**Anniversary Mailer - Version 1.0**

**Software Design Document**

**Version: 1.0**

**Author: Avik Deb | avikdeb@gmail.com**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Contents

[1. Document Change History 2](#_Toc497060112)

[2. Requirement 3](#_Toc497060113)

[3. Solution Design 3](#_Toc497060114)

[4. Class Design 3](#_Toc497060115)

[5. Sequence Diagram 3](#_Toc497060116)

[6. Operation and Maintenance 3](#_Toc497060117)

# Document Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Change Description** | **Changed by** | **Changed on** | **Version** |
| Initial version | Avik Deb | 29-10-2017 | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Requirement

The software should send automated e-mail greetings to the users on their birthdays with custom greetings message and greetings cards. Birthdates, greetings messages and cards to be furnished and updated by the admin or HR. The application should run on different machines and should be deployable without being compiled or involving any program change. As it will be primarily used and maintained by non-technical folks.

# Solution Design

Anniversary dates of the selected people are stored in an excel file – which is maintained by the administrators or authorized persons. They can add / remove entries in the excel. This excel serves the purpose of datasource.

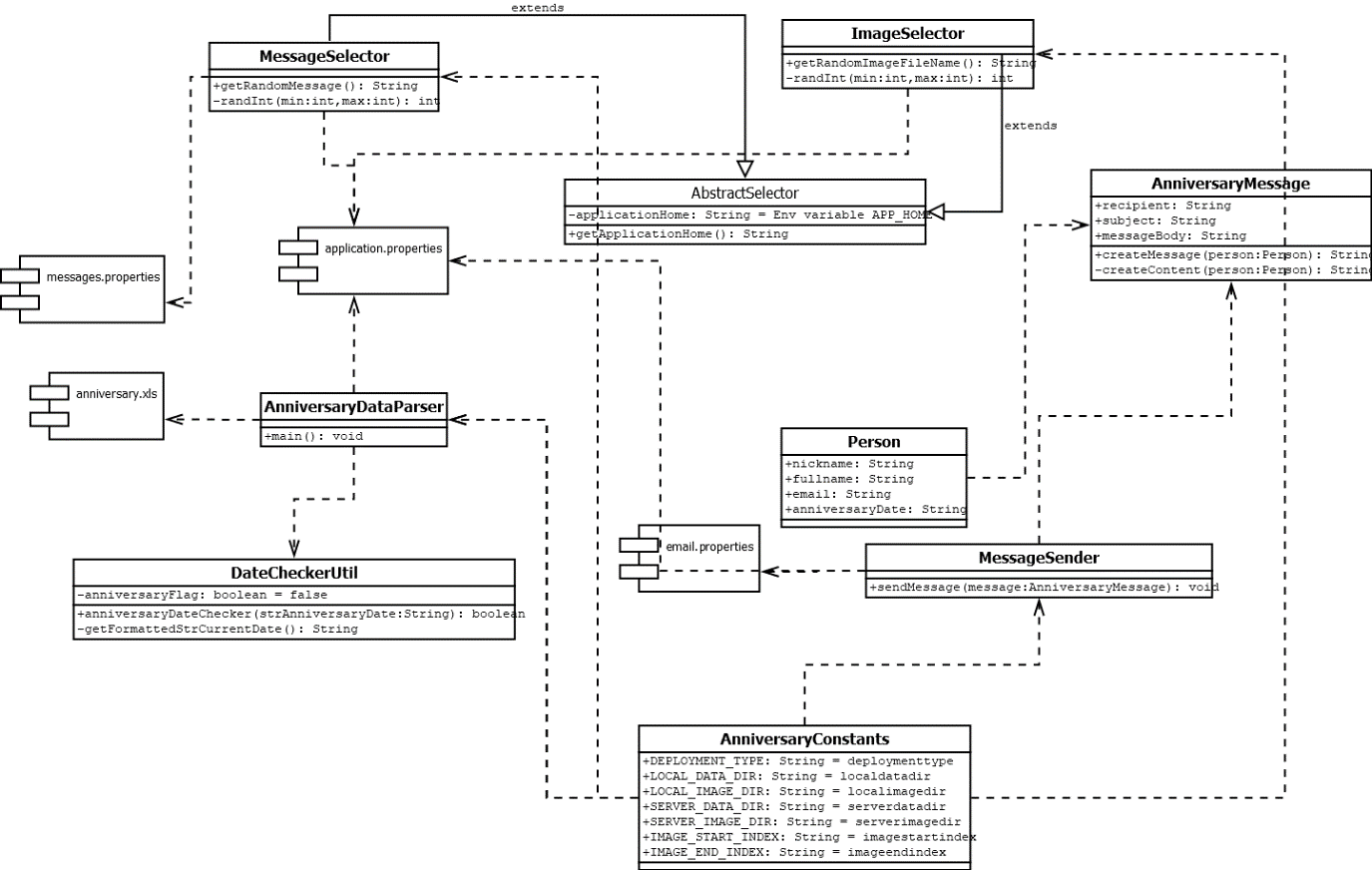
Selected images of greetings and messages are also chosen, maintained and uploaded by the authorized persons. The images follow a naming convention like 1.png, 2.png, 3.png ... 30.png etc. Images and messages are selected randomly.

The software runs daily in the morning and checks if any person is having birthday on the same day – if yes, it sends the wish message as discussed above. The executable is a Runnable JAR and it is called from Windows batch file (.bat) or Unix / Linux shell script and is scheduled to run through Windows Scheduler or Cron depending upon the host operating system of the server.

The software is developed in core Java (SDK version 8) so it may be build and deployed on any operating system Windows or Unix / Linux having Java installed (JDK/JRE version 8).

# Class Design

The class design can be summarized in the diagram below (the diagram is separately available in GitHub location: https://github.com/trainingport/anniversary) – zoom in for better viewability.

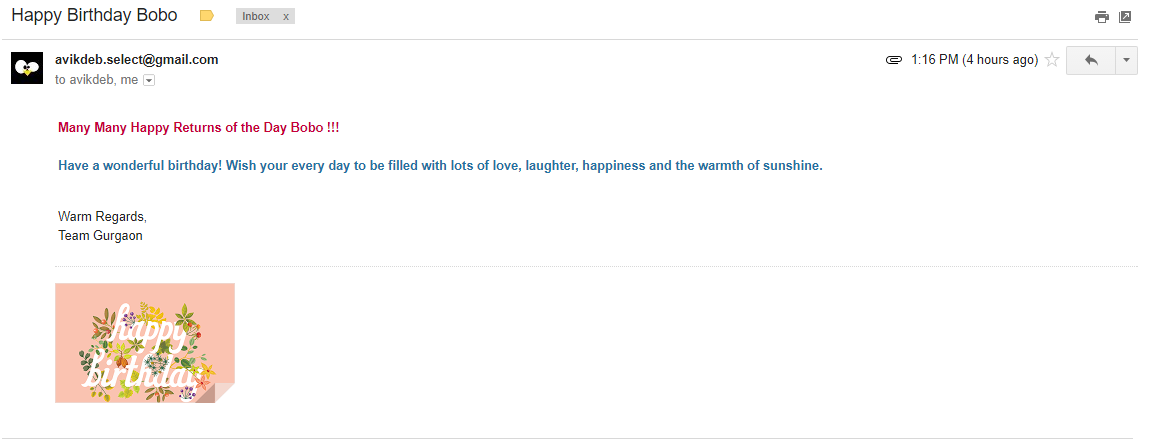


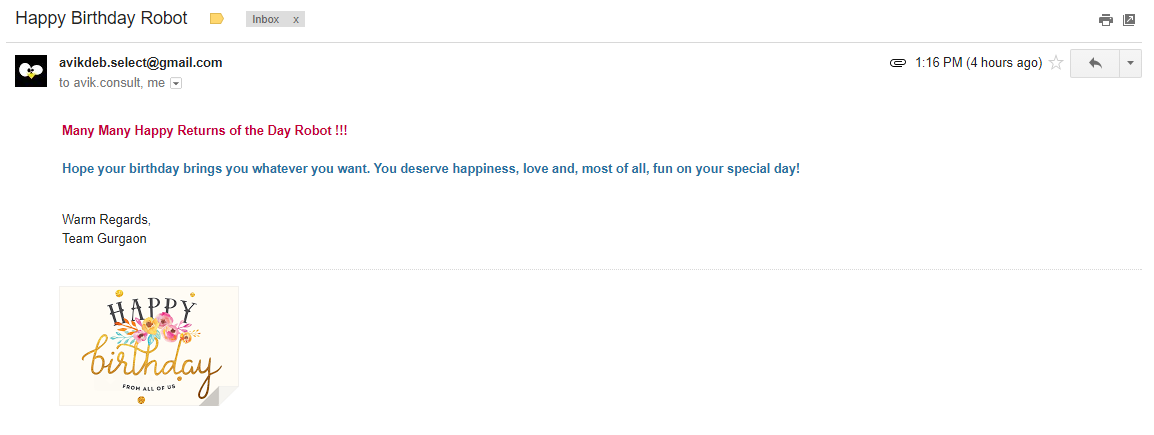
**Important Classes and their behaviour:**

1. The data source for this application is application.xls file.
2. Location of this file is specified by the properties file with keys such as **localdatadir** and/or **serverdatadir** – **application.properties**. Value in **localdatadir** is used for local developer testing while **serverdatadir** refers to the server environment.
3. **AnniversaryDataParser** class in **util** package is the main class containing the **main()** method – It uses the Apache POI API to parse the excel file and take necessary actions.
4. **Person** class is a model class written as per Java Beans specification and is kept in **model** package. It represents the person to whom the birthday mail is sent.
5. **AnniversaryMessage** is another model class written as per Java Beans specification and is kept in **model** package. It represents the message to be sent in the birthday mail. In addition to usual getter and setter methods, it contains **createMessage()** method that returns specific **AnniversaryMessage** object with recipient name set in the message along with all details. The private method **createContent()** is used to create the content internally.
6. **Selector** Classes in **util** package – **MessageSelector** and **ImageSelector** classes both extends the parent class **AbstractSelector** and inherits the properties file location in order to get the location of the messages and images. They select random messages and images. Private method **randInt()** is used to generate the random number within a specific range.
7. **DateCheckerUtil** class in **util** package is used to format and check the dates by comparing current date and the date mentioned in the excel file. Method **anniversaryDateChecker()** checks the date and returns Boolean value based on which the Parser class takes the decision to send email.
8. **MessageSender** class in package **message** is responsible for sending emails. It uses Java mail API. The required properties are read from the interface **AnniversaryContants** and the properties file.
9. **AnniversaryConstants** is a java interface found in constants package – it contains the keys that are used in various externalized properties file. This interface is the bridge that connects the externalized properties with the application alleviating the need for deployments with any change in properties.

# Output

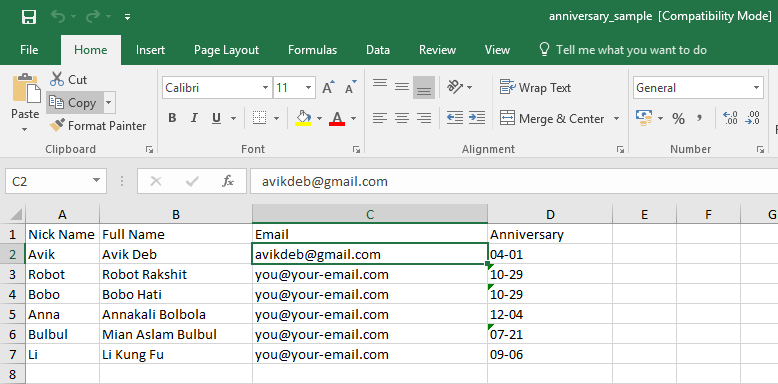
Sample e-mail received by different people on their birthday:





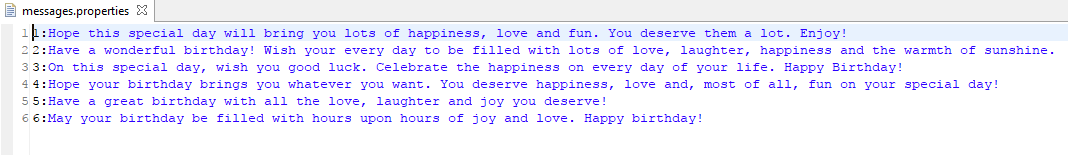
# Build, Operation and Maintenance

* 1. Source code is maintained in GitHub location – main development being delivered in master while stable releases can be downloaded from the respective branches: https://github.com/trainingport/anniversary
  2. To set the code is action – first define an environment variable ANNIVERSARY\_HOME and set the value to the location where the data files, images and the executable would be kept.
  3. Then create a runnable jar file either from eclipse IDE or using suitable maven plug-in or ant. Once the runnable jar is created schedule the run through a batch file for Windows systems or shell script for Unix / Linux systems.
  4. Data file contains the excel where person and their anniversary dates are specified in the format as shown below:

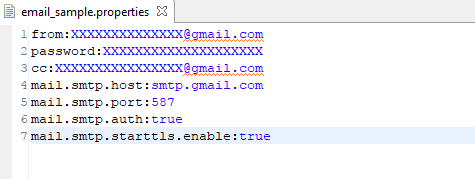


Anniversary date being given in MM-DD format and as General type in excel (not Date type).

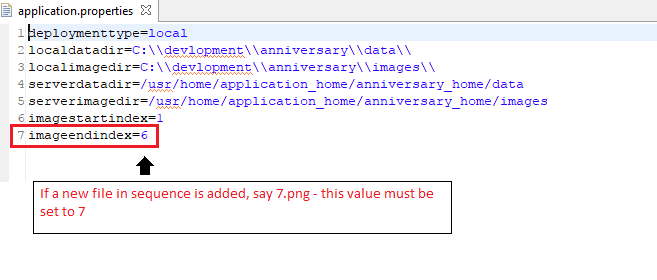
* 1. Custom messages are entered in a file – messages.properties in data folder. The format of the messages entered are as shown below. No compilation and deployment required for such change.



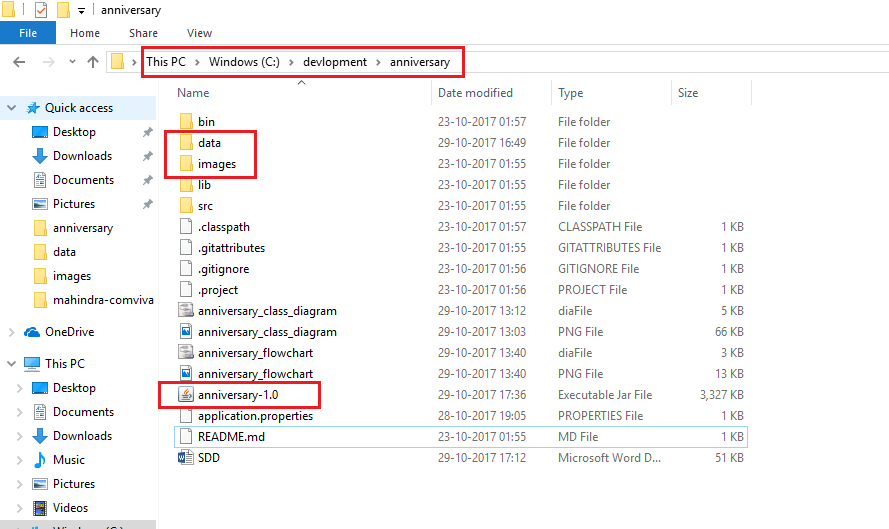
* 1. The sender’s email properties may be changed and entered in a file – email.properties in data folder. No compilation or deployment required for such changes.



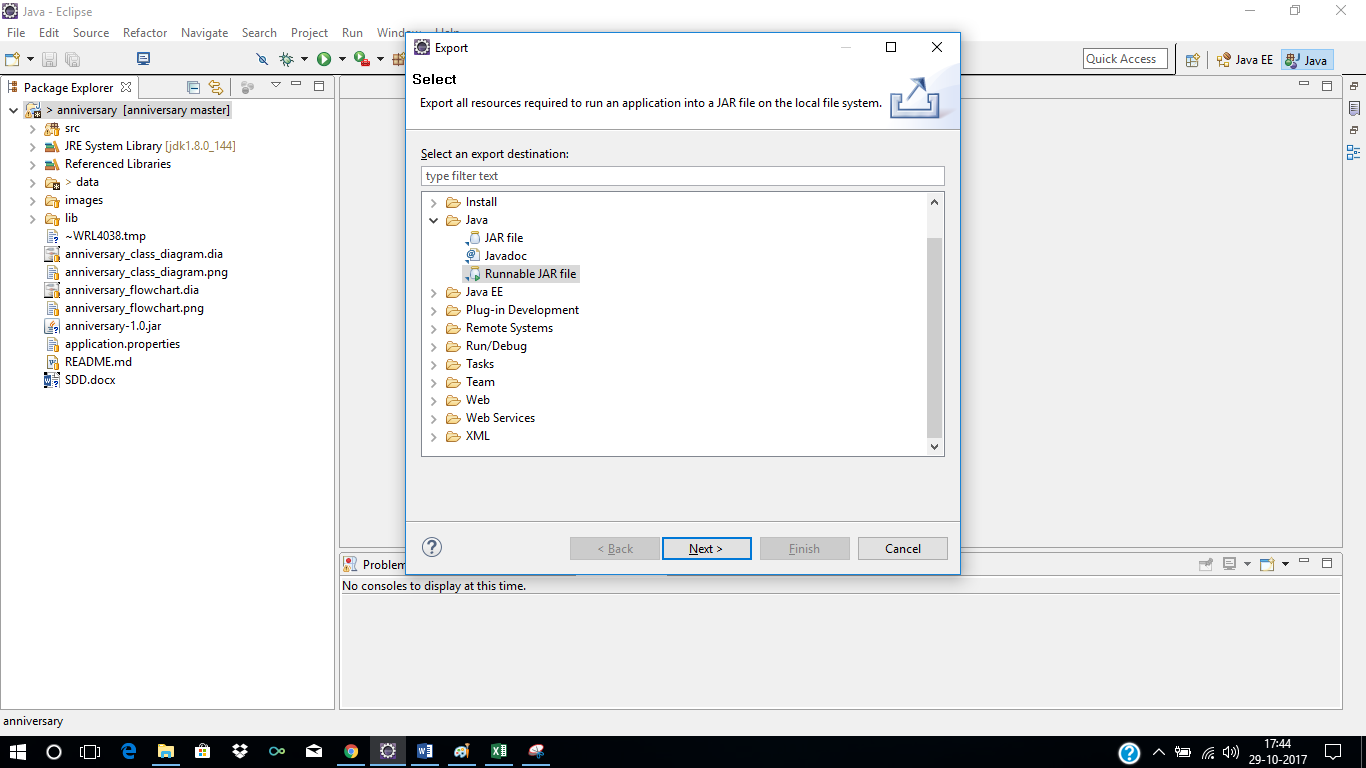
* 1. In case any new images are added in images folder – a change is required in application.properties file in ANNIVERSARY\_HOME location. The added file index number (which must be the max number in the image sequence) must be changed as shown below. There is no need to compile / deploy for this change.



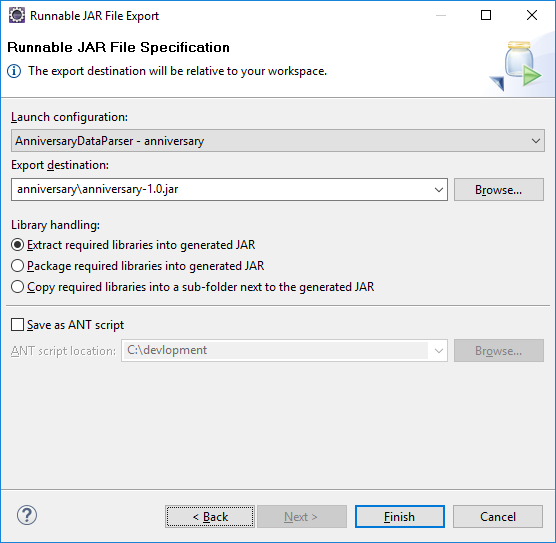
* 1. A sample ANNIVERSARY\_HOME structure may look as below:



* 1. To create runnable jar from eclipse – open the project in eclipse IDE > select the project in Project Explorer > Right click and select Export.

In the Export wizard, select Java > Runnable Jar File > 

Select the Class having the main() method – here AnniversaryDataParser class > Click Finish. Ignore the warnings, if any.



* 1. The JAR should be created in the specified location.