Staff Management Utility

Software Developer Manual

Developed by:
Aditya Chourasia (a1779498),
Junzhe Huang (a1780577),
Mingyi Li (a1760412),
Minh Anh Hoang (a1776251),
Yen You Chew (a1686965),
Qianyang Tang (a1733483),
Tianyuan Li (a1775656),
Weiling Zeng (a1784523),
Ziqi Xu (a1767092).

Table of Contents

1. General Information

1.1 Purpose

The primary purpose of this document is to provide information to developers at Maptek to endorse further development, maintenance, hosting sufficiently, and use of Staff Management Utility (SMU). This document explains the functional requirement and provides the high-level implementation of code. This document also explains the roles and responsibilities required for a developer to install SMU and the minimum system requirement. This document does not include the guideline or instruction about Google Sheet API. It is assumed that the developer team is familiar with Google Sheet API, JavaScript, HTML, and CSS.

1.2 Scope

This Functional and Technical Requirements Document outlines the functional and implementation of scripts or plugins developed in JavaScript that can be added to Google Spreadsheet to facilitate the end-user and developers team at Maptek with various features.

Scripts developed is supposed to

- **Compatibility:** script file or code must support the existing Google Spreadsheet.
- **Search:** an efficient and detailed search script with the capability of searching employees whenever search conditions are met, in the existing spreadsheet.
- **Organizational chart:** script should be able to generate an organizational chart. The chart can be of the entire organization or the individual section.
- **Report:** the script must be able to generate a report on the entire organization or the subsection. The report must contain male to female ratios and numbers of members in each division.

- **Email:** the script should be able to send emails to filtered or searched individuals.
- **Email Content:** the script must support a working email template and can replace parameters with specific information.

1.3 Acronyms and Definitions

API	Application Programming Interface, a set of protocols or standards for communicating with web-based applications
Community	Maptek member institutions
CSS3	Cascading Style Sheets; language used to describe the presentation of a document written in a markup language, e.g., HTML
Git version control	Free and open-source version control system
Google Spreadsheet	Online application by Google for managing data and information in tabular form or sheets.
HTML 5	Hypertext Markup Language; the fifth and current version of the HTML standard
JavaScript	Programming language used extensively in website development
jQuery for JavaScript	JavaScript library
SMU	Staff management Utility
Sheet	An individual table or page or a spreadsheet file.

2. Current System Summary

The current system is capable of generating reports, performing search functions, and generating organizational graphs. The current system has scripts files developed by JavaScript that can be added to an existing spreadsheet. Once the scripts files are added to the sheet, scripts menus are created automatically, and an end-user can use those scripts simply by clicking the specific option.

3. System Requirements

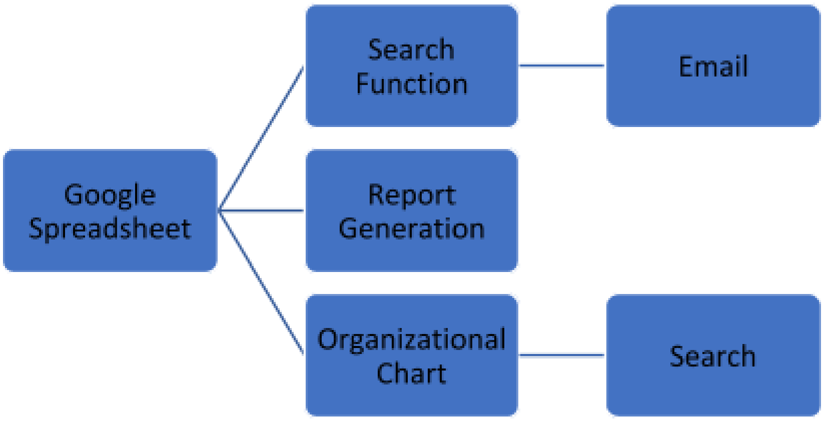
Software	Platform	Version
Chrome Browser	Windows or Linux	86.x.x
JavaScript	Windows or Linux	ECMAScript 2018
Node.js	Windows or Linux	12.x.x
Google Spreadsheets	Windows or Linux	4.x.x

Note: The user also needs to have a Google account with privileges to edit the spreadsheet.

4. Architectural Design

Microkernel architectural patterns or plugin architectural patterns are preferred to fulfill the required functionality. One of the biggest advantages of this architecture is that each plugin developed is independent and runs independently. Another advantage is that it would include less code with minimum bugs and fails.

Plugins like search, report generation, and organizational chart are built on the google spreadsheet. Each plugin is developed independently, and the change or upgrade on one will not affect others.



5. Installation

1. Create a new blank Google spreadsheet and open the ‘script editor’ in ‘Tools (Figure 1).

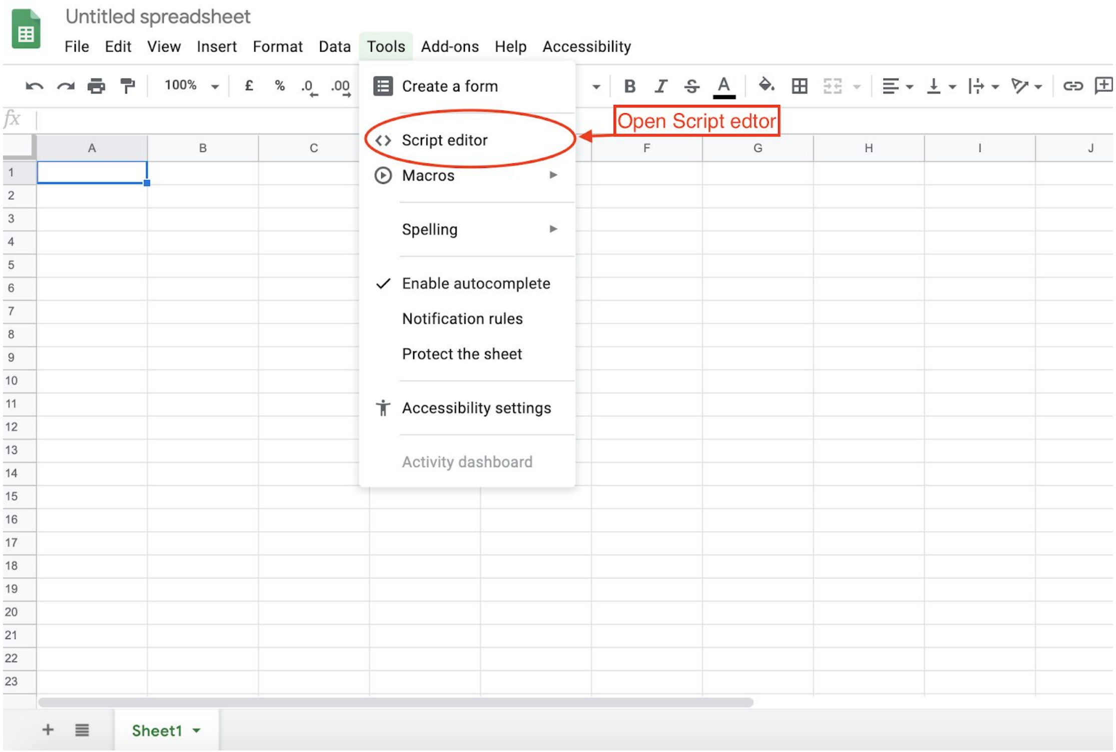
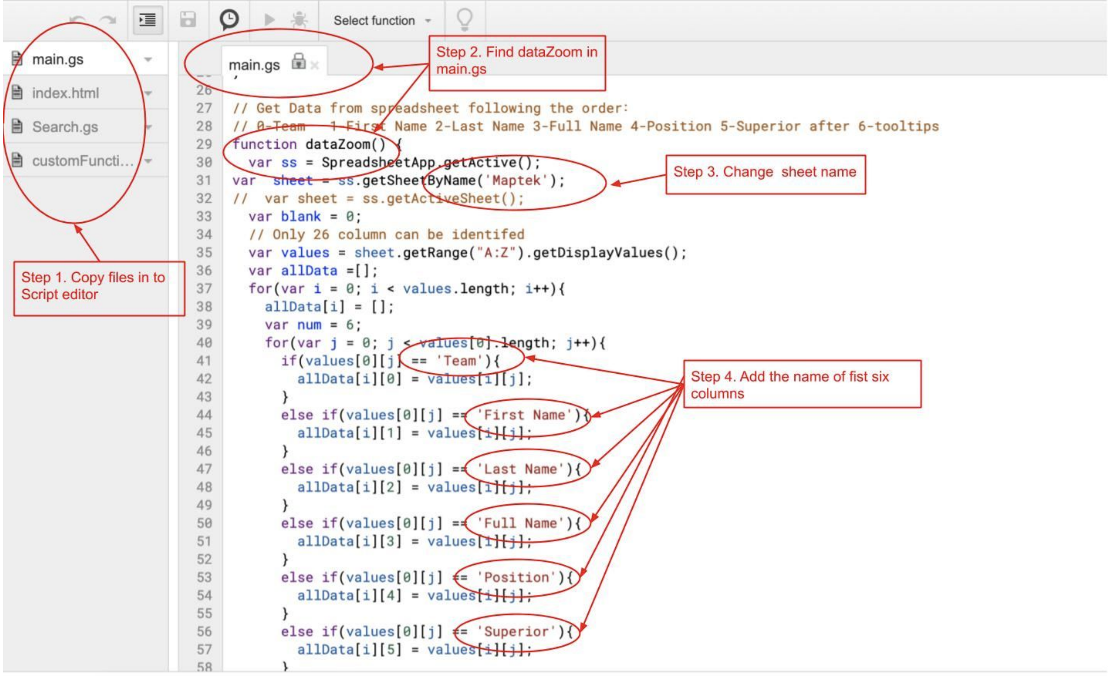


Figure 1

2. Copy files into ‘Script editor’ (Step 1 in Figure 2).

3. Find the function ‘dataZoom’ in file ‘main.gs’(Step 2 in Figure 2).



Step 1. Copy files in to Script editor

Step 2. Find dataZoom in main.gs

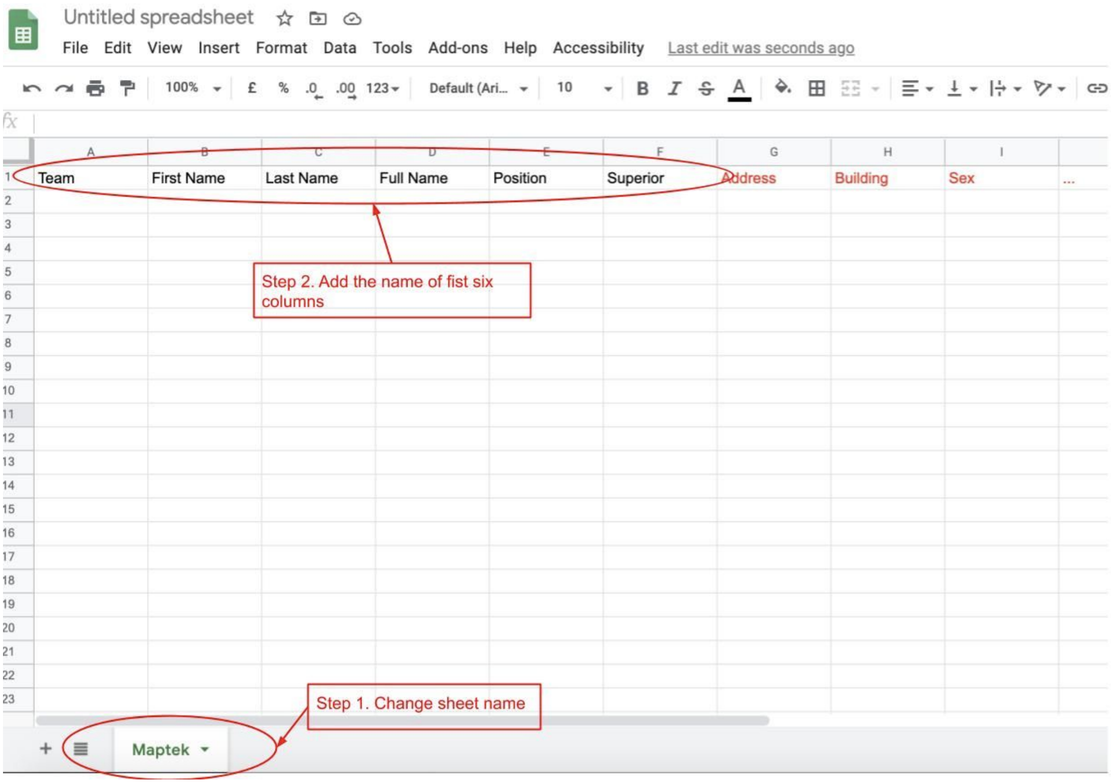
Step 3. Change sheet name

Step 4. Add the name of fist six columns

```
// Get Data from spreadsheet following the order:
// 0-Team   1-First Name 2-Last Name 3-Full Name 4-Position 5-Superior after 6-tooltips
function dataZoom() {
  var ss = SpreadsheetApp.getActive();
var  sheet = ss.getSheetByName('Maptek');
//  var sheet = ss.getActiveSheet();
  var blank = 0;
  // Only 26 column can be identifed
  var values = sheet.getRange("A:Z").getDisplayValues();
  var allData =[];
  for(var i = 0; i < values.length; i++){
    allData[i] = [];
    var num = 6;
    for(var j = 0; j < values[0].length; j++){
      if(values[0][j] == 'Team'){
        allData[i][0] = values[i][j];
      }
      else if(values[0][j] == 'First Name'){
        allData[i][1] = values[i][j];
      }
      else if(values[0][j] == 'Last Name'){
        allData[i][2] = values[i][j];
      }
      else if(values[0][j] == 'Full Name'){
        allData[i][3] = values[i][j];
      }
      else if(values[0][j] == 'Position'){
        allData[i][4] = values[i][j];
      }
      else if(values[0][j] == 'Superior'){
        allData[i][5] = values[i][j];
      }
```

Figure 2

4. Change the sheet name to what you want to store the data of this project in both function ‘dataZoom’(Step 3 in Figure 2) and sheet (Step 1 in Figure 3), but make sure they are exactly the same.



	A	B	C	D	E	F	G	H	I	
1	Team	First Name	Last Name	Full Name	Position	Superior	Address	Building	Sex	...

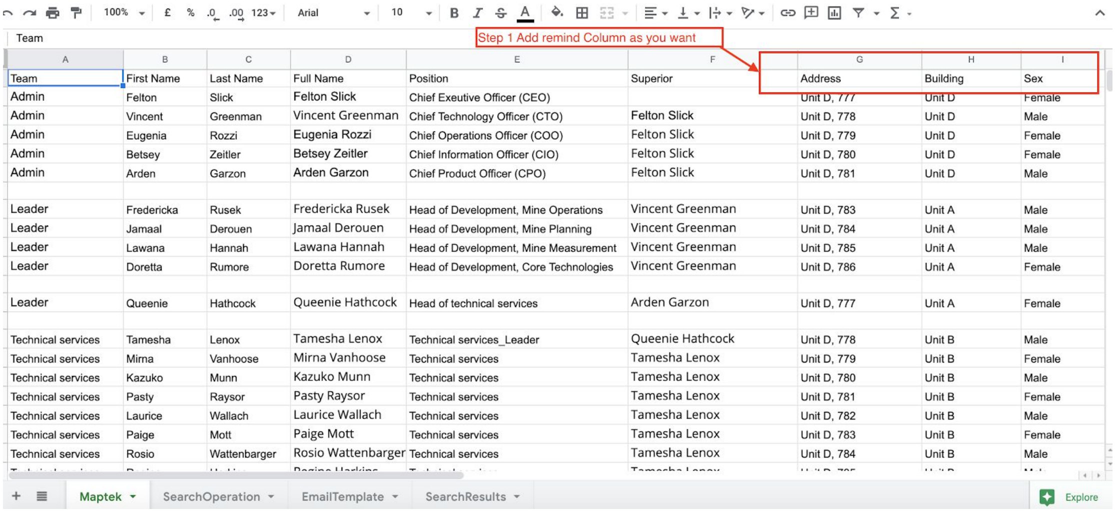
Step 2. Add the name of fist six columns

Step 1. Change sheet name

Figure 3

5. Add the name of the first six columns into the sheet (Step 2 in Figure 3) based on the data in function 'dataZoom' (Step 4 in Figure 2) the names of the Columns can be changed, but make sure they are exactly the same.

6. Add reminder data. You can add other columns as you want (Step 1 in Figure 4).



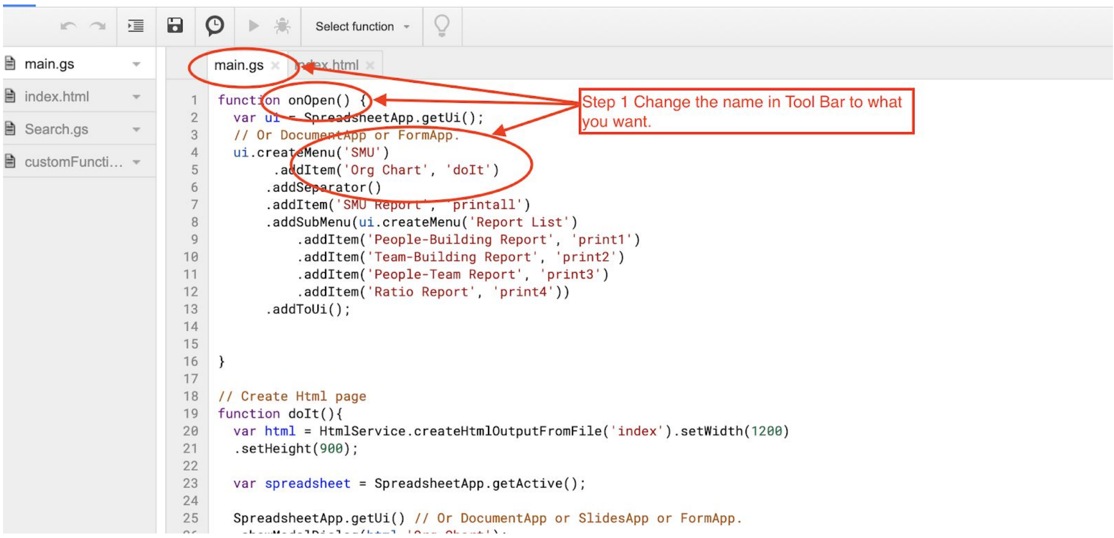
Team

Step 1 Add remind Column as you want

A	B	C	D	E	F	G	H	I
Team	First Name	Last Name	Full Name	Position	Superior	Address	Building	Sex
Admin	Felton	Slick	Felton Slick	Chief Exeutive Officer (CEO)		Unit D, 777	Unit D	Female
Admin	Vincent	Greenman	Vincent Greenman	Chief Technology Officer (CTO)	Felton Slick	Unit D, 778	Unit D	Male
Admin	Eugenia	Rozzi	Eugenia Rozzi	Chief Operations Officer (COO)	Felton Slick	Unit D, 779	Unit D	Female
Admin	Betsey	Zeitler	Betsey Zeitler	Chief Information Officer (CIO)	Felton Slick	Unit D, 780	Unit D	Female
Admin	Arden	Garzon	Arden Garzon	Chief Product Officer (CPO)	Felton Slick	Unit D, 781	Unit D	Male
Leader	Fredericka	Rusek	Fredericka Rusek	Head of Development, Mine Operations	Vincent Greenman	Unit D, 783	Unit A	Male
Leader	Jamaal	Derouen	Jamaal Derouen	Head of Development, Mine Planning	Vincent Greenman	Unit D, 784	Unit A	Male
Leader	Lawana	Hannah	Lawana Hannah	Head of Development, Mine Measurement	Vincent Greenman	Unit D, 785	Unit A	Male
Leader	Doretta	Rumore	Doretta Rumore	Head of Development, Core Technologies	Vincent Greenman	Unit D, 786	Unit A	Female
Leader	Queenie	Hathcock	Queenie Hathcock	Head of technical services	Arden Garzon	Unit D, 777	Unit A	Female
Technical services	Tamesha	Lenox	Tamesha Lenox	Technical services_Leader	Queenie Hathcock	Unit D, 778	Unit B	Male
Technical services	Mirna	Vanhoose	Mirna Vanhoose	Technical services	Tamesha Lenox	Unit D, 779	Unit B	Female
Technical services	Kazuko	Munn	Kazuko Munn	Technical services	Tamesha Lenox	Unit D, 780	Unit B	Male
Technical services	Pasty	Raysor	Pasty Raysor	Technical services	Tamesha Lenox	Unit D, 781	Unit B	Female
Technical services	Laurice	Wallach	Laurice Wallach	Technical services	Tamesha Lenox	Unit D, 782	Unit B	Male
Technical services	Paige	Mott	Paige Mott	Technical services	Tamesha Lenox	Unit D, 783	Unit B	Female
Technical services	Rosio	Wattenbarger	Rosio Wattenbarger	Technical services	Tamesha Lenox	Unit D, 784	Unit B	Male

Figure 4

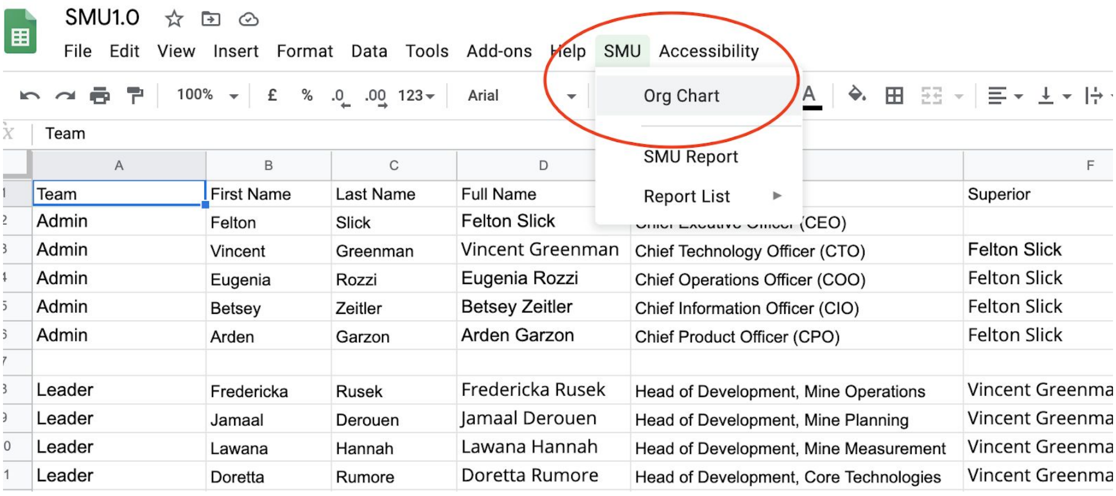
7. Save the files and back to 'main.gs' in 'Script editor', find the function 'onOpen' (Step 1 in Figure 5), You can change the name which will be displayed in the Toolbar (Figure 6).



```
function onOpen() {
  var ui = SpreadsheetApp.getUi();
  // Or DocumentApp or FormApp.
  ui.createMenu('SMU')
       .addItem('Org Chart', 'doIt')
      .addSeparator()
      .addItem('SMU Report', 'printall')
      .addSubMenu(ui.createMenu('Report List')
          .addItem('People-Building Report', 'print1')
          .addItem('Team-Building Report', 'print2')
          .addItem('People-Team Report', 'print3')
          .addItem('Ratio Report', 'print4'))
      .addToUi();


}

// Create Html page
function doIt(){
  var html = HtmlService.createHtmlOutputFromFile('index').setWidth(1200)
  .setHeight(900);

  var spreadsheet = SpreadsheetApp.getActive();

  SpreadsheetApp.getUi() // Or DocumentApp or SlidesApp or FormApp.
```

Step 1 Change the name in Tool Bar to what you want.

Figure 5



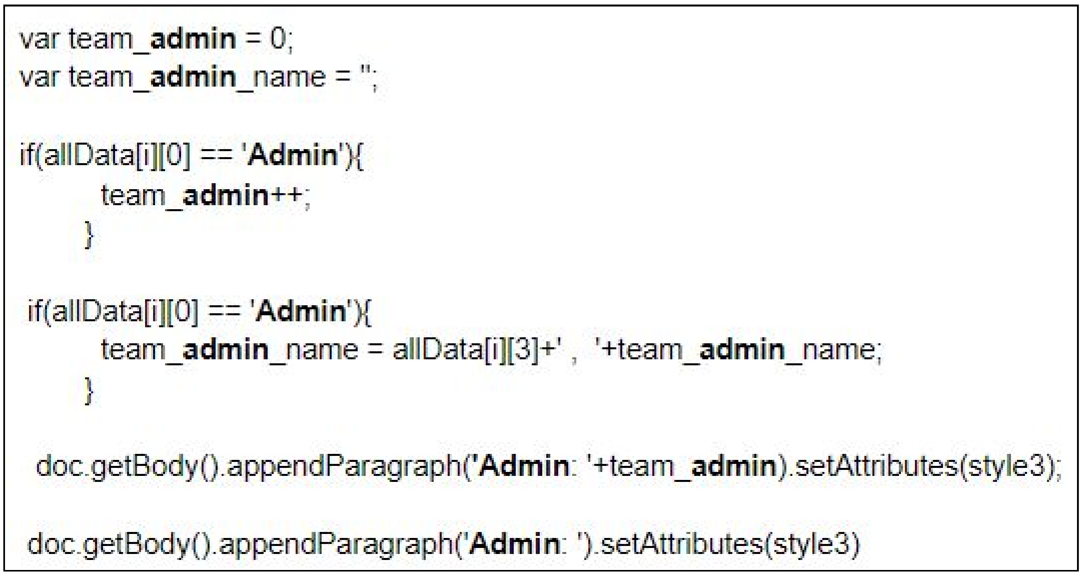
SMU1.0
File Edit View Insert Format Data Tools Add-ons Help SMU Accessibility
100%
Arial
Org Chart
SMU Report
Report List

Team	First Name	Last Name	Full Name		Superior
Admin	Felton	Slick	Felton Slick	Chief Executive Officer (CEO)	
Admin	Vincent	Greenman	Vincent Greenman	Chief Technology Officer (CTO)	Felton Slick
Admin	Eugenia	Rozzi	Eugenia Rozzi	Chief Operations Officer (COO)	Felton Slick
Admin	Betsey	Zeitler	Betsey Zeitler	Chief Information Officer (CIO)	Felton Slick
Admin	Arden	Garzon	Arden Garzon	Chief Product Officer (CPO)	Felton Slick
Leader	Fredericka	Rusek	Fredericka Rusek	Head of Development, Mine Operations	Vincent Greenma
Leader	Jamaal	Derouen	Jamaal Derouen	Head of Development, Mine Planning	Vincent Greenma
Leader	Lawana	Hannah	Lawana Hannah	Head of Development, Mine Measurement	Vincent Greenma
Leader	Doretta	Rumore	Doretta Rumore	Head of Development, Core Technologies	Vincent Greenma

Figure 6

8. Create 2 buttons in the spreadsheet. One is on the "**SearchOperation**" sheet and another is on the "**SearchResults**" sheet. Assign "searchDynamic" script to the button on the "**SearchOperation**" sheet and "**sendEmail**" script to button on the " SearchResults" sheet.

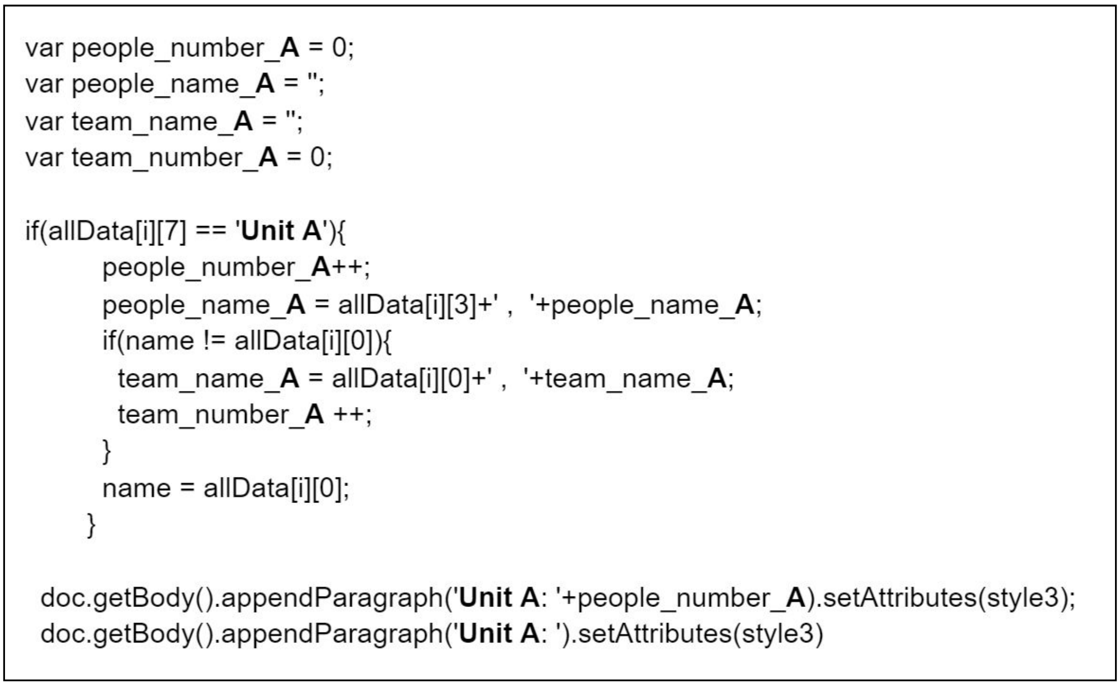
9. Customize the table. If you want to customize the team name, all variables that refer to the team name need to be changed. We strongly recommend that you copy the table together during initialization. Please replace all 'Admin' with your team names.



```
var team_admin = 0;
var team_admin_name = '';

if(allData[i][0] == 'Admin'){
        team_admin++;
     }

 if(allData[i][0] == 'Admin'){
        team_admin_name = allData[i][3]+' ,  '+team_admin_name;
     }

  doc.getBody().appendParagraph('Admin: '+team_admin).setAttributes(style3);

 doc.getBody().appendParagraph('Admin: ').setAttributes(style3)
```

Figure 7

10. If you want to customize the building name, you need to change the following content in the code as well. Please replace all 'Unit A' with your building names.



```
var people_number_A = 0;
var people_name_A = '';
var team_name_A = '';
var team_number_A = 0;

if(allData[i][7] == 'Unit A'){
         people_number_A++;
         people_name_A = allData[i][3]+' ,  '+people_name_A;
         if(name != allData[i][0]){
           team_name_A = allData[i][0]+' ,  '+team_name_A;
           team_number_A ++;
         }
         name = allData[i][0];
       }

  doc.getBody().appendParagraph('Unit A: '+people_number_A).setAttributes(style3);
  doc.getBody().appendParagraph('Unit A: ').setAttributes(style3)
```

Figure 8

Well Done! You have finished your installation!

6. Implementation Instructions

6.1 Sheet Detail

Spreadsheet App comprises 4 individual sheets. Maptek, SearchOperation, SearchResult, and EmailTemplate.

- **Maptek:** this sheet contains all the necessary details of employees at Maptek.
- **SearchOperation:** this sheet provides the end-user with the search function.
- **SearchResult:** this sheet displays the search results.
- **EmailTemplate:** this sheet contains the email template and the necessary keys.

6.2 Organizational Chart

Company organization chart application mainly contains two files, code.gs, and index.html.

Code.gs:

This file contains the features to run the application and the methods to read data from the google sheets file.

Function “dataZoom”	Used to read all the columns needed to build the graph, they are “Team”, “First Name”, “Last Name”, “Full Name”, “Position”, and “Superior”.
Function “doIt”	Used to open the index.html file.

Index.html:

This file contains the methods to build the graph and all the other methods about the graph. The features of each function are shown below.

Function setZoom	This function is used to set graph elements.
Function showVal	This function is used to show the basic elements we need for graphs.
Function onDrag showCoords Scroll_move:	These functions are used to move the graph through mouse drag.
Function drawChart	This function is used to draw the chart.
Function teamChart	This function is used to build a chart for each

	team.
Function updateOption	This function is used for sub-selection.
Function allOption	This function is used to select options.
Function teamSelect	This function is used to show the selection option of teams.
Function checkSelect	This function is used to show the select elements in the team.
Function searchChart	This function is used to search for a particular person or a group of persons.

6.3 Report Generation

All the coding for user story 3 can be found in main.gs.

- onOpen function is used for setting menu:
- addItem can make subtitles, the input of this function is name and function related to this menu.
- addSubMenu can add a sub menu, the input of this function is name and function related to this menu.

dataZoom function is used for reading the data from the table.

A two-dimensional array is used to read the data in the table. Currently, only 26 columns are defined. If necessary, you can just increase the number of columns. The name of the column needs to be written in advance, which means that when you need to add a column, you need to write the name of the column into this method.

We have defined a lot of global variables at the same time to facilitate subsequent methods to be obtained at any time during use.

var people_number_A = 0;	var people_name_A = ";
var people_number_B = 0;	var people_name_B = ";
var people_number_C = 0;	var people_name_C = ";
var people_number_D = 0;	var people_name_D = ";
var people_number_E = 0;	var people_name_E = ";

var team_admin = 0;	var team_admin_name = '';
var team_leader = 0;	var team_leader_name = '';
var team_tech = 0;	var team_tech_name = '';
var team_sales = 0;	var team_sales_name = '';
var team_finance = 0;	var team_finance_name = '';
var team_point = 0;	var team_point_name = '';
var team_sentry = 0;	var team_sentry_name = '';
var team_eureka = 0;	var team_eureka_name = '';
var team_vulcan = 0;	var team_vulcan_name = '';
var team_blast = 0;	var team_blast_name = '';
var team_evo = 0;	var team_evo_name = '';
var team_material = 0;	var team_material_name = '';
var team_data = 0;	var team_data_name = '';
var team_work = 0;	var team_work_name = '';
var team_uix = 0;	var team_uix_name = '';
var team_hardware = 0;	var team_hardware_name = '';
var team_integration = 0;	var team_integration_name = '';

var team_number_A = 0;	var team_name_A = '';
var team_number_B = 0;	var team_name_B = '';
var team_number_C = 0;	var team_name_C = '';
var team_number_D = 0;	var team_name_D = '';
var team_number_E = 0;	var team_name_E = '';

At the same time, we have also defined the format of the output generated globally, and this part can be freely changed according to font preferences.

Function doReport	This function is used to calculate the numbers required for document generation and store them in global variables.
Function printAll	This function is used to print all the sections in the report.
Function Print1	This function is used to print people and building reports.
Function Print2	This function is used to print team and building reports.
Function Print3	This function is used to print people and team reports.
Function Print4	This function is used to print ratio reports.

6.4 Search Function

During the search function implementation, the Maptek sheet's information is categorized into two data types—firstly string and secondly numbers. Values that represent dates are categorized into number type and support various arithmetic operators. The search function is dynamic, i.e., all the search parameters are selected from the information table. All the column's labels in the information sheet are automatically added to the search operation table and used as a search parameter. In addition to that, developers can also add their custom parameters and respective working column to the search operation table by modifying the array labeled "customPara” in the "getCustomParameters" function in the source code. The picture below describes the "getCustomParameters" function and array named "customPara". For example, if a developer wants to add the "Salary" column in the information sheet and wants other custom parameters like "SalaryGreaterThan" or "SalaryLowerThan" in the search function, he/she can simply add two elements “["SalaryGreaterThan","Salary"], ["SalaryLowerThan","Salary"],” to the "customPara" array.

getInfoColumnTitles()	This function returns an array of elements with columns labels of Information or data sheet.
getColumnTitles()	This function returns an array of elements with column labels and custom search parameters.
getCustomParameters()	This function returns an array of elements ONLY custom search parameters.
findCustomParaWorkVariable(x)	This function returns the working column for a custom search parameter “x”.
find_column_index(x)	This function returns the index of a column “x”.
createOperationsMenu()	This function created the search menu in the “SearchOperation” sheet.
writeToSheet()	This function writes the filtered or searched result on the “SearchResults” sheet.
getInformation()	This function retrieves the information from the “Information” sheet and performs a search function.

6.5 Mail Function

Mail function works on search results by sending an email to filtered employees.

Function updateTemplate	This function updates the email template and generates automatic keys as a new column is added to the information sheet or deleted from the information sheet.
Function getTemplateKeys	This function returns a two dimensional array with key and value as elements. This function replaces all the keys with corresponding values in the email body.
Function updateMailBody	This function creates a copy of the mail body and calls getTemplateKeys and replaces keys with corresponding values for individual email.
Function sendEmail	This function sends the email to the filtered result.