// Modified from http://pipetree.com/qmacro/blog/2011/10/automated-email-to-task-mechanism-with-google-apps-script/

// Globals, constants

var LABEL\_PENDING = "pending\_requirement";

var LABEL\_DONE = "done\_requirement";

var parentFolder = DriveApp.getFolderById('1FGOsBXofYYcsJC0nqtIzHbfAslhLK-P2')

var fileTypesToExtract = ['jpg', 'jpeg', 'tif', 'png', 'gif', 'bmp', 'svg'];

// processPending(sheet)

// Process any pending emails and then move them to done

function processPending\_(sheet) {

  // Get out labels by name

  var label\_pending = GmailApp.getUserLabelByName(LABEL\_PENDING);

  var label\_done = GmailApp.createLabel(LABEL\_DONE);

  // The threads currently assigned to the 'pending' label

  var threads = label\_pending.getThreads();

  // Process each one in turn, assuming there's only a single

  // message in each thread

  for (var t in threads) {

    // Read the last SRID

    lr = sheet.getLastRow();

    lc = sheet.getLastColumn();

    if (lr > 1) {

    rid\_cell\_hist = sheet.getRange(lr, 1);

    rid\_cell = sheet.getRange(lr+1, 1);

    rid = rid\_cell\_hist.getValue()+1;

    rid\_cell.setValue(rid).setNumberFormat("000000");

    } else {

    rid\_cell = sheet.getRange(lr+1, 1);

    rid = 1;

    rid\_cell.setValue(rid).setNumberFormat("000000");

    }

    var thread = threads[t];

    // Gets the message body

    var message = thread.getMessages()[0];

    var content = thread.getMessages()[0].getPlainBody();

    var attachments = message.getAttachments();

    // TODO: Process the messages here

    // Add message to sheet

    date\_cell = sheet.getRange(lr+1, 2)

    date\_cell.setValue(message.getDate());

    username = message.getFrom();

    from\_cell = sheet.getRange(lr+1, 3)

    from\_cell.setValue(username);

    regExp = new RegExp("(?<=" + "Customer:" + ").\*");

    customer = content.match(regExp);

    if (customer === null){

      customer = '';

    } else {

      customer = customer.toString().trim();

      customer\_cell = sheet.getRange(lr+1, 4);

      customer\_cell.setValue(customer);

    }

    regExp = new RegExp("(?<=" + "Requirement:" + ").\*");

    requirement = content.match(regExp);

    if (requirement === null){

      customer = '';

    } else {

    requirement = requirement.toString().trim();

    requirement\_cell = sheet.getRange(lr+1, 5);

    requirement\_cell.setValue(requirement);

    }

    regExp = new RegExp("(?<=" + "Purpose:" + ").\*");

    purpose = content.match(regExp);

    if (purpose === null){

      customer = '';

    } else {

    purpose = purpose.toString().trim();

    purpose\_cell = sheet.getRange(lr+1, 6);

    purpose\_cell.setValue(purpose);

    }

    i = 0;

    var root = DriveApp.getRootFolder();

    for(var k in attachments){

      var attachment = attachments[k];

      var isDefinedType = checkIfDefinedType\_(attachment);

      if(!isDefinedType) continue;

      var attachmentBlob = attachment.copyBlob();

      var file = DriveApp.createFile(attachmentBlob).setName('SR\_'+rid+'-'+k+'\_'+username);

      parentFolder.addFile(file);

      stored\_file = DriveApp.getFolderById('1FGOsBXofYYcsJC0nqtIzHbfAslhLK-P2').getFilesByName(file);

    // Get the id of the saved attachment

    while (stored\_file.hasNext()) {

      var file = stored\_file.next();

      file.setSharing(DriveApp.Access.ANYONE\_WITH\_LINK, DriveApp.Permission.VIEW);

      image\_file\_id = file.getId();

    };

    // Generate the image link and store the value

    image\_link = 'https://drive.google.com/file/d/'+image\_file\_id+'/view';

    attachment\_cell = sheet.getRange(lr+1, i+7);

    attachment\_cell.setValue(image\_link);

    root.removeFile(file);

    i += 1;

    }

    thread.removeLabel(label\_pending);

    thread.addLabel(label\_done);

    if(i>0){

      var attachment\_Status = "Yes";

    } else {

      var attachment\_Status = "No";

    }

    message.replyAll("",{

    htmlBody: "Hello,<br/>We have recieved your requirement with the below details:"

    +"<br/><b>Customer</b>: "+customer+"<br/><b>Requirement</b>: "+requirement+"<br/><b>Purpose</b>: "+purpose

    +"<br/><b>Attachment</b>: "+attachment\_Status

    +"<br/><br/><b>RID</b>: "+ Utilities.formatString("%06d", rid)+"<br/> Please save this RID (Requirement Id) for any future correspondence."

    +"<br/><br/>Now you please sit back, relax and enjoy your day as the team has started to cook your requirement right away..!"

    +"<br/><br/>Regards,<br/>SPEG Team",

    noReply: true

    })

  }

}

function checkIfDefinedType\_(attachment){

  var fileName = attachment.getName();

  var temp = fileName.split('.');

  var fileExtension = temp[temp.length-1].toLowerCase();

  if(fileTypesToExtract.indexOf(fileExtension) !== -1) return true;

  else return false;

}

// main()

// Starter function; to be scheduled regularly

function main\_emailDataToSpreadsheet() {

  // Get the active spreadsheet and make sure the first

  // sheet is the active one

  var ss = SpreadsheetApp.getActiveSpreadsheet();

  var sh = ss.setActiveSheet(ss.getSheets()[0]);

  // Process the pending emails

  processPending\_(sh);

}