HR Process Automation

This document is intended to describe the functional and technical details involved in automating an HR Process for Aspose HR department.

# Problem

Aspose employees were given increments and the HR department has to inform each employee about their new increment. The employees are to be informed by email with the increment letter attached. HR department wants to automate this process so they don’t have to create increment letters and email each employee manually.

# Solution

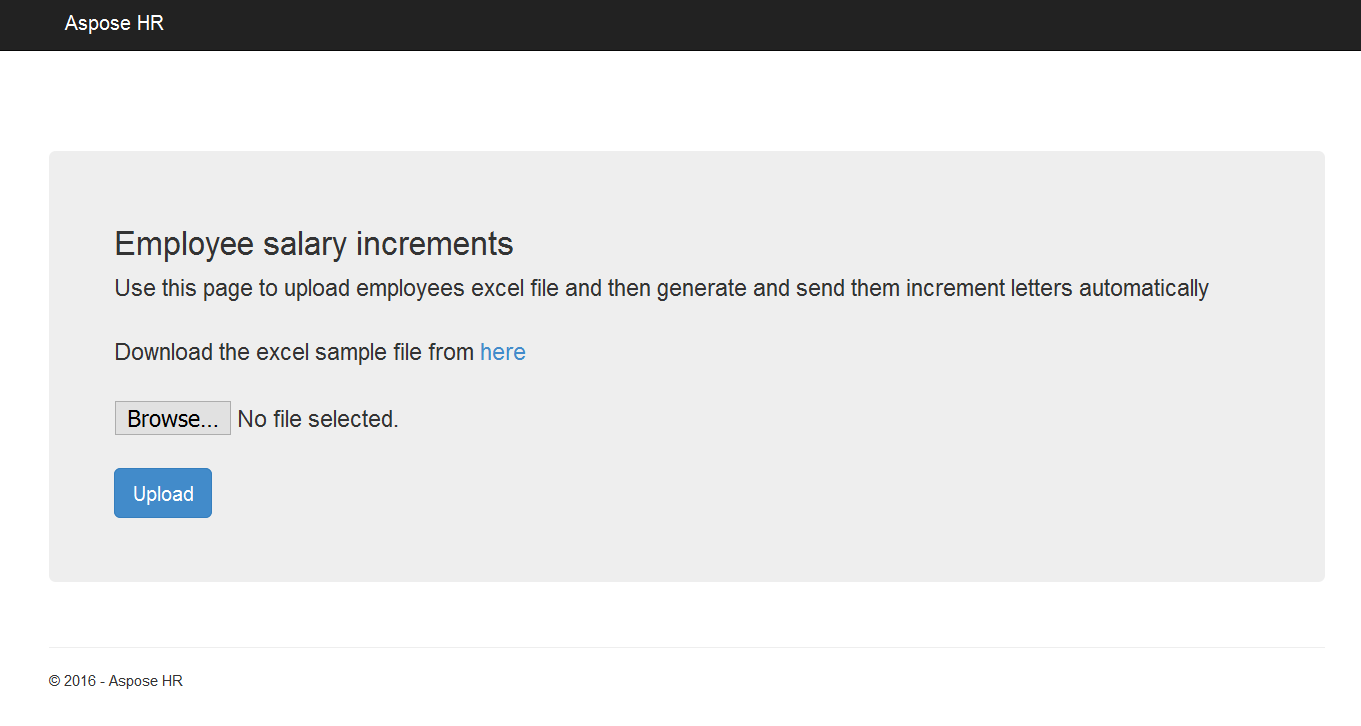
The solution suggested is to create a screen where HR can upload an excel file with the list of employees including the following fields,

1. Full name
2. Email
3. Address
4. Salary

After successful upload HR manager will be presented with a table to review the employee list uploaded and send increment letters later. To simplify the generation of increments letter, a mail merge template will be created, which will be used to generate increment letters programmatically and will be send to each employee as an email attachment.

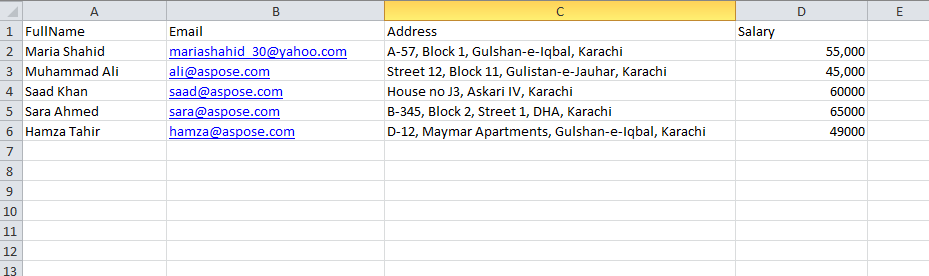
# Implementation

The described HR process is automated using an ASP.NET Web Application. A simple screen will be created using Web Forms where HR can upload an excel file.

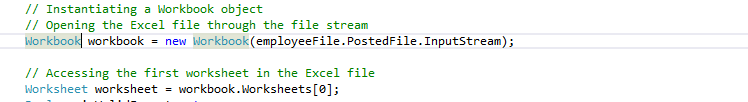


## Excel File Processing

The uploaded excel file will be processed using Aspose.Cells API to simplify data extraction and validation. The format of the excel file is shown below,

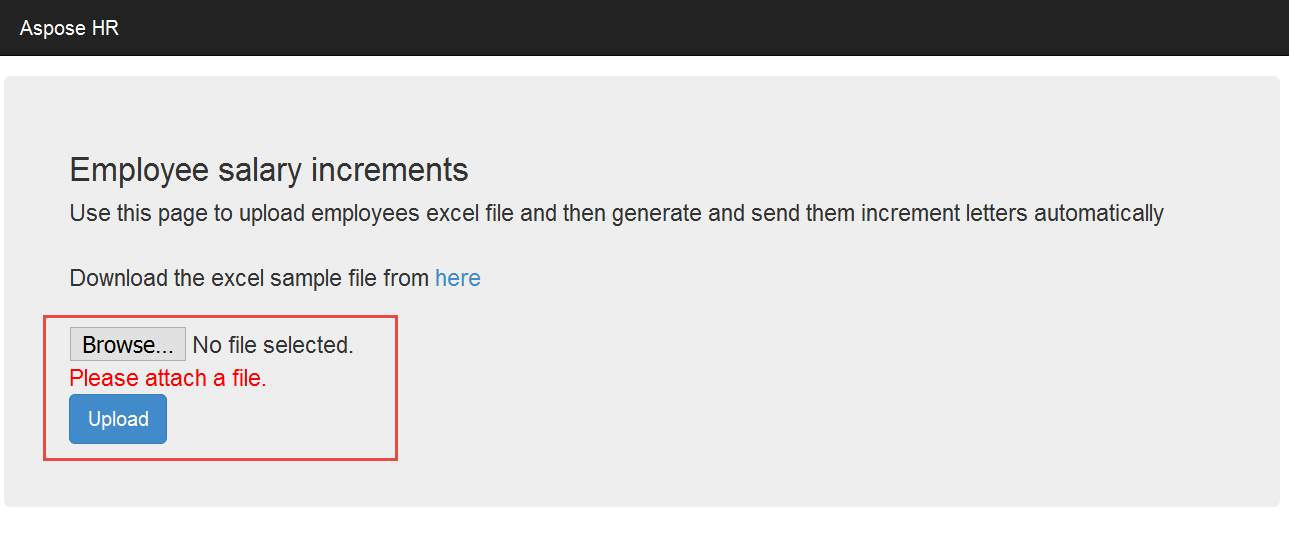


Data from the excel file will be accessed using Aspose.Cells API for extraction and validation. The worksheet in the excel file will be opened using Aspose.Cells Workbook and Worksheet classes respectively.

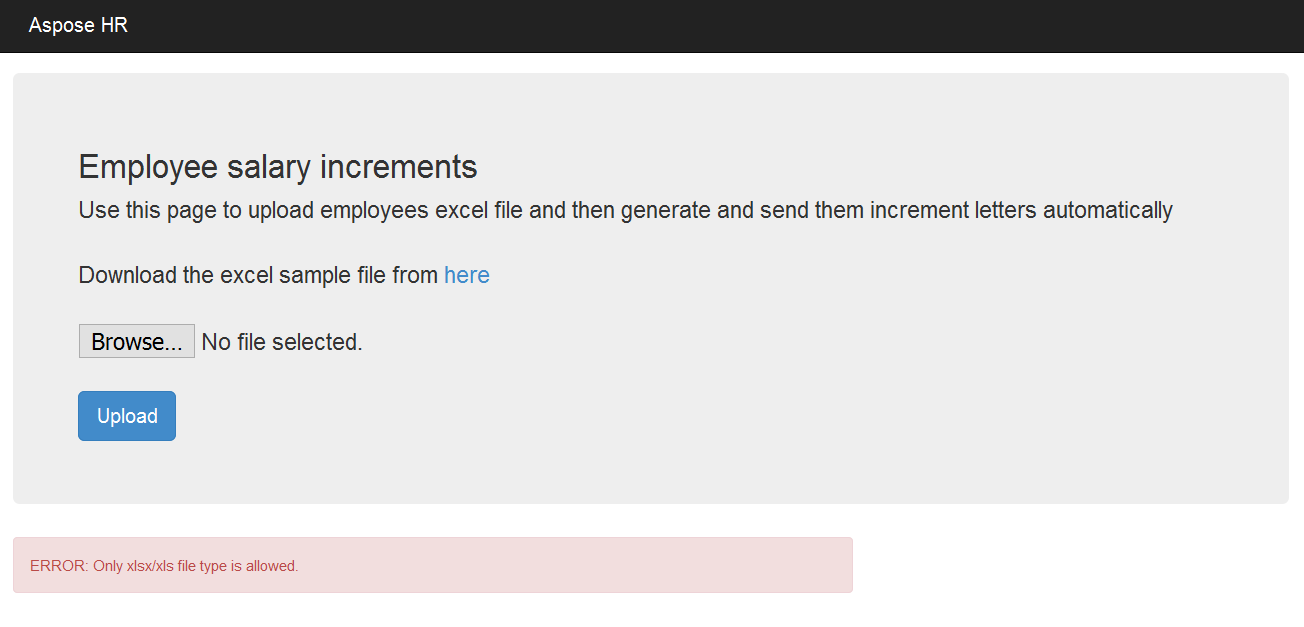


Following validations will be performed on the uploaded file.

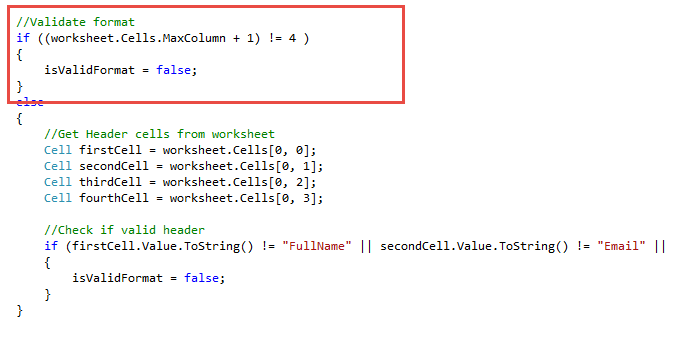
* Required Validation - This validation is placed on both client and server side to ensure that a file is attached for processing.



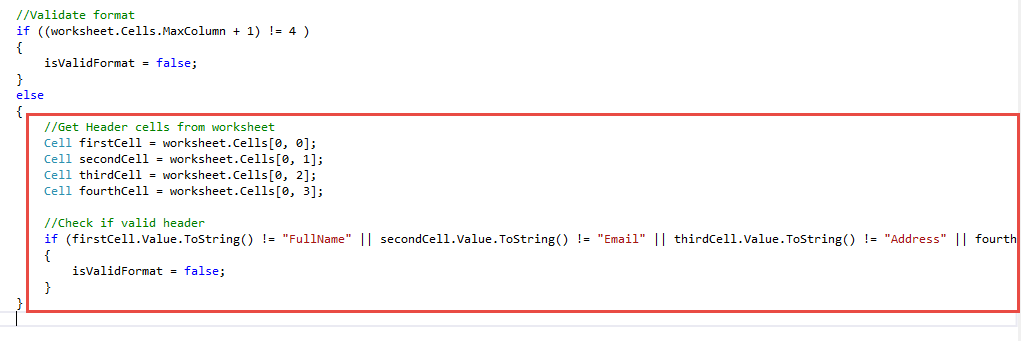
* File Extension Validation - After file is uploaded to server, file will be validated for a valid extension. The file extension expected is .xlsx or .xls. If a file is attached in any other format, following message will be displayed,



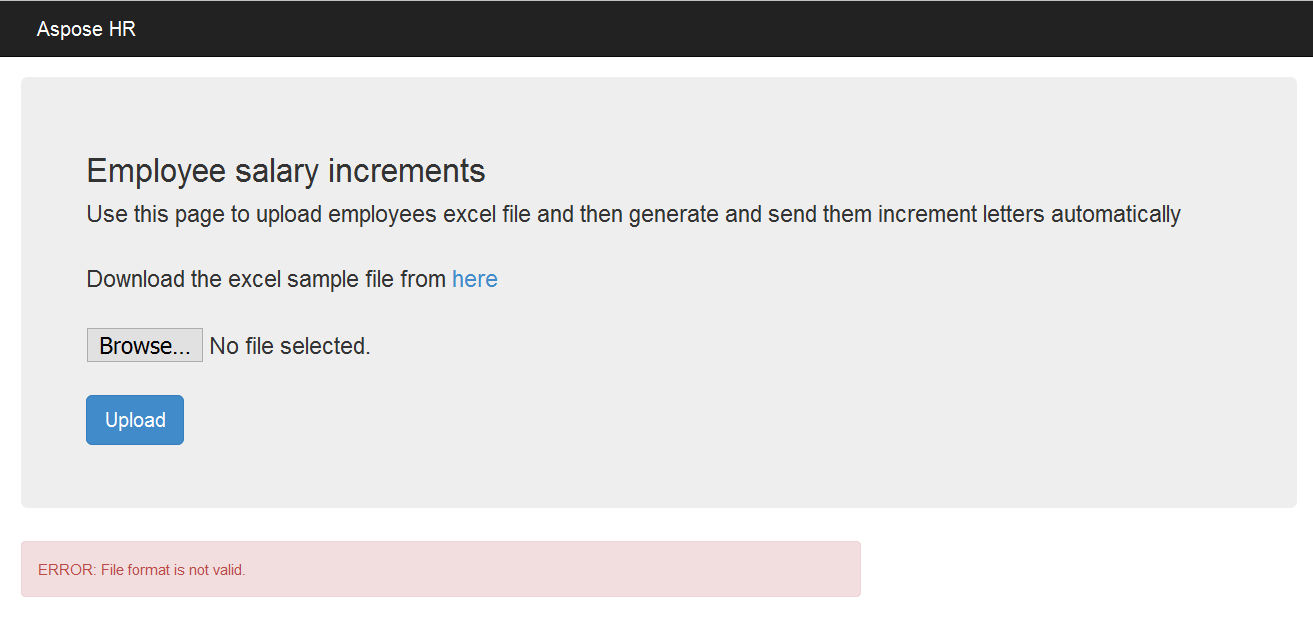
* File Format Validation – There are two format validations done on the worksheet accessed from Aspose.Cells API. First the number of columns will be checked. If number of columns is not equal to 4, file format will be marked as invalid.



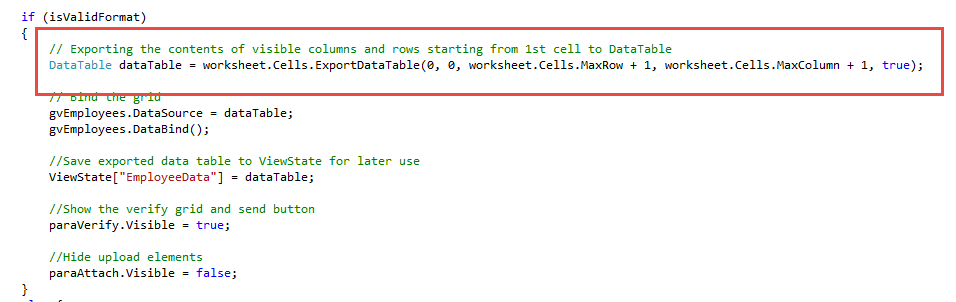
Secondly the names of the columns will be checked by accessing the header cell values. If the values don’t match as in the supplied format, then also the file format will be marked as invalid.



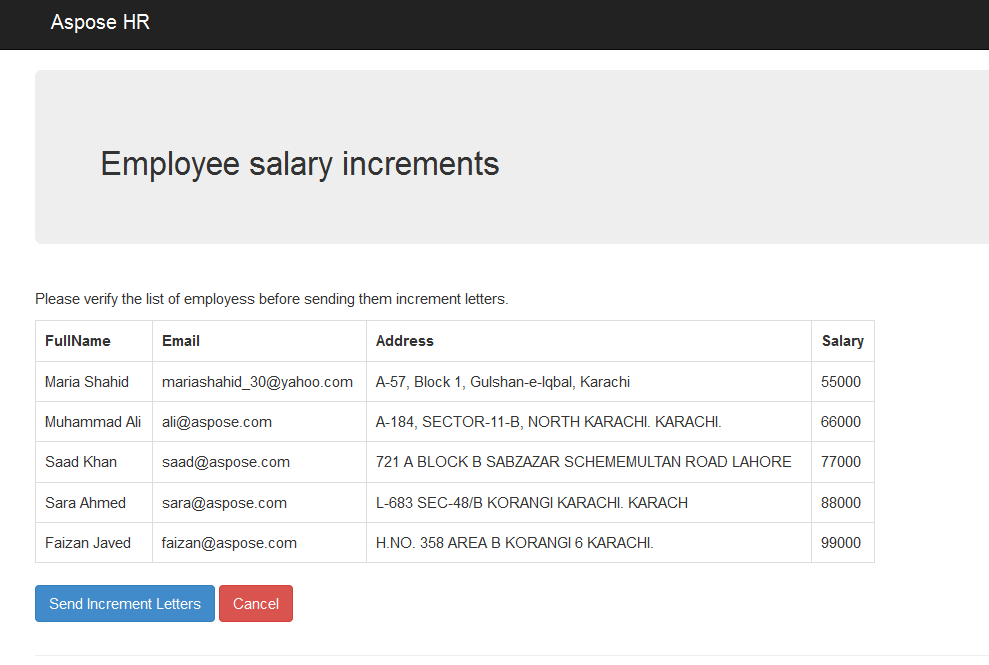
Following message will be displayed to user if file format is invalid.



After all validations pass, data will be extracted. To extract data into an ADO.NET datatable, the public method ExportDataTable will be invoked on the worksheet object. The same datatable will be used to bind the gridview displayed to user for verification and to send increment letters later in the process. Datatable will be saved into ViewState for later use.

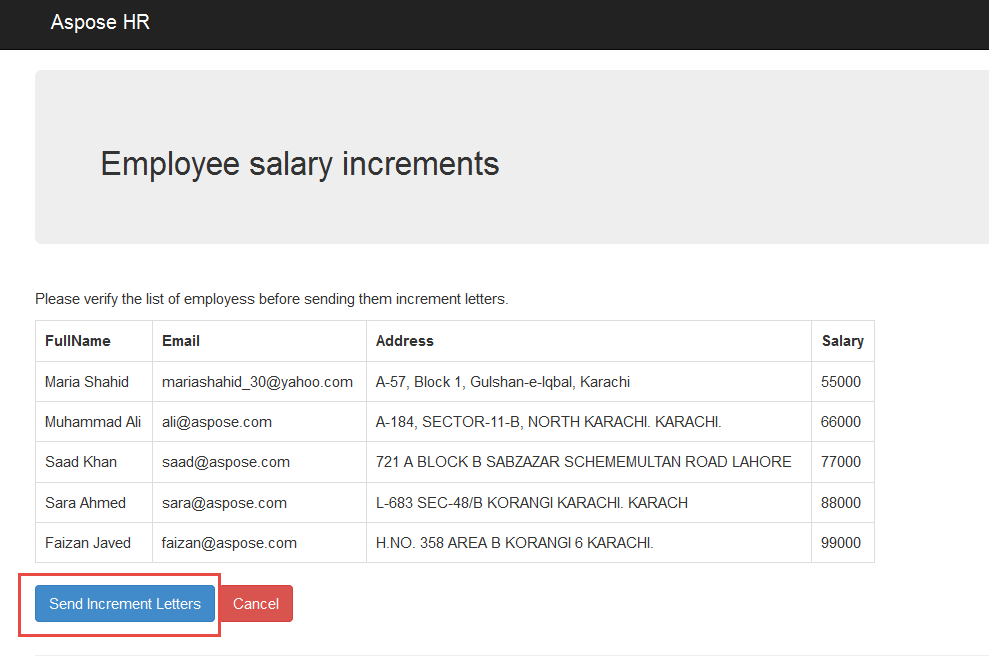


Finally, extracted data will be displayed to user for verification,

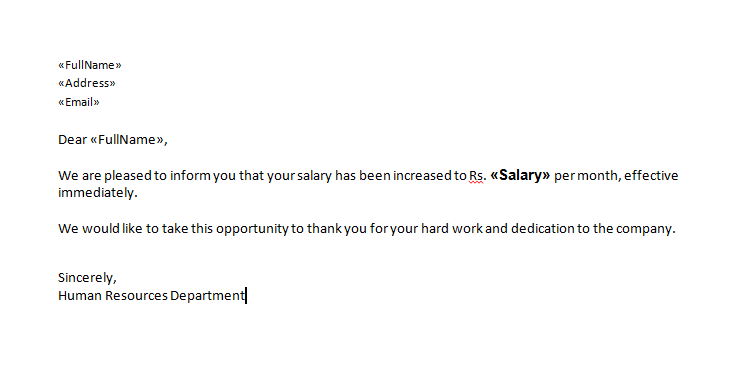


## Increment Letter Generation & Email

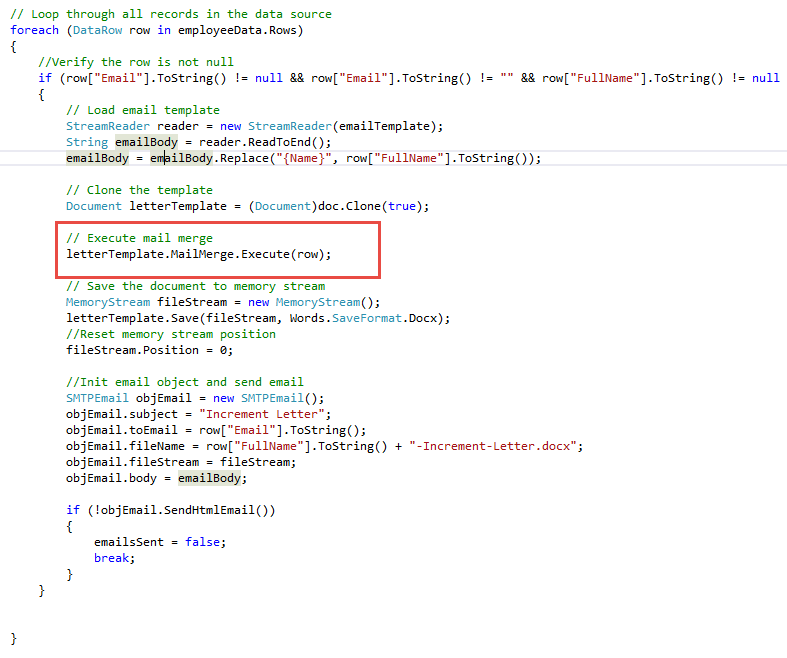
The primary purpose of this process automation is to generate increment letters and email them to respective employees. Once the user has verified the data, this process will be triggered when they click on **Send Increment Letters** button.



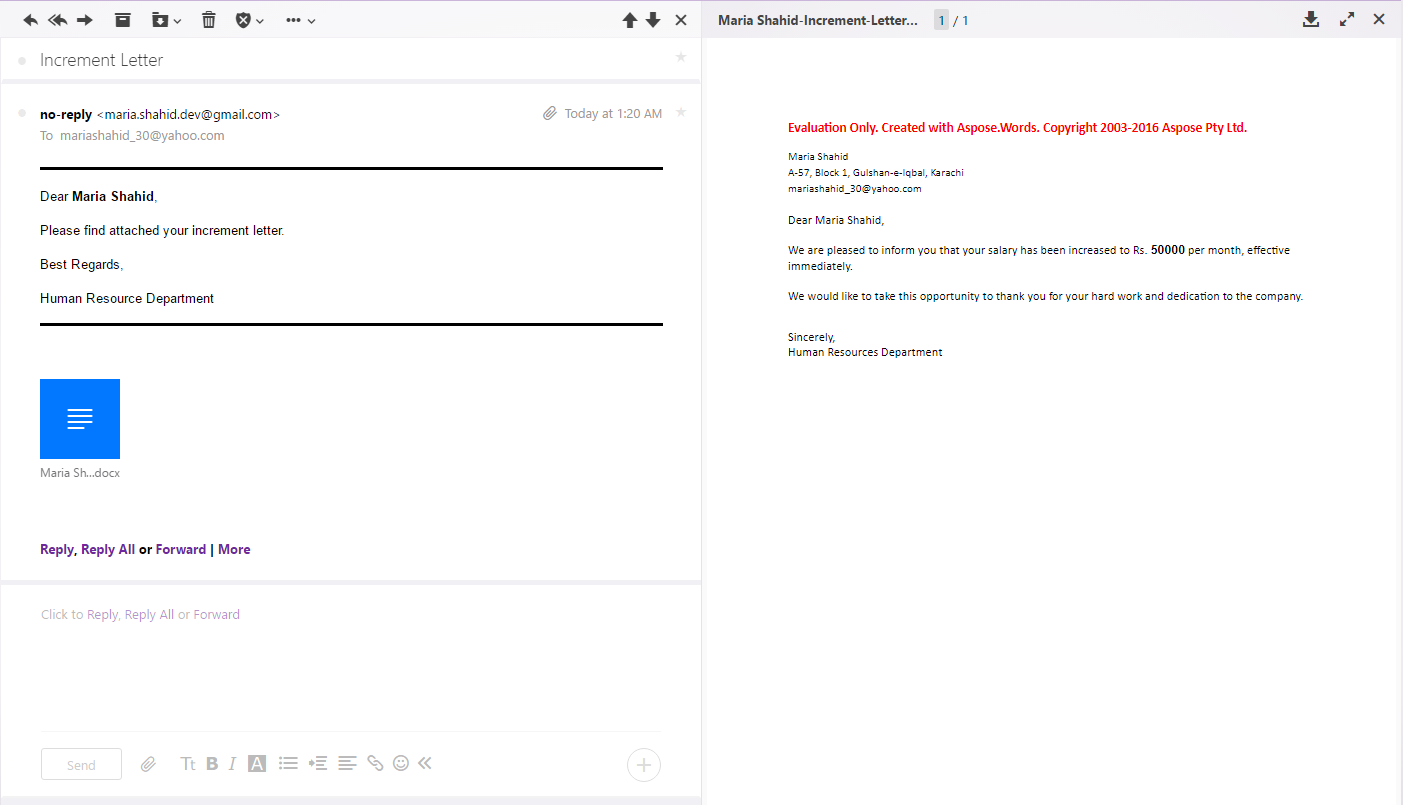
Now, to generate the increment letters, a mail merge template is pre-created with mail merge fields as follows,



Aspose.Words API will be used to merge the data extracted from excel file earlier during upload with the merge mail template created. For this purpose a loop will be executed on extracted data which is loaded from ViewState and each row will then be merged with the template.



The merged document is saved to a memory stream, which is used to create an attachment for the email. Final step in the process is to send email to the respective employee. Email is sent using System.Net.Mail. Below is a screenshot for a sample email.



Once the process is finished sending emails to all employees, success message is displayed to the user.

