

ICS Department at KFUPM

# ICS 201 Lab Project - 191

Mail-Merge Organizer

Originally Designed by Mr. Adil Alsuhaim

Edited by Esam



2019

## Objective

In this project the students are expected to apply what they learned during the ICS 201 course to implement a fully working project. The application will help students establish a better understanding of how real world projects work. The students will be working on an existing code and utilize it to add more functionalities and a graphical user interface to it. In addition, the student is also expected to read and research how certain functionalities could be done in Java programming language.

We hope that the experience with this project will establish a foundation on which students can learn how to program new functionalities into their future project.

## Introduction to the concept of Mail-Merge

Mail-Merge is a process of automation which extracts information from a data source (such as Microsoft Outlook contacts) to generate large amount of mail documents. These mail documents could be regular post mail or electronic mail documents (E-mails).

You have probably received documents generated by mail-merge software. For example bank correspondences come in your mail box folded with your mailing address on the top. Another example is when you get emails because you are on a mailing list. Take a look at this example, which uses the name “Adil” as if the mailer is taking to the recipient personally:

Hey Adil,

Today I want to discuss with you how you can improve your concentration with the right diet. Many people believe that there is no need for medication for people who would suffer from lack of concentration and short attention spans. I find it even outrageous that millions of children are put on medications such as Ritalin. Wouldn't you agree, Adil?

To start, you need to look at what type of substances are you putting in your system. Too much sugar will give you a burst of energy but it is only followed by a crash that would make you feel tired and unwilling to perform, which greatly affects your concentration.

Now, the sender of such e-mail did not actually write the recipient names on every single email. He simply created a template and generated many e-mail documents automatically. The sender facilitated a data source of contact information of the entire subscribers. This makes it easy to send thousands of personalized e-mail documents automatically.

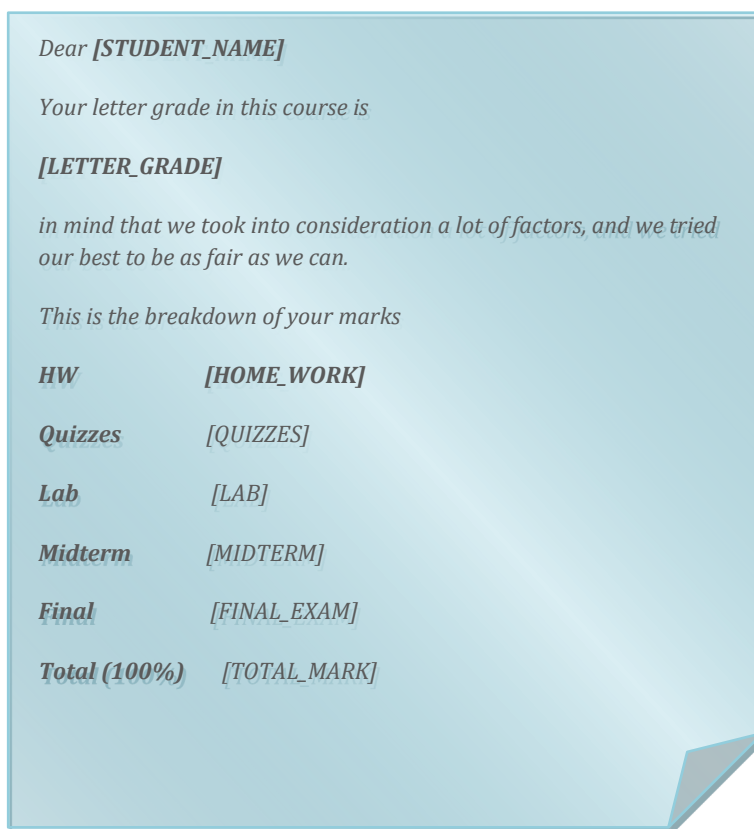
Take a look at this *template* for a regular mail document of an invitation to a conference



All the generated mail-merge documents will have this format. The place holders are enclosed by `[[ ]]`. For example, an example letter could be addressed to “Dr. Khalid Al-Sultan” where “Dr.” replaces `[[RECIPIENT_TITLE]]` and “Khalid Al-Sultan” replaces `[[RECIPIENT_FULLNAME]]`.

The sender can then print hundreds of documents and fold them in proper envelopes and take them to the post office. This practice is done every day by secretary workers, organizer, entrepreneurs, etc.

Here is an example of a MailMerge application for reporting student's grades.



Notice that the “place holders” templates include letter grade, homework, quizzes, etc. So since the data is read from a file (which looks is typically a table) that has headers that would corresponds to the “place holders” shown in this template.

## Preparation for the project

You are given a set of java classes to use within your project. Those classes are used to send e-mail messages and create PDF document. You do not need to write the code for them as they are already provided, but you will need to make any appropriate changes you may need for your project.

Your **first task** is to learn how to use those classes. There is a tutorial given later in this document. We have also provided you with JAR files, which contain the libraries needed to create PDF files or send e-mails so you need to include these into your project, or otherwise the imports for javax.mail (for example) would not work.

## Post mail document creation

For post mail document, PDF documents need to be created. Java does not support PDF by default. However, there are freely available libraries that can be used to generate PDF documents. For this project, you will learn how to use iText PDF library. Please refer to the references [1] for a quick tutorial. The supplied code already has the classes with sample code. You are done if the PDF file is created successfully in a reasonable format.

## Recommended SDK and Tools for this Project:

For this project, it is recommended that you work with Eclipse Neon that has all in one package support for JavaFX, it can be downloaded from the following link:

<http://efxclipse.bestsolution.at/install.html#all-in-one>

Scene Builder, the drag and drop support for JavaFX development is also recommended for this project, you may download it from the following link:

<https://www.oracle.com/java/technologies/javafxscenebuilder-1x-archive-downloads.html>

JAR files needed for the project are included within the project folder.

## Sending emails using JavaMail API

The MailMerge.java has an example that sends an email using your KFUPM portal user name.

The MailMerge.java file also has code to write a PDF document from your Java file.

## Project Requirements (what you should do)

### Part 1 : Design of classes

You are required to design class files that would be useful for your project.

### Part 2 : Your Main Program

Your main program will be a graphical user interface application. Refer to the screenshots section for better clarification.

The program will read the contacts information from a file. A text file created by you. Store your contacts in an ArrayList of type "Contact", which is the class you have designed.

For the printable PDF document, we intend to create one PDF document with multiple pages. Every page would have information of a different contact. A secretary person would use this PDF file by printing and folding the documents.

A printable document made for the mail should have the mailing address printed on the top quarter of the page and to the left. It is advised to leave a good space after the address and then start with the actual letter.

#### Opening a template file

The user will be asked to browse for the name of a template text file to be read. When the file is loaded successfully, *an appropriate message should be displayed to the user.*

#### Sending Emails

This means that you want to send email letters. The program should prompt the user by a window or a dialog box to browse for a contacts file before proceeding.

#### Create a contact

This option should only be enabled if a certain contact list is loaded. It should prompt the user with a window which has the fields that are relevant to the loaded contact list. This means that you may deal with a variable number of text fields and labels.

**Your program should meet the following minimum requirements:**

- Develop a way to create contact and save them to a file. The file can be a text file.
- Contains a text area in which the letter template can be written. The user will not type the place holders manually but rather select them from a list or menu.
- The program should give the user a choice whether to create PDF files or to send email to the list.
- The possible “place holders” for your program should be loaded from the data files
- All added functionalities and notification received from the program (including error messages resulting from exceptions) should be handled using graphical user interface.

**Add-ons for a bonus**

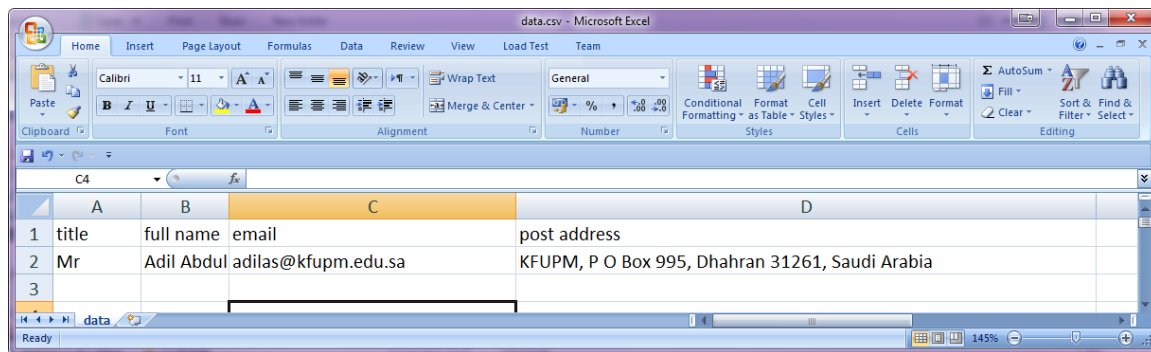
- Add a functionality to manage contacts (i.e. add, delete and edit).
- Add a functionality where the user has to set up an email account. For example, if a user want to send emails from an account that is other than KFUPM, such as Gmail, Yahoo ... etc.

## Guidelines and Screenshots

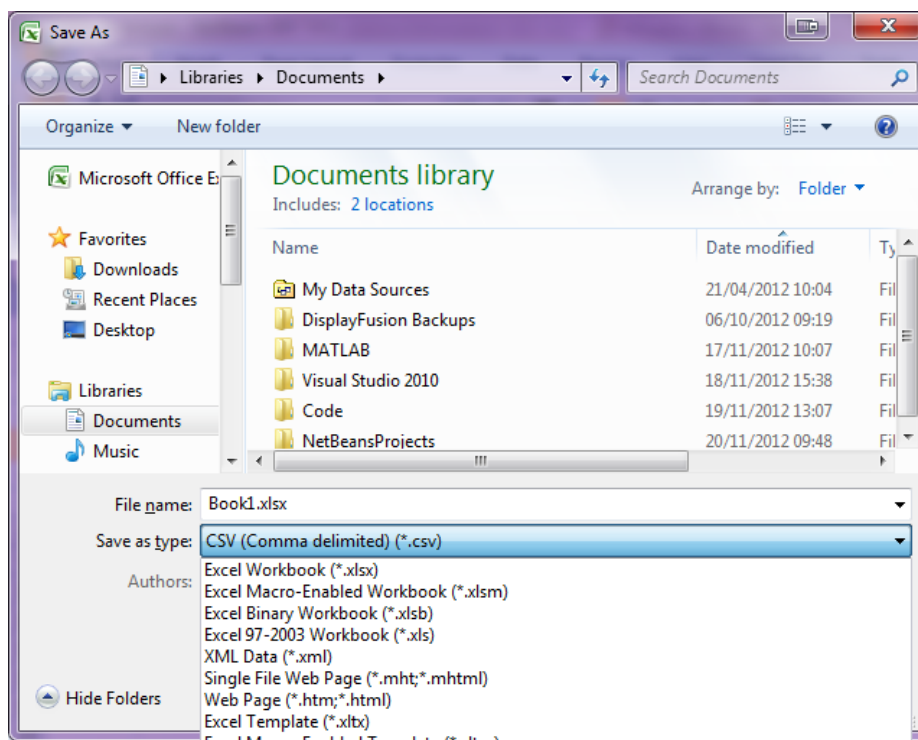
For your reference, there is a ProjectDemo runnable jar that is included in the project folder, if you run that file you will be able to see a limited version of the project. This will aid you in knowing how the project should look like and how should it function. You must, however, implement the mailing feature that is not available in the demo limited version.

Make sure you test your program's mailing capability inside and outside the campus.

We can create a file in Microsoft Excel (or any other spreadsheet application like OpenOffice) and save it in CSV file.

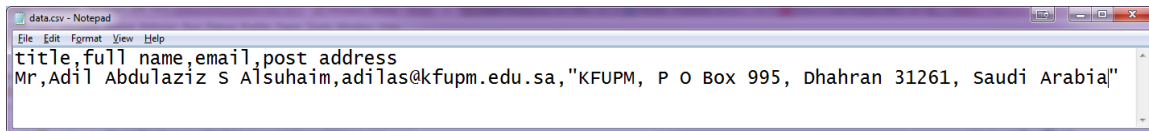


Saving a file as a Comma Separated Values (CSV) file in Microsoft Excel.



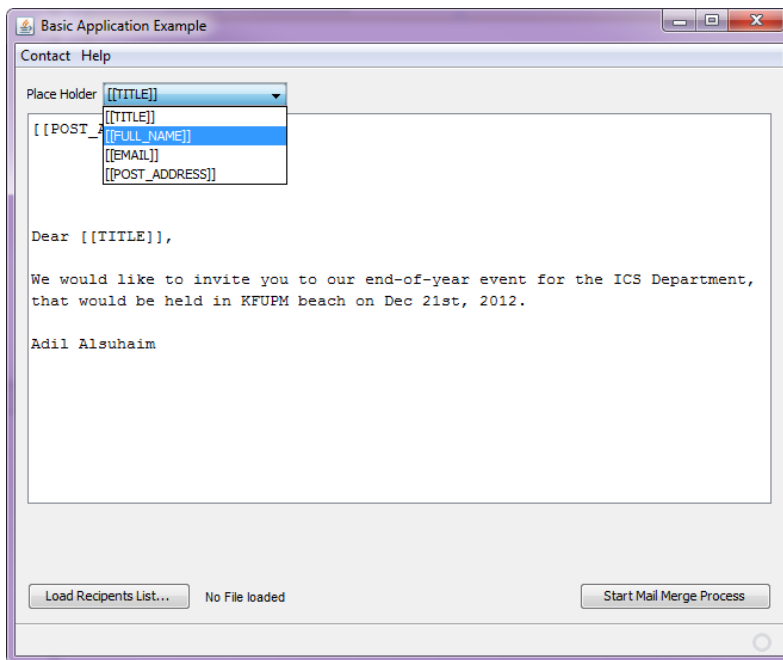


This is how the CSV file look. It's only a text separated by commas ",". Every line represents a table row, and every comma symbol represents moving to the next adjacent cell.

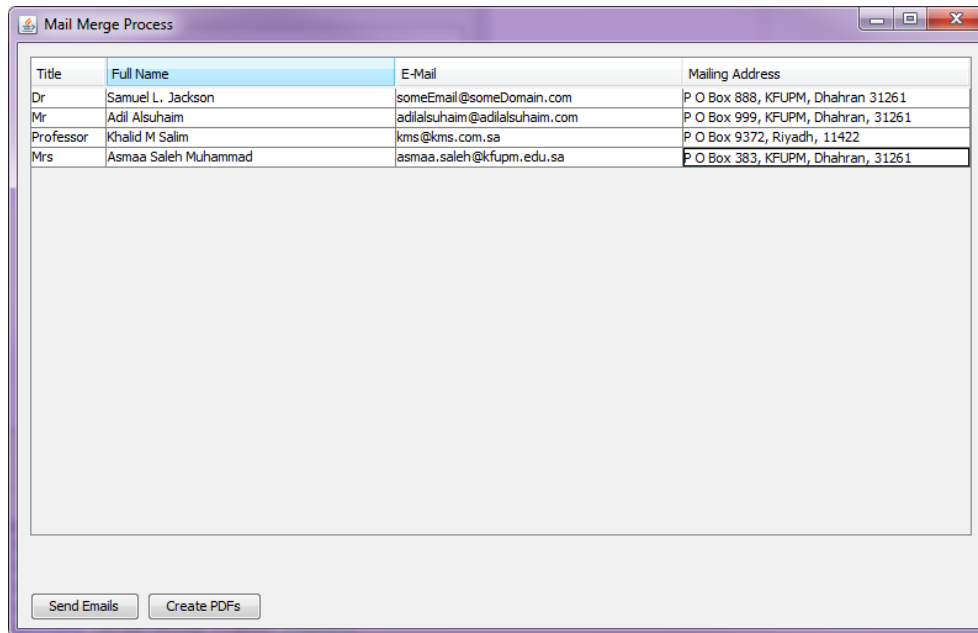


```
data.csv - Notepad
File Edit Format View Help
title,full name,email,post address
Mr,Adil Abdulaziz S Alsuhaim,adilas@kfupm.edu.sa,"KFUPM, P O Box 995, Dhahran 31261, Saudi Arabia"
```

Use the following screenshots as an example of how you should designed your user interface



A screenshot showing a list of contacts before processing. The user would make a choice of whether to create a PDF file of the intended recipients or just choose to send emails.



## Grading Criteria

Keep these grading criteria in mind when doing your project:

- Follow proper Java coding style. Use proper names for variables. Write comments and indent your code appropriately.
- You will be graded for implementing the functionalities of this program as requested. The program should work for files of any length and not be fixed to a specific file size.
- You are supposed to be working in teams of two students each.

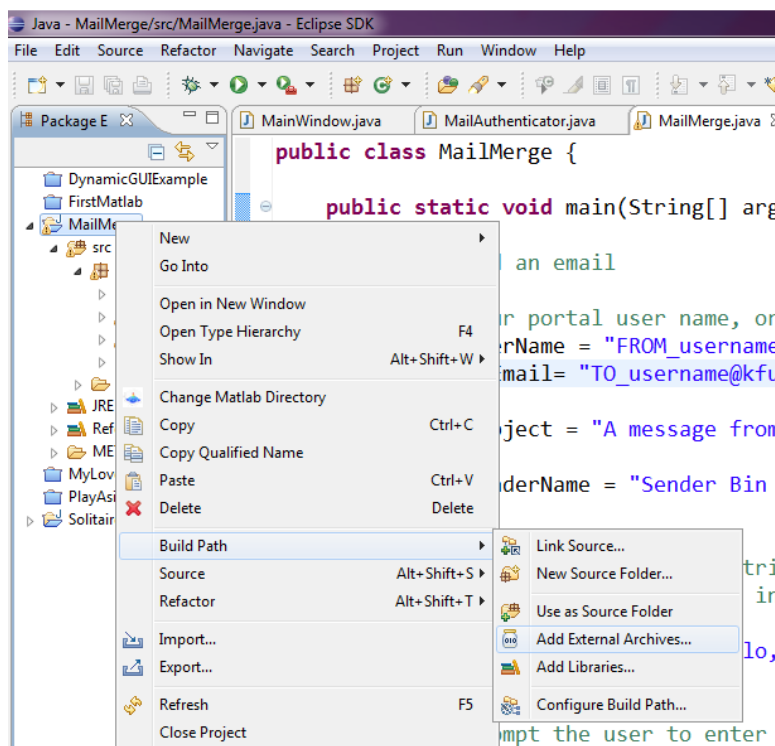
## References

1. Generate PDF files from Java applications dynamically  
<http://www.ibm.com/developerworks/opensource/library/os-javapdf/>
2. Real's How-To <http://www.rgagnon.com/javadetails/java-0321.html>
3. Ross Olweiler's personal blog site <http://ross-o.com/index.php?/archives/17-No-AUTH-command-has-been-given.html>

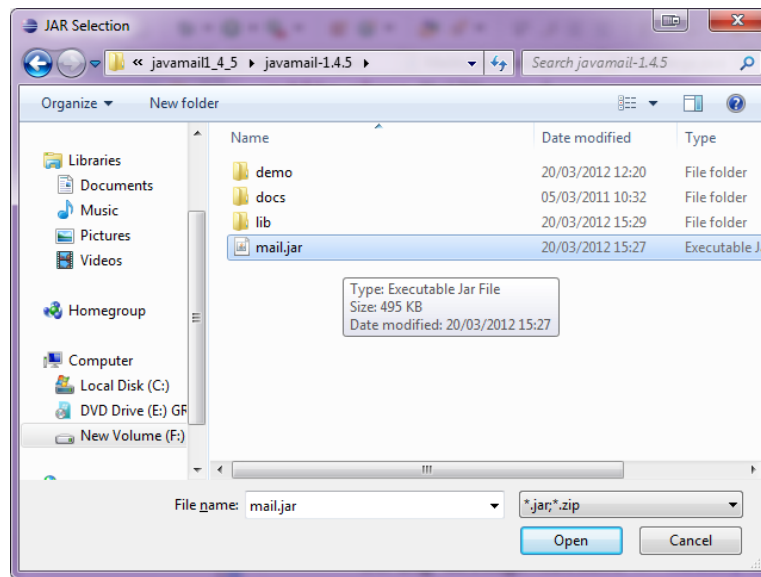
## Appendices

### Importing JAR files into an Eclipse project

Importing jar files by right-clicking on the project and then Build Path > Add External Archives...



Then browse for the JAR file(s) and include them in the project



## **Sending an email using JavaAPI**

For this purpose the following java files are provided:

*SendEmailOffice365.java*, *PasswordDialog.java*, and *LoginCredentials.java*

Leave the classes “*PasswordDialog.java*” and “*LoginCredentials.java*” untouched, as they are helper classes for the “*SendEmailOffice365.java*” class, which is the class that is used to send email messages.