mainpanel {
    width:320px; /* Define width */
    height:40px; /* Define height */
    border:1px solid; /* Define border style*/
}

/* applies to the id 'SMSlog' */
#SMSlog {
    width:320px; /* Define width */
    height:500px; /* Define height */
    border:1px solid; /* Define border style*/
    overflow:scroll; /* The contents will be scrolled to fit into the screen */
```

```
}

/* applies to the html tag label */

label{
  color:#0000FF; /* Defining font color for the label */
}
/* The following CSS will work only if CSS3 is supported by mobile */
/* This style sheet differentiates even and odd rows. */
#smstable tr:nth-child(even) {
  background-color: #EEEEEE;
  border-bottom: 2px #006600 solid;
}

/* Setting background color, font color of the body */

body {
  font-weight: bold;
  color: #FFFFFF;
  background-color: #3399FF;
}
```

SmsNameFilter.js

```
/*
 * JavaScript file
 */

// Variable so is defined as global variable

var so;

// Function hide and show will show or hide the SMS body text.

function hideshow(mytext)
{
  var el = document.getElementById(mytext);
  // style.display property has two values.
  // 1. Block will show the contents and 2. None will hide the contents

  if(el.style.display != 'block')
  {
    el.style.display = 'block';
  }
  else
  {
    el.style.display = 'none';
  }
}

function filterSMS()
{
```

```
try
{
    so = device.getServiceObject("Service.Messaging", "IMessaging");
} catch(e ) {
    alert("Error in getting Messaging interface \n" + e);
    return;
}
document.getElementById('SMSlog').innerHTML = "<img src='loading.gif' />";

// Our criteria is to get inbox SMS messages.
var criteria = {
    // Specifies the type of messaging objects to retrieve with values
    // like "Inbox", "NewMessage" etc..
    Type : "Inbox",
    Filter :{
        //Specifies the type(s) of messages to retrieve from "SMS", "MMS" & "unknown"
        MessageTypeList : ["SMS"]
    },
    //The Sort object element used to sort the sms inbox contents.
    //But it seems that the sort feature not supporting yet.
    Sort :{
        Key : "Date",
        Order: "Descending"
    }
};

// Lets get the messages with the above prepared criteria, synchronously.
var result = so.IMessaging.GetList(criteria);

// Lets print the received message list in html page.
var items = result.ReturnValue;
var message = "";
var list = "";
var sender;
// Get the value entered into search string and convert into upper case.
var searchString = document.getElementById('search').value.toUpperCase();
if(document.getElementById('radio1').checked)
    var searchContent = document.getElementById('radio1').value.toUpperCase();
if(document.getElementById('radio2').checked)
    var searchContent = document.getElementById('radio2').value.toUpperCase();

list = "<table border=1 id=smstable align='center' width=100%
<tr><th>Message</th></tr>";
while (( message = items.getNext()) != undefined )
{
    //converting the name of sender into upper case.
    sender = message.Sender.toUpperCase();

    //Comparing the sender with the search string.
    if (searchContent == "SENDER")
    {
        if (sender.search(searchString) != -1) {
            list += "<tr>";
            list += "<td><label id='sms' onclick='hideshow(" + message.MessageId + ")';>" +
http://developer.nokia.com/community/wiki/SMS_search_based_on_content_or_sender_Ver-1:_Web_Runtime                (C) Copyright Nokia 2014. All rights reserved.
```

```
message.Sender +
" Time : " +
message.Time +
"</b></label>";
list += "<div id = '" + message.MessageId + "' style ='display:none'>" +
message.BodyText + "</div></td>";
list += "</tr>";
}
}
else
{
contents = message.BodyText.toUpperCase();
if (contents.search(searchString) != -1)
{
list += "<tr>";
list += "<td><label id='sms' onclick='hideshow(" + message.MessageId + ")';>" +
message.Sender +
" Time : " +
message.Time +
"</b></label>";
list += "<div id = '" + message.MessageId + "' style ='display:none'>" +
message.BodyText + "</div></td>";
list += "</tr>";
}
}

}
list += "</table>";
document.getElementById('SMSlog').innerHTML = list;
}
```

Tested On

Nokia 5230.

Created Using

Nokia Web SDK 1.0.1

Other Images

The following figure-4 shows the SMSes are filtered based on contents of SMS.



SMSe are
filtered based
on the
contents which
contains the
word "God"
within it

Figure-4

What next ?

This is the first version of application. The application is under more enhancement and the other versions will be released soon.

Download Source Code of Application

[File:SmsNameFilter Ver1.zip](#)

Extra Information

I have used the **Active Area** as a scrollable control. The Scrollable Control using WRT has been discussed in my previous article [Creating Scrolling Control on Mobile Screen](#)

