Email Spam and Non-Spam Classification

**Software Requirements Specification**

Version 1.0



**Group Id: F2002F5DD9**

**Supervisor Name : Muhammad Tayyab Waqar**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 11/12/2020 | 1.0 | Email Spam and Non-Spam Classification Project categorizes email messages into two heads Spam and Non-spam using a Python based text classification technique. Furthermore, the project also includes accuracy, time and error rate calculation using suitable algorithms on an Email Dataset along with algorithms comparison to identify the best suitable model for the use case. | MC190402102 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. [Scope (of the project)](#scope)
2. [Functional Requirements Non Functional requirements](#FRNFR)
3. [Use Case Diagram](#UCD)
4. [Usage Scenarios](#UCS)
5. [Adopted Methodology](#Adopted)
6. [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**SRS Document**

**Scope of Project:**

Email becomes a powerful tool for communication as it saves a lot of time and cost. It is one of the most popular and secure medium for online transferring and communication messages or data through the web. But, due to the social networks, most of the emails contain unwanted information which is called spam. These unsolicited emails have a high cost in terms of time, storage space, network bandwidth consumption and indirect costs to protect privacy and security breaches. To identify such spam email is one of the important challenges.

The purpose of the Email Spam and Non-Spam Classification Project is to understand step by step working of the spam filter and how it helps in making everyone life easier. The project is based on binary classification of email messages as spam or not spam and will use Python text classification technique to identify or classify email spam message. Furthermore, the project will also contain accuracy, time and error rate found by applying suitable algorithms on Email Dataset along with algorithm comparisons to identify the best algorithm for text classification.

**Functional and non Functional Requirements:**

**Functional Requirements:**

Administrator will perform all these tasks.

1. **Collect Data Set**

• Gathering the data for Email spam contains spam and non-spam messages

1. **Pre-processing**

• As most of the data in the real world are incomplete containing noisy and missing values. Therefore we have to apply Pre-processing on your data.

1. **Feature Selection**

• After the pre-processing step, we apply the feature selection algorithm, the algorithm which deploy here is Best First Feature Selection algorithm.

1. **Apply Spam Filter Algorithms.**

• **Handle Data**: Load the dataset and split it into training and test datasets.

• **Summarize Data**: summarize the properties in the training dataset so that we can calculate probabilities and make predictions.

• **Make a Prediction**: Use the summaries of the dataset to generate a single prediction.

• **Make Predictions:** Generate predictions given a test dataset and a summarized training dataset.

• **Evaluate Accuracy**: Evaluate the accuracy of predictions made for a test dataset as the percentage correct out of all predictions made.

1. **Train & Test Data**

• Split data into 70% training & 30% testing data sets.

1. **Confusion Matrix**

• Create a confusion matrix table to describe the performance of a classification model.

1. **Accuracy**

• Find Accuracy of all algorithm and compare.

**Non-Functional Requirements:**

**1. Data Availability:**

• Ensuring high availability of emails data.

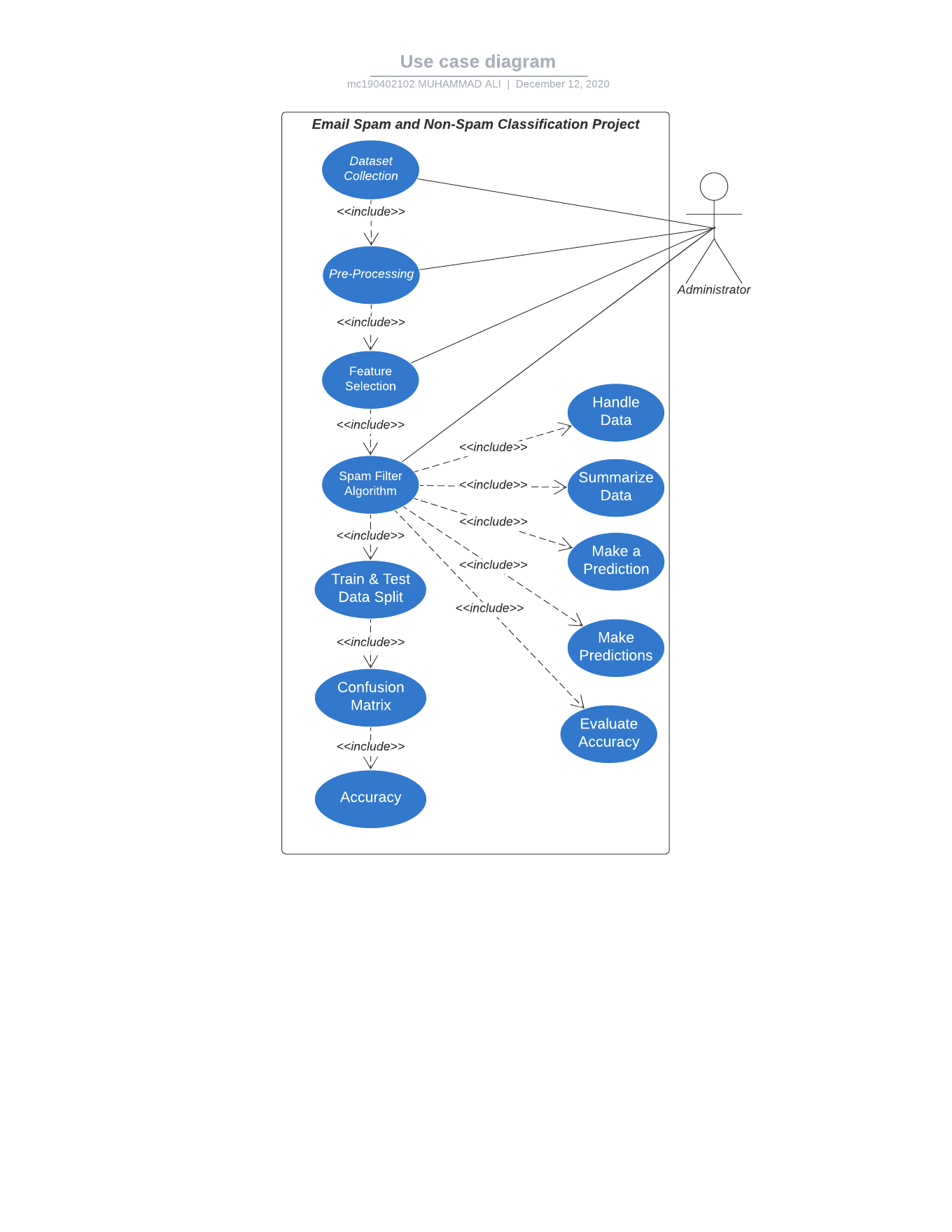
**2. Efficient Utilization:**

• Ensuring the resource utilization must be made efficiently.

**Commenting:**

• The project files should include proper commenting in order to facilitate the code viewer to easily understand the whole flow and each items purpose.

**Use Case Diagram(s):**



**Usage Scenarios:**

Usage Scenarios 1:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Data Collection |
| Use Case Id | UC-01 |
| Action | Collecting Dataset |
| Description | This use case provides a dataset to the system which serves as a raw data input. |
| Alternative Paths | Instead of collecting an available dataset, one is made manually from the scratch. |
| Precondition | There must be a dataset available containing both spam and non-spam messages. |
| Post-Condition | The system will have the dataset as an unprocessed data. |
| Author/Actor | Administrator |
| Exception | Dataset is not available anywhere, in this case one will be created from the very scratch by collecting both spam and non-spam messages. |

Usage Scenarios 2:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Pre-processing |
| Use Case Id | UC-02 |
| Action | Pre-processing the dataset. |
| Description | This use case pre-processes the dataset in order to transform the email message to be in form that is suitable for feature selection use case to have as an input. |
| Alternative Paths | Some pre-processing function of a python library gets deprecated so it will raise deprecation warning but won’t affect the system flow. |
| Precondition | Dataset must be provided for pre-processing |
| Post-Condition | The system will have the email message in the pre-processed form, ready to be forwarded into feature extraction use case. |
| Author/Actor | Administrator |
| Exception | The machine on which the system will be deployed does not have python installed. |

Usage Scenarios 3:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Feature Selection |
| Use Case Id | UC-03 |
| Action | Selecting relevant features from the pre-processed dataset. |
| Description | This use case extracts out the important features from the dataset in order to forward only those words to the spam filter algorithm that will contribute the most in the classification of spam and non-spam emails. |
| Alternative Paths | Some feature selection function of a python library gets deprecated so it will raise deprecation warning but won’t affect the system flow. |
| Precondition | Dataset must be of pre-processed form. |
| Post-Condition | The spam filter algorithm will be ready to have extracted features to be used in classifying spam and non-spam emails. |
| Author/Actor | Administrator |
| Exception | An email message containing only non-important words like articles which should logically lead to the removal of all the words |

Usage Scenarios 4:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Spam Filtering Algorithm |
| Use Case Id | UC-04 |
| Action | Applying classification technique on the selected features. |
| Description | This use case classifies the dataset into spam or non-spam using selected features as input and text classification techniques. |
| Alternative Paths | Some function of a python based machine learning library gets deprecated so it will raise deprecation warning but won’t affect the system flow. |
| Precondition | The selected features must includes words that convey meaning and should not include words that are meaning-less like articles. |
| Post-Condition | The algorithm will provide a single result of whether the message is spam or non-spam. |
| Author/Actor | Administrator |
| Exception | The spam filtering algorithm fails identify the spam message and incorrectly classifies as non-spam (false positive). |

Usage Scenarios 4.1:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Handle Data |
| Use Case Id | UC-4.1 |
| Action | Loading the dataset and split it into training and test datasets. |
| Description | This use case loads the dataset into spam filter algorithm and splits it into training and test datasets. |
| Alternative Paths | If no pre-built library function will not be available for loading dataset then we will load it using self-written python code. |
| Precondition | The dataset must be available for loading the data into algorithm. |
| Post-Condition | The algorithm will have the dataset loaded into it. |
| Author/Actor | Administrator |
| Exception | The dataset fails to be loaded into the dataset. |

Usage Scenarios 4.2:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Summarize Data |
| Use Case Id | UC-4.2 |
| Action | Summarizing Data |
| Description | This use case summarize the properties in the training dataset so that we can calculate probabilities and make predictions. |
| Alternative Paths | Instead of using pre-built python function to get the summary of all the properties in the training dataset we write the code from the very scratch. |
| Precondition | The dataset must be loaded into algorithm. |
| Post-Condition | We will be able to calculate probabilities and make predictions |
| Author/Actor | Administrator |
| Exception | The dataset fails to be summarized. |

Usage Scenarios 4.3:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Make a Prediction |
| Use Case Id | UC-4.3 |
| Action | Making a Prediction |
| Description | This use case uses the summaries of the training dataset to generate a single prediction. |
| Alternative Paths | Instead of using summaries we take the whole dataset features to make the prediction. |
| Precondition | The dataset must be in a summarized form. |
| Post-Condition | We will get a single prediction. |
| Author/Actor | Administrator |
| Exception | The algorithm fails to predict accurately based on summaries (the training loss is very high) |

Usage Scenarios 4.4:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Make Predictions |
| Use Case Id | UC-4.4 |
| Action | Making Predictions |
| Description | Generating predictions given a test dataset and a summarized training dataset. |
| Alternative Paths | Instead of using summarized dataset we take the whole dataset features and divide them into two to make the predictions. |
| Precondition | The dataset must be available into two groups i.e. train and test datasets. |
| Post-Condition | We will get multiple prediction outputs |
| Author/Actor | Administrator |
| Exception | The algorithm fails to generalizes the predictions. |

Usage Scenarios 4.5:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Evaluate Accuracy |
| Use Case Id | UC-4.5 |
| Action | Evaluating Preconditions Accuracy |
| Description | Evaluate the accuracy of predictions made for a test dataset as the percentage correct out of all predictions made. |
| Alternative Paths | Instead of checking correct ones we check the percentage incorrect out of all predictions made. |
| Precondition | The algorithm must be generating predictions. |
| Post-Condition | We will get accuracy of predictions made for a test dataset as as the percentage correct out of all the predictions made. |
| Author/Actor | Administrator |
| Exception | The algorithm produces very low correct predictions rate i.e. high false positives and low true positives. |

Usage Scenarios 5:

|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Train & Test Data |
| Use Case Id | UC-05 |
| Action | Spliting the dataset into train and test datasets. |
| Description | This use case splits data into 70% training & 30% testing data sets |
| Alternative Paths | The dataset is divided into 50% for train and 50% for test. |
| Precondition | The algorithm must be generating predictions with fair accuracy. |
| Post-Condition | We will get 70:30 ratio based division of our dataset into train and test datasets respectively. |
| Author/Actor | Administrator |
| Exception | The datapoints i.e. email messages in the dataset are in odd number therefore, dividing it into 70:30 will likely result in loss of some datapoints. |

Usage Scenarios 6:

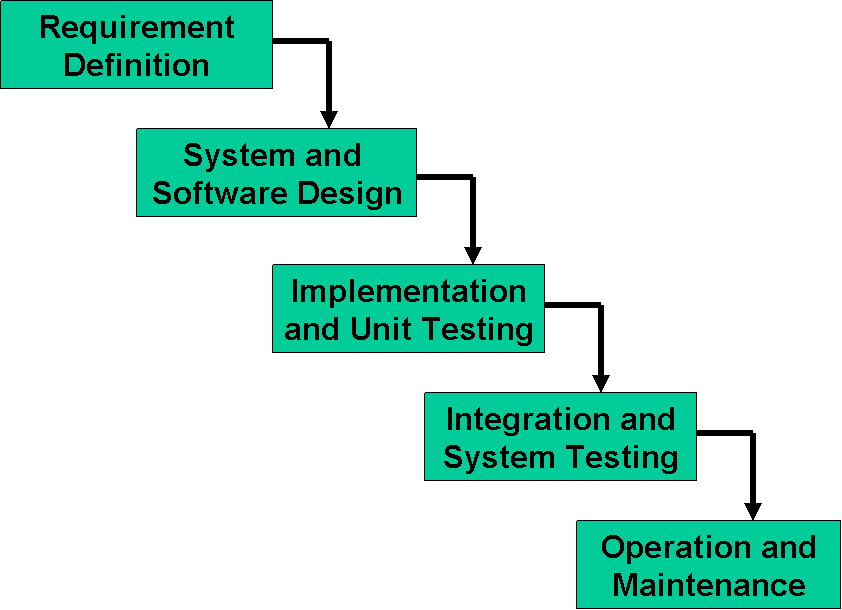
|  |  |
| --- | --- |
| USE CASE SCENARIO | |
| Use Case Title | Confusion Matrix |
| Use Case Id | UC-06 |
| Action | Creating a Confusion Matrix |
| Description | This use case creates a confusion matrix table to describe the performance of a classification model. |
| Alternative Paths | We create confusion matrix with summarized training dataset based predictions instead of testing dataset based predictions. |
| Precondition | The dataset must be divided into two parts train and test dataset and we also have predictions. |
| Post-Condition | The Confusion Matrix will be created for further analysis. |
| Author/Actor | Administrator |
| Exception | The confusion matrix shows high value for false positives and low value for true positives. |

Adopted Methodology:

* Waterfall Model
* Spiral Model
* VU Model

WaterfallModel

The first published model of the software development process was derived from other engineering processes. Because of the cascade from one phase to another, this model is known as the waterfall model. This model is also known as linear sequential model. This model is depicted in the following diagram.



Spiral Model

This model was developed by Barry Boehm. The main idea of this model is to avert risk as there is always an element of risk in development of software. For example, key personnel may resign at a critical juncture, the manufacturer of the software development may go bankrupt, etc.

In its simplified form, the Spiral Model is Waterfall model plus risk analysis. In this case each stage is preceded by identification of alternatives and risk analysis and is then followed by evaluation and planning for the next phase. If risks cannot be resolved,

Risk Analysis

Verify

Rapid Prototype

Specification

Design

Implementation

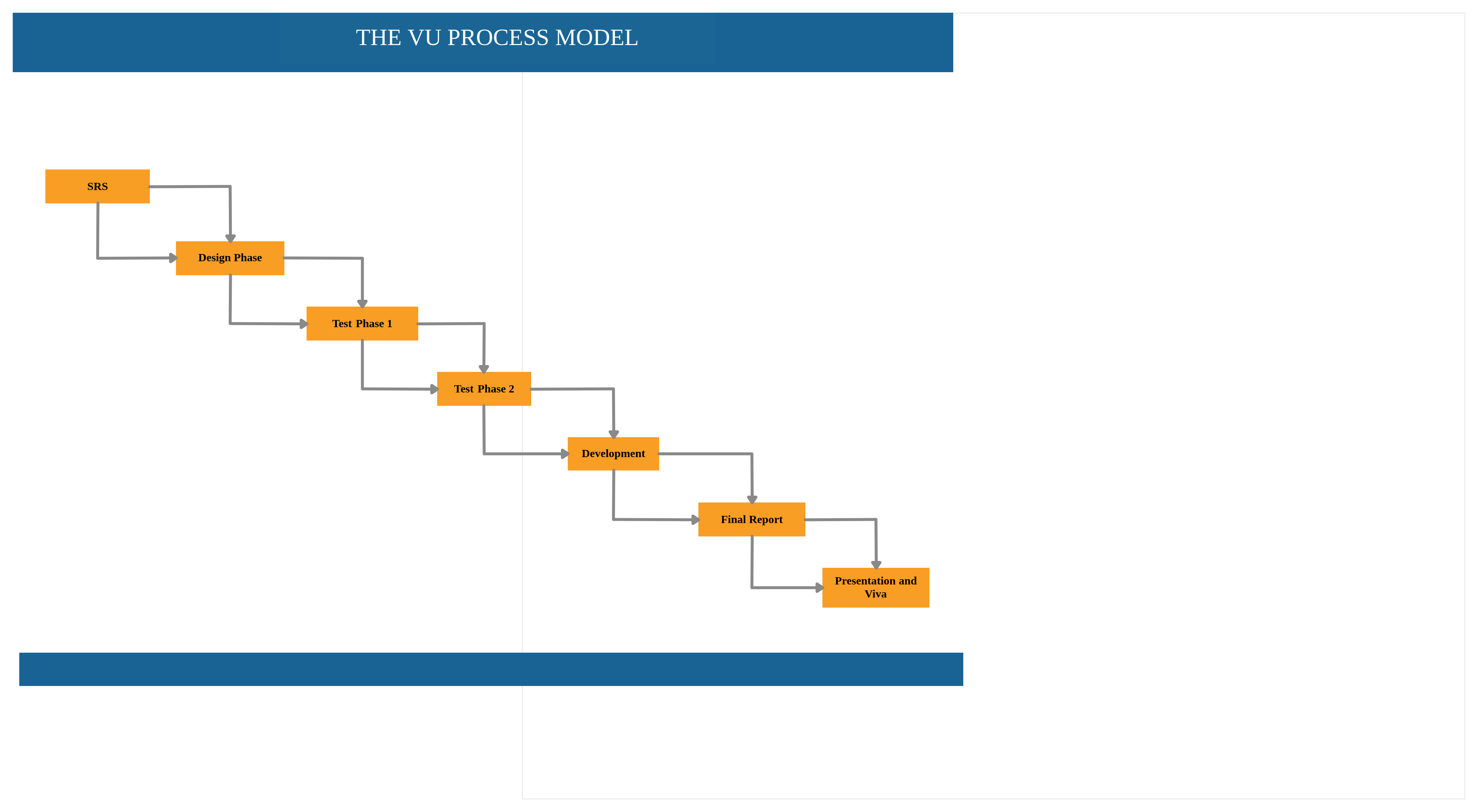
project is immediately terminated. This is depicted in the following diagram.

VU Process Model: (Adopted Methodology)

I have studied both the above methodologies and then I have decided to choose the VU Process Model for my project. University also guides me that you must use Virtual University Process Model for the project as it is a combination of both waterfall and spiral model and it fulfills all the project requirements. Therefore, I chose VU Process Model. The detail of VU Process Model is as under:

1. System Requirement Specification (SRS)
2. Design Phase
3. Test Phase 1
4. Test Phase 2
5. Development
6. Final Report
7. Presentation and Viva

Diagram of Project Model:



Reasons for adopting Methodology

I choose VU Process Model for my project. Reasons of adopting this methodology is as under:

1. First of all Virtual University of Pakistan guides us that you must select VU Process Model for the project.
2. Second reason is that only VU Model fulfills all the requirements of my project.
3. VU Process Model is the combination of Waterfall and Spiral Methodologies. One feature of waterfall methodology is the sequence of steps that we take in our methodology and one feature of spiral methodology is to go previous steps which is also included in our methodology.

Team Structure

The project Email Spam and Non-Spam Classification includes two members, one Supervisor and one Developer.

**Supervisor**

**Developer**

Work Plan (Use MS Project to create Schedule/Work Plan)

