Institute of Technology Carlow
Kilkenny Road,
Carlow
Bsc (Hons) Cybercrime and IT Security

The Spam Catcher
Design manual

Author: Aya Aloraimi
Student Number: C00212086
Supervisor: James Egan

Abstract

The purpose of this project is to create a web application that filter spam emails by using supervised machine learning algorithm which is Naïve Bayes classifier. This classifier should classify email messages as spam or ham. The web application should present the user the options to scan their email messages and these options are scanning email content by uploading a text file or by copy the content of email message and paste it into the web application. Another option is by scanning the email header features like subject, from and to. After scanning the uploaded email message, the web application should display the result to the user to determine if their email messages is spam or ham.

Table of Contents

Table of Contents

Introduction

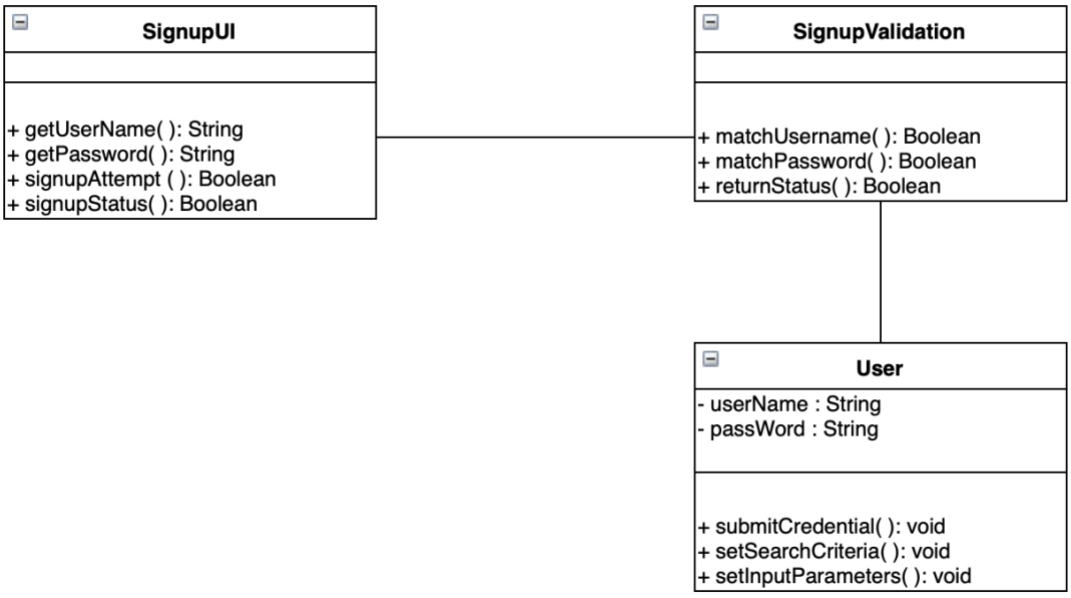
This document presents relevant UML diagrams such as class diagrams and sequence diagrams. Additionally, it provides User Experience (UX) which is considered with the entire process of acquiring and integrating a web application including design, usability and functions. It also will provide pseudo-code of implementing the spam filtering system in the spam catcher web application.

UML

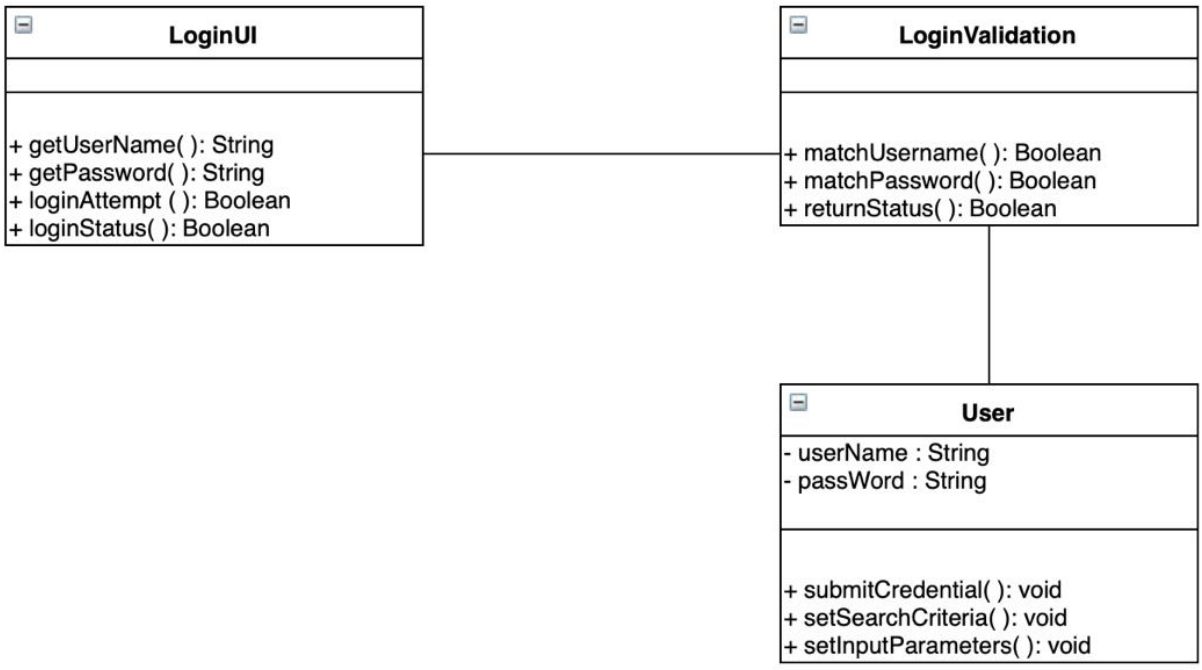
UML class diagram is used to describe the spam filtering system by identifying appropriate classes, attributes, methods, and relationships. Starting from with creating a diagram before creating the code itself helps to define a fundamental feature to be done during the time of the project. As shown, Figure1 and 2 describes classes in registration and login system. Moreover, Figure3 shows the main classes with attributes and methods of implementing the spam filtering system.

Class diagram

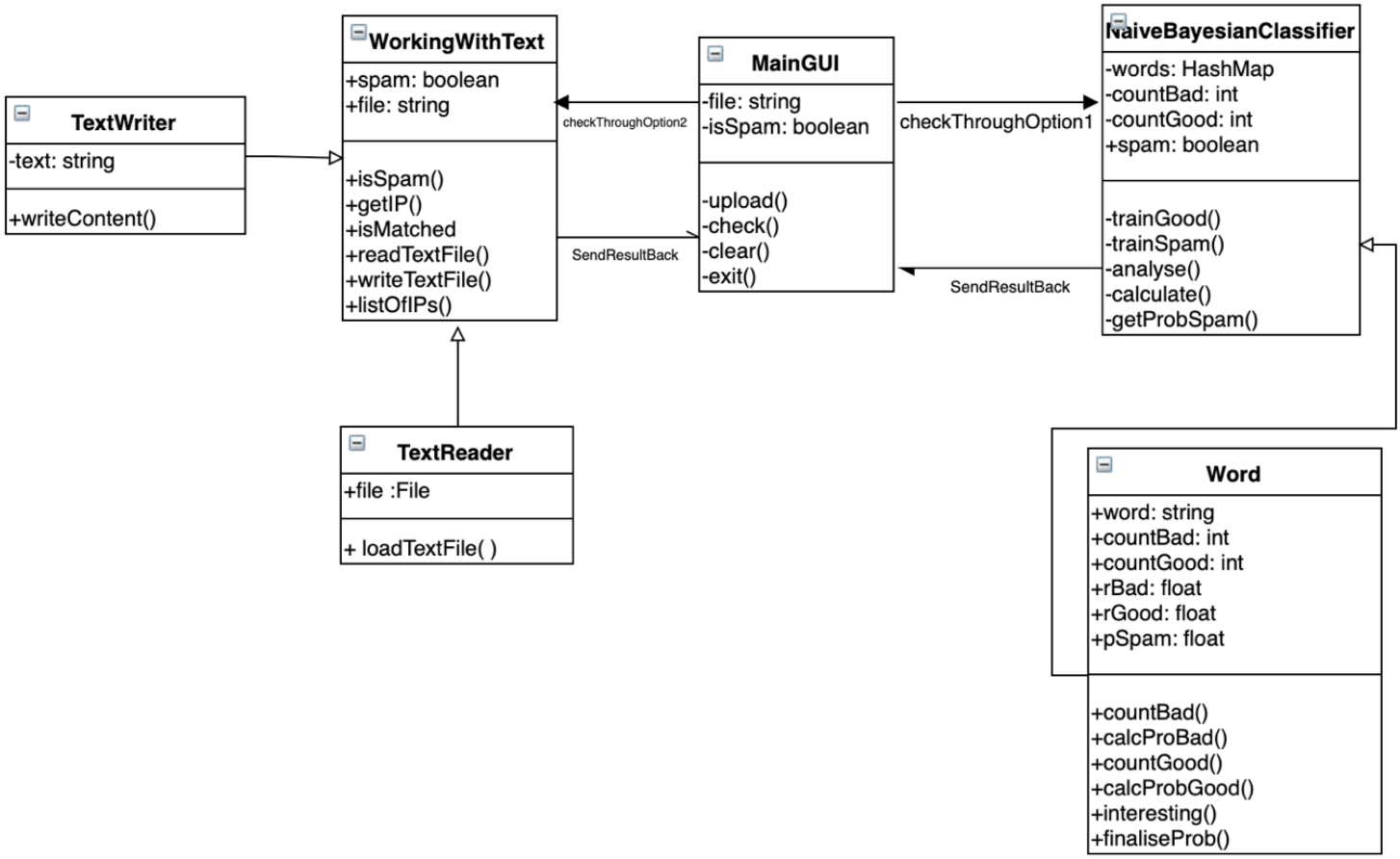
Registration class diagram:



Login class diagram

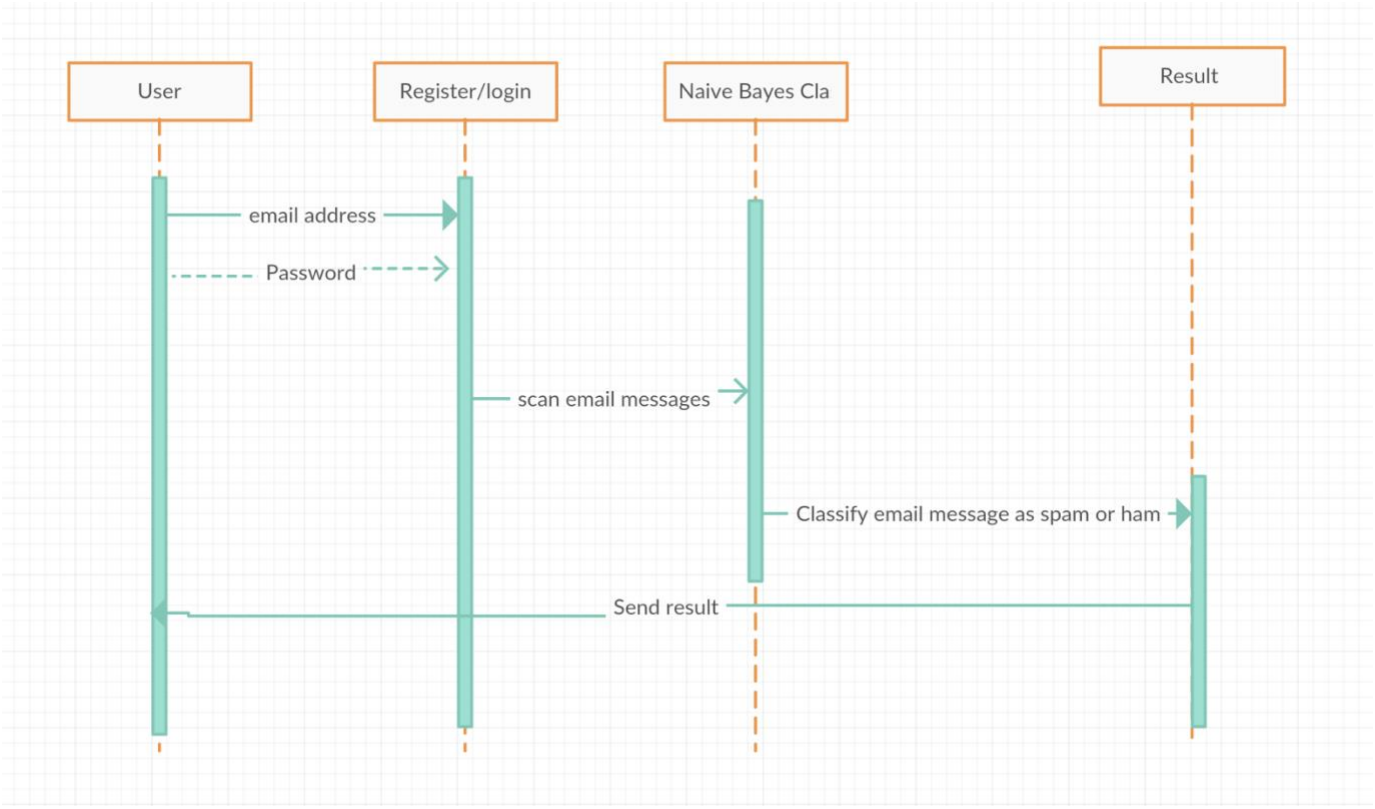


Email filtering system class diagram



Sequence diagram

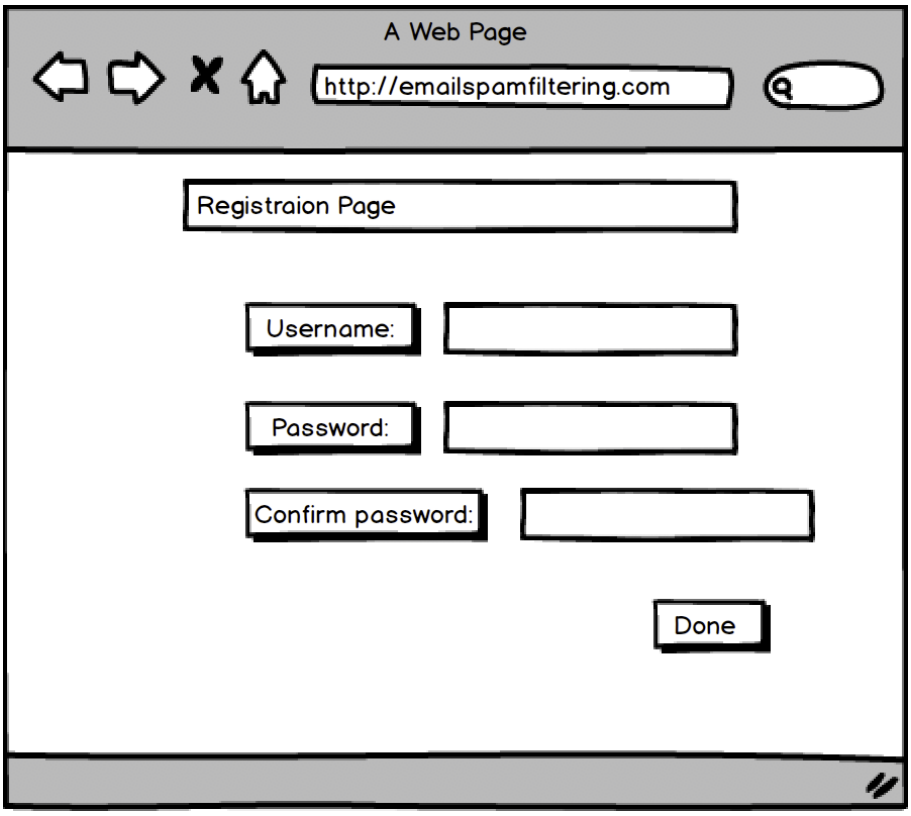
Email spam filtering system:



User Experience

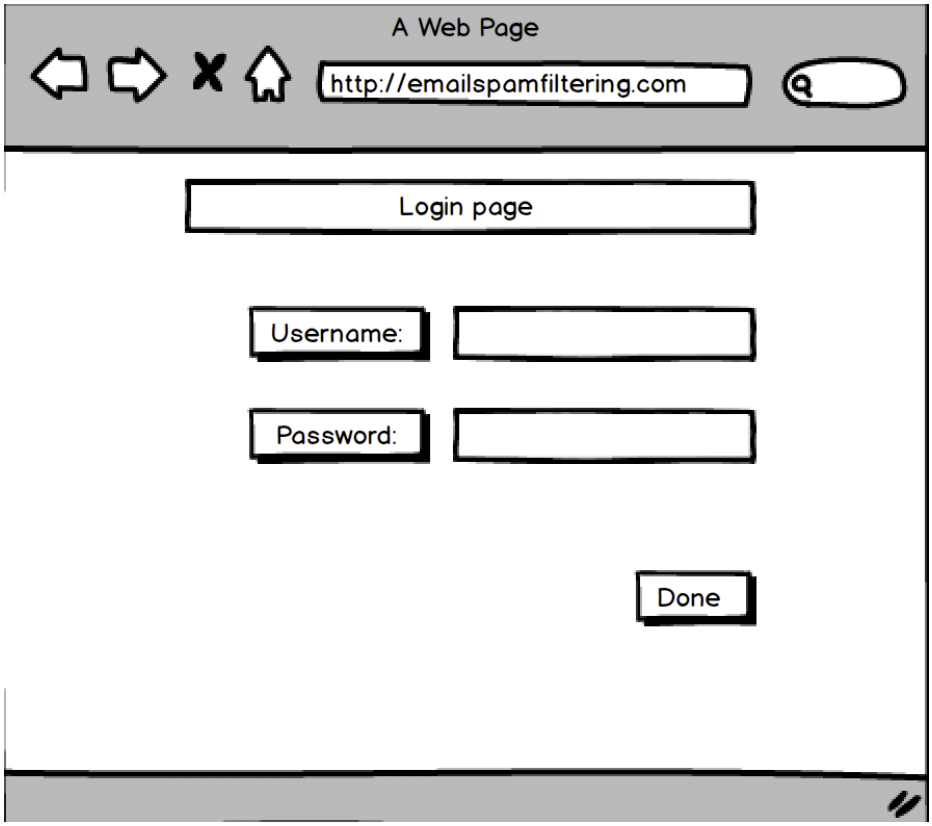
Registration Page

The user should sign up to use the web application's services

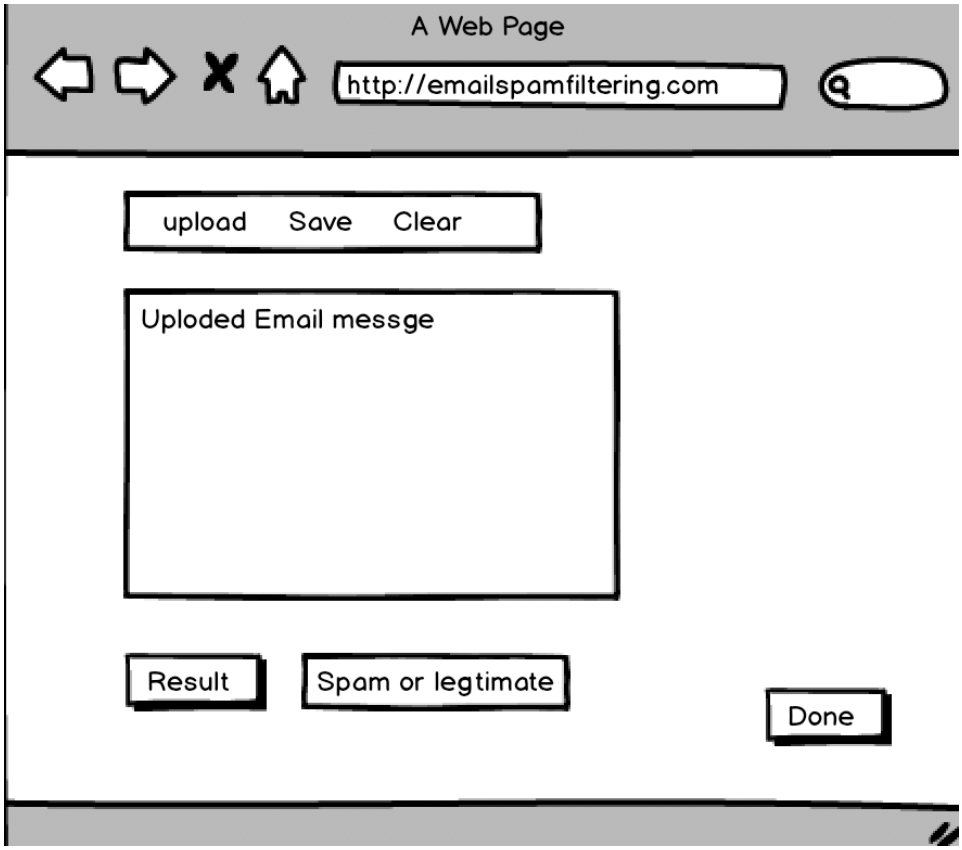


Login Page:

After registration, the user should login to validate their information and get access to the system



Scanning page



Naive Bayesian approach

Since spams are originally designed to trick people so spams are already had some typosquatting behaviors on the sender's email as well as some linguistical pattern in the content. For example, senders email can be info@goole.com but it almost looks like info@google.com. The Naïve Bayes classifier captures those linguistic characteristics and calculates the probability of being a spam email on both headers itself as well as the content of the email.

The process of classifying email messages is described below in pseudo-code for spam emails. Also the same method is used to non-spam messages, but instead of searching for spam messages, it will be changed to non-spam messages.

```
If word is in spam
        Increment the countSpam by 1;
If word is in non-spam
        Increment the countNonSpam by 1;
```

This work by counting the total number of words that appear in spam non-spam messages. Then the filter checks for the presence of words from the spam phrases dictionary that is created after gathering large number of spam email from dataset.

the same steps are repeated for non-spam messages

```
Load the message ;
                For each word
                    Convert to lower case;
                    If word is in the spam phrases dictionary
                        get it;
                    increase the number of appearance in spam;
                    calculate word's probability;
                otherwise
                increase the number of appearance in non spam;
                    calculate word's probability;
```