High Level Design (HLD)

**Spam Ham Classifier**

Revision Number: 1.0

Last Date of Revision: 21/12/2022

# **Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 21/12/2022 | 1 | Initial HLD – V1.0 | Himanshu Goswami |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Contents**

Document Version Control……….……………………………………….……………………….……………………………….…….2

Abstract………………..………………………………………………………………………………………………………………………..…4

1. Introduction…………………….……………………………………………………………………..…………………………………..5
   1. Why this High-Level Design Document? .…………………………………….……………..……………………….5
   2. Scope……………….………………………………………………………………………….…………………………..………….5
   3. Definitions……..……………………………………………………………………………………………………………..…….5
2. General Description…………………………………………………………………………………………………………………….6
   1. Product Perspective…………………………………………………………………………………………………………….6
   2. Problem Statement……………………………………………………………………………………………………………..6
   3. Proposed Solution……………………………………………………………………………………………………………….6
   4. Further Improvements…………………………………………………………………………………………………………6
   5. Data Requirements………………………………………………………………………………………………………………6
   6. Tools Used……………………………………………………………………………………………………………………………6
   7. Constraints…………………………………………………………………………………………………………………………..7
   8. Assumptions………………………………………………………………………………………………………………………..7
3. Design Details……………………………………………………………………………………………………………………………..8
   1. Process Flow………………………………………………………………………………………………………………………..8
      1. Model Training and Evaluation…………………………………………………………………………………8
      2. Deployment Process…………………………………………………………………………………………………9
   2. Event Log……………………………………………………………………………………………………………………………..9
   3. Error Handling……………………………………………………………………………………………………………………..9
4. Performance……………………………………………………………………………………………………………………………..10
   1. Reusability…………………………………………………………………………………………………………………………10
   2. Application Compatibility…………………………………………………………………………………………………..10
   3. Deployment……………………………………………………………………………………………………………………….10
5. Web App……………………………………………………………………………………………………………………………………10
6. Conclusion…………………………………………………………………………………………………………………………………11
7. References…………………………………………………………………………………………………………………………………11

# **Abstract**

Spam text messages and Spam Emails have become a major annoyance in our day-to-day life. According to the reports, the problem has reached unprecedented levels: -

* [RoboKiller.com](https://www.jdoqocy.com/cl116efolfn2AC58CA6248354833), a spam and robocall blocking service, recently released Its[2021 Phone Scam Insights](https://www.robokiller.com/robocall-insights/robokiller_yearly_phone_report_2021.pdf) which show that Americans received 87.8 billion spam text messages last year. That is the most ever and a 58% increase compared to the previous year.
* According to Truecaller.com, a staggering multi-billion dollars were lost to spam SMS and emails in America alone in past 12 months.
* Every year over hundred million people worldwide fall victim to a phone scams.

This work discusses the implementation of the increasingly reliable and robust antispam filters.

# **Introduction**

## **Why this High-Level Design Document?**

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* Present all of the design aspects and define them in detail.
* Describe the user interface being implemented.
* Describe the hardware and software interfaces.
* Describe the performance requirements
* Include design features and the architecture of the project
* List and describe the non-functional attributes like:
  + Security
  + Reliability
  + Maintainability
  + Portability
  + Reusability
  + Application Compatibility
  + Serviceability

## **Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

## **Definitions**

|  |  |
| --- | --- |
| *Term* | *Description* |
| Spam | Any kind of unwanted digital communication that gets sent out in bulk. |
| Ham | Non-Spam or Good Mail |
| Database | Collection of all the information monitored by this system |
| IDE | Integrated Development Environment |

# **General Description**

## **Product Perspective**

The Spam Ham Classifier is a machine learning based text reading model which will help us in detecting and filtering spam emails.

## **Problem statement**

To create an AI solution (web app) to detect whether a text is a ham or spam.

## **Proposed Solution**

The solution proposed here is a Spam Ham Classifier which will be a web application. Whenever a user will input a text in the given field in web app, within a second, the user will get to know whether the given text is a spam or ham.

## **Further Improvements**

Spam Ham Classifier web app can be integrated into websites or software commonly used by marketers to help them write better SMS or emails which can save much of their time and money by making sure their text will not be categorized as spam by any web-based email service.

## **Data Requirements**

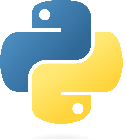
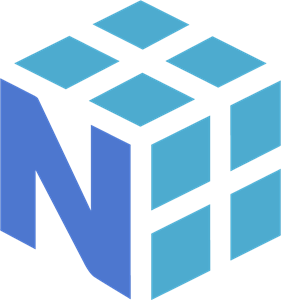
Data requirement completely depend on our problem statement. We just require a dataset with following information: -

* Label -> This column will specify whether the text is spam or ham.
* Message -> This column will have the complete text which is defined as spam or ham in first column.

Data can be in a table formatted file or database. Also, it can be in JSON format in NOSQL database too. Using various pythonic libraries we can easily interpret the data.

## **Tools Used**

Python programming language and frameworks such as NumPy, Pandas, Scikit-Learn and Natural Language Toolkit are used to build the whole model.





* PyCharm is used as IDE.
* For visualization of the plots, Matplotlib and Seaborn are used.
* Streamlit Cloud is used for deployment of model.
* MongoDB is used to retrieve, insert and update the database.
* Front end is handled by Streamlit package.
* GitHub is used as version control system.

## **Constraints**

The Spam Ham Classifier solution web app must be user friendly, as automated as possible and users should not be required to know any of the workings.

## **Assumptions**

The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement). Machine Learning based text classifier model is used to detect and filter spam texts based on the input data. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

# **Design Details**

## **Process Flow**

For identifying the text and transforming it, we will use a machine learning base model. Below is the process flow diagram.

## **Proposed Methodology**

ML Model for Text Classification

Training/Validation on Data

User Inputs Text on App

## 3.1.1 Model Training & Evaluation

Display result to user on App

Detection of Spam/Ham Text

Prediction

User Input

Training

Training Set



Result of Evaluation

Model

Evaluation

Test Set

Ingestion

Database

Split

Dataset Ingested

Validation Set

Prediction

Validate New Data

## 3.1.2 Deployment Process

Load Model

Get Text from user on app

Text Transformation

Predicted Result

Make Prediction

## **Event Log**

The system should log every event so that the user will know what process is running internally.

**Initial Step-By-Step Description:**

1. The system identifies at what step logging required.
2. The system should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## **Error Handling**

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

# **Performance**

The Spam Ham Classifier solution is used for detection and filtration of spam emails or SMS. It will give result to the user within a second as accurate as possible so that it will not mislead the user. Also, model retraining is very important to improve the performance.

## **Reusability**

The code written and the components used should have the ability to be reused with no problems.

## **Application Compatibility**

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

## **Deployment**



# **Web App**

Web App will be implemented to display a text field & a button. User can input text and in one click the user will get the prediction whether the text is spam or ham. We will use Streamlit Cloud service which allows us to get a clean & quick responsive dashboard which will enhance the efficiency of web app.

# **Conclusion**

The Spam Ham Classifier will be an AI based web app which will detect and filter the spam / ham text provided by the user. Identifying a text whether it’s Spam or Ham can greatly help the marketers or individuals know whether the text they have is a Spam or Ham in general and can take action based on the same data.

# **References**

1. [www.truecaller.com](https://www.truecaller.com/blog/insights/truecaller-insights-2022-us-spam-scam-report) for Spam Insights
2. [www.clark.com](https://clark.com/scams-rip-offs/spam-text-messages/) for Spam insights.