

LIBRARY SERVICE CATALOG TASK

INTRODUCTION

Catalog Item to provide Books based on categories.

User cannot take more than 2 books.

Sending Notifications on 15, 20 and 25 days to the user a reminder.


Lending Period: 30 days

[Home](#) > [Service Catalog](#) > [Office](#) > [Services](#) > **Library Service**

Search Catalog

Library Service

Book Borrowing Service



Welcome to Books borrowing services. Please select the category and the books.
Description of the book will be provided on selecting the book along with the returning due date. Lending period is for 30 days. Please make sure to return the previously borrowed books before lending new ones. User can request 2 books at a time.

Happy Reading 😊📖

* Indicates required

* User

System Administrator

* Select Category

-- None --

Mobile Number

1209394950

Email

admin@example.com

* User

System Administrator

* Select Category

Poems

* Select Book

Leaves of Grass

✕

Book Description

A collection of poetry which Whitman famously self-published and is known for its focus on democracy, nature, love, and friendship.

Due Date

2024-06-14

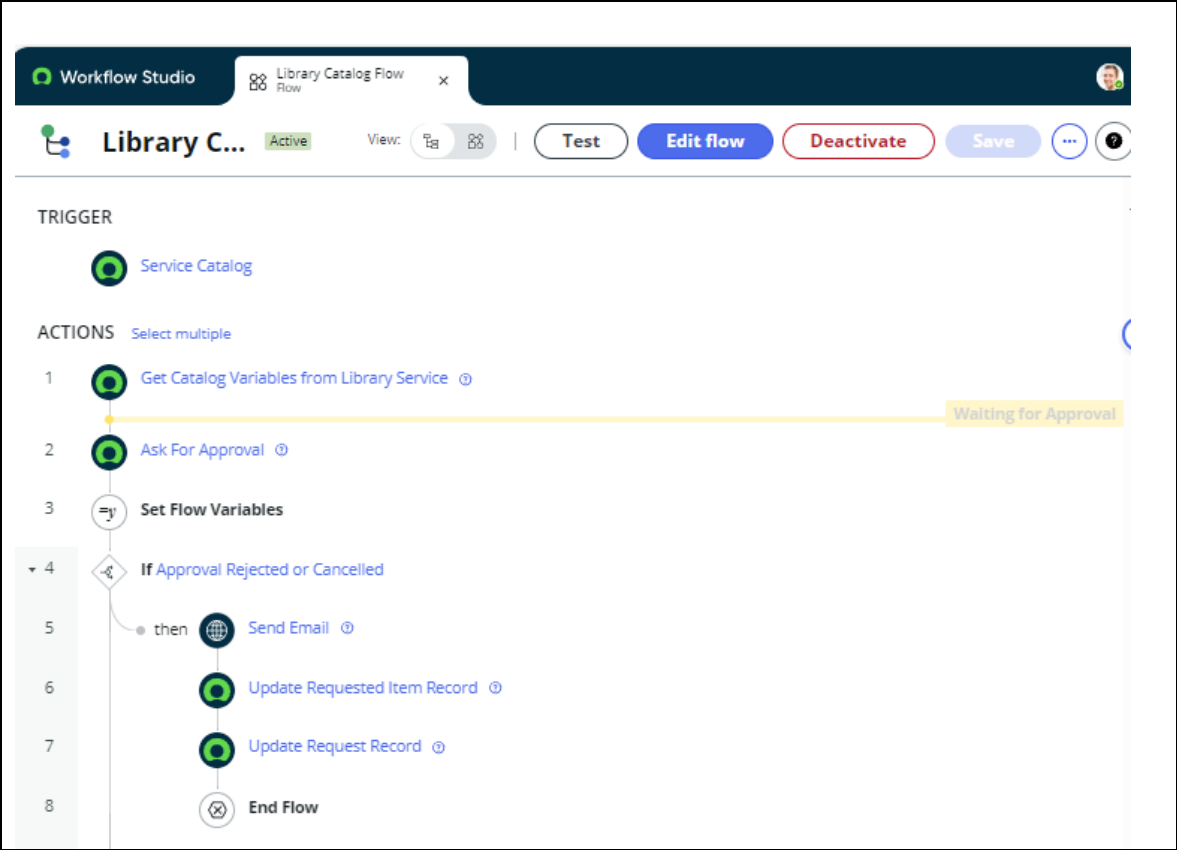
Mobile Number

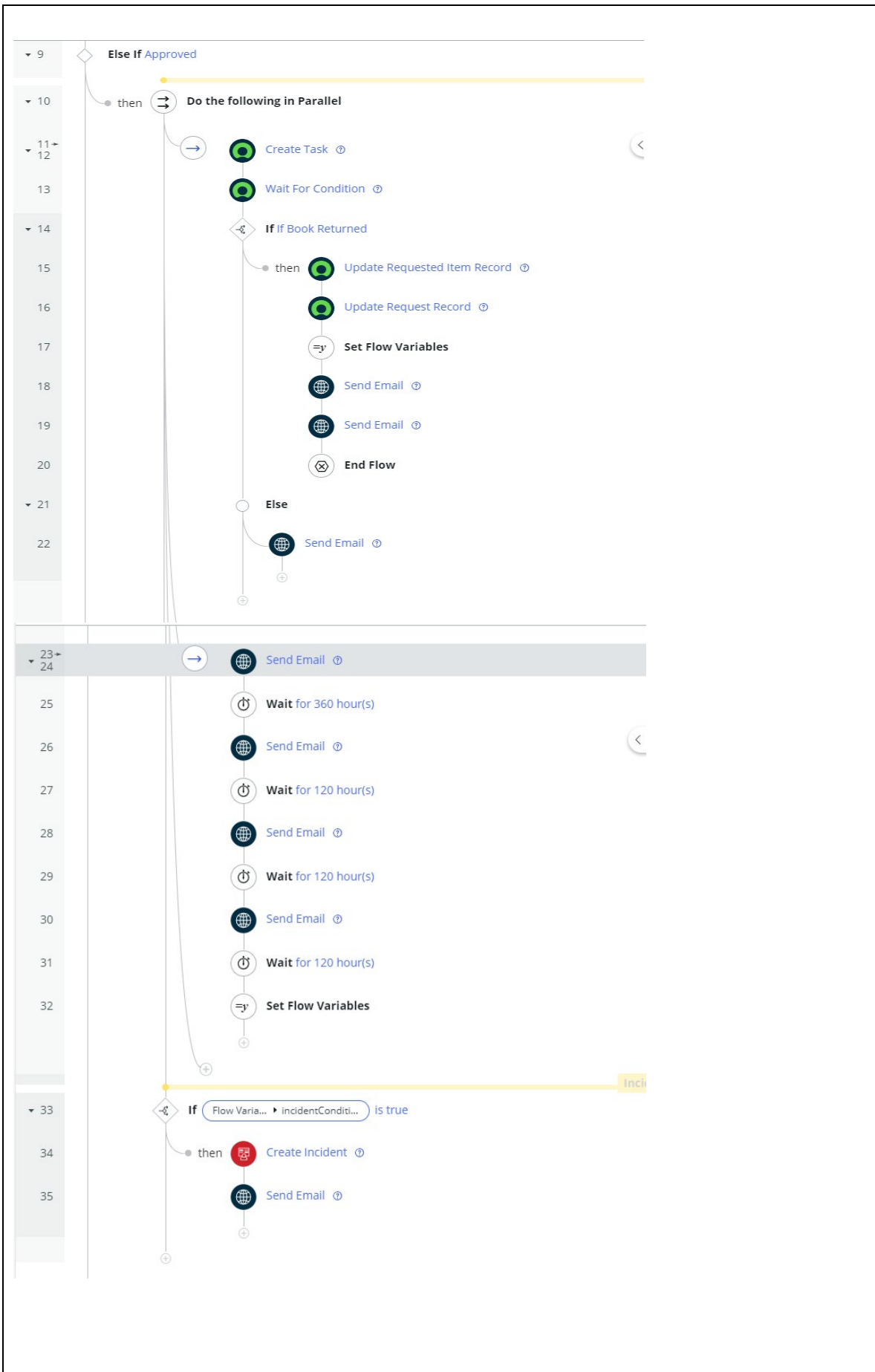
1209394950

Email

admin@example.com

FLOW DESIGN





SCRIPT INCLUDES

GetUserContactInfo

To fetch email and mobile number of the user.

```
var GetUserContactInfo = Class.create();
GetUserContactInfo.prototype = Object.extend(Object(AbstractAjaxProcessor, {
  getMailMobile:function(){
    var userID = this.getParameter('sysparm_user_id');
    var userGR = new GlideRecord('sys_user');

    userGR.get(userID);
    //gs.log("Inside IF BLOCK");
    var userInfo = {
      "mobile": userGR.getValue('mobile_phone'),
      "email": userGR.getValue('email')
    };

    gs.log(userInfo);

    return JSON.stringify(userInfo);
  },
  type: 'GetUserContactInfo'
});
```

LibraryBookOrderCondition

Check whether the User already has 2 Active Library Service Request

Only 2 books can be provided at a time.

```
var LibraryBookOrderCondition = Class.create();
LibraryBookOrderCondition.prototype =
Object.extend(Object(AbstractAjaxProcessor, {

  checkCondition:function(){
    var userID = this.getParameter('sysparm_userid');

    var gr = new GlideRecord('sc_cart_item');
    gr.addQuery('cat_item', '887bf985932102106e7bf15efaba1008');
    gr.query();
    var total_lib_books = 0;
    while(gr.next()){
      //gs.log(gr.quantity);
      total_lib_books += gr.quantity;
    }
  }
});
```

```

        var grReq = new GlideRecord('sc_req_item');
        grReq.addEncodedQuery('stateIN-
5,1,2^cat_item=887bf985932102106e7bf15efaba1008^requested_for='+ userID + '');
        grReq.query();

        return grReq.getRowCount() + "," + total_lib_books;

    },

    type: 'LibraryBookOrderCondition'
});

```

CLIENT SCRIPTS

Show Order Button

To hide the Order Button if the user already has two books.

```

function onChange(control, oldValue, newValue, isLoading) {
    if (newValue == '') {
        return;
    }
    if (newValue == 'Yes') { // hide submit button
        if (window == null) { // service portal
            var z = this.document.getElementsByClassName("btn btn-primary btn-
block ng-binding ng-scope");
            z[0].style.display = 'none';
        } else { // ui platform
            document.getElementById("oi_order_now_button").style.display =
'none';
        }
    } else { // display submit button
        if (window == null) { // service portal
            var y = this.document.getElementsByClassName("btn btn-primary btn-
block ng-binding ng-scope");
            y[0].style.display = 'initial';
        } else { // ui platform
            document.getElementById("oi_order_now_button").style.display =
'initial';
        }
    }
}

```

Check for Cart & Request for Books

Show error message.

Initiate hiding `Order Button`

```
function onLoad() {
    var ga = new GlideAjax('LibraryBookOrderCondition');
    ga.addParam('sysparm_name', 'checkCondition');
    ga.addParam('sysparm_userid', g_user.userID);
    ga.getXMLAnswer(callBackFunction);
}

function callBackFunction(response){
    var data = response.split(",");
    var reqs = parseInt(data[0]);
    var carts = parseInt(data[1]);

    if(reqs>=2){

        g_form.setReadOnly('select_book', true);

        alert("Please return the previously borrowed books before ordering new books");

        g_form.addErrorMessage("Only 2 books can be borrowed at a time, Return previous books before borrowing new ones");
        g_form.disable();
        g_form.setReadOnly('select_book', true);

        return false;
    }
    else if(carts == 2 && reqs == 1){
        g_form.setReadOnly('select_book', true);
        alert("Only 2 books can be taken at a time");
        g_form.addErrorMessage("Return one book or remove one book from cart and Refresh");
        g_form.setReadOnly('select_book', true);

        return false;
    }
    else if(carts>= 2){
        g_form.setReadOnly('select_book', true);

        alert("Please remove books from cart");
        g_form.addErrorMessage("Only 2 books can be borrowed at a time, Remove books from the cart and Refresh");
        g_form.setReadOnly('select_book', true);
    }
}
```

```

        return false;
    }
    else{
        var t = reqs + carts;
        alert("Total Library Book Requests: " + t);
        //g_form.submit();
    }
}

```

Validate Mobile Number

```

function onChange(control, oldValue, newValue, isLoading) {
    if (isLoading || newValue == '') {
        return;
    }

    if(isNaN(newValue)){
        g_form.showFieldMsg('mobile_number', "Enter only Numeric Data");
        return false;
    }
    else if(newValue.length != 10){
        //alert(newValue.length);
        g_form.showFieldMsg('mobile_number', "Mobile must be of 10
digits");
        return false;
    }
    else{
        return true;
    }

    //Type appropriate comment here, and begin script below
}

```