Using Machine Learning to Help Detect Sensitive Information

**Design Document CDAC, Bangalore**

# Version <version No.>

**<date of creation>**

# Prepared By:

Ratnesh Kumar-240350125058 Harshit Sehgal-240350125031

Abhishek Kumar Mishra-240350125002 Paras Jamkar-240350125047

Prathima Tiwari S.- 240350125056

|  |  |
| --- | --- |
| [Introduction](#_bookmark0) | 4 |
| * [Purpose](#_bookmark1) | 4 |
| * [Scope](#_bookmark2) | 4 |
| * [Intended Audience](#_bookmark3) | 4 |
| * [References](#_bookmark4) | 4 |
| [Acronyms, terms and definitions](#_bookmark5) | 4 |
| [Assumptions and constraints](#_bookmark6) | 4 |
| [Basic Design approach](#_bookmark7) | 5 |
| [Risks](#_bookmark8) | 5 |
| [System overview](#_bookmark9) | 5 |
| [Architecture Design](#_bookmark10) | 5 |
| [Data Design](#_bookmark11) | 5 |
| [Component Design](#_bookmark12)  [**Work-flow Summary w.r.t Identifying Components : -**](#_h7xhv885g5eq) **26**  Data Collection  Data Preprocessing  Model Selection and Training  Model Evaluation  Integration and Deployment  Detection and Monitoring  Reporting and Visualization  Continuous Improvement  Summary of Workflow Steps | 5 |

[**Design Test**](#_bookmark15)8

[**Cross Reference with System Requirement Specification**](#_bookmark16)8

# Introduction

This project involves designing a system that utilizes machine learning to identify and handle sensitive information within a document.

## Purpose

The purpose of this document is to give a detailed design for the ***Using Machine Learning to Help Detect Sensitive Information***. It shall explain the architecture of the system, component design, data design and UI design. This document is primarily intended for developers to develop and meet the functionality requirements. This document provides development and deployment architectures, database design, details for development of APIs and UI development.

## Scope

This design document pertains to the implementation for the ***Using Machine Learning to Help Detect Sensitive Information***. Scalability and extensibility at the component level are considered but not tested for practical implementation.

## Intended Audience

This document is intended for the design, development, and testing teams and should be referred to at all stages of the project.

## References

#### Functional Requirement Specifications (FRS)

* + - [https://blog.developer.adobe.com/using-machine-learning-to-help-](https://blog.developer.adobe.com/using-machine-learning-to-help-detect-sensitive-information-5bfb32eeb34e)

[detect-sensitive-information-5bfb32eeb34e](https://blog.developer.adobe.com/using-machine-learning-to-help-detect-sensitive-information-5bfb32eeb34e) (by - Renae Kang)

# Acronyms, terms and definitions

|  |  |
| --- | --- |
| FRS | Functional requirements specification |
| User | User refers to department/employee/worker |
| API | Application Programming Interface |
| PII | Personally Identifiable Information |

1. **Assumptions and constraints**

#### The application follows the API based development which requires a subscription by the user to make use of provided functionality.

* + The project assumes that data used for training and detection adheres to data protection regulations.

# Basic Design approach

* + - The project follows a Structured Design approach. The elements are identified from the FRS and organized into APIs to handle related functions independently, with minimal interdependence.

# Risks

#### Risks in the project eg personal data handling

* + - Handling of sensitive data, ensuring data privacy and compliance with regulations such as GDPR.
    - Performance and accuracy of the machine learning model in identifying sensitive information.

# System overview

The system detects and manages sensitive information within datasets using machine learning. It includes components for data ingestion, preprocessing, model training, detection, and reporting.

# Architecture Design

## Architecture Diagram

### Explanation of Components in the Architecture Diagram:

* + - **Data Ingestion**: Module for ingesting data from various sources.
    - **Preprocessing**: Cleaning and preparing data for model training and detection.
    - **Model Training**: Training machine learning models to identify sensitive information.
    - **Detection**: Using trained models to detect sensitive information in new datasets.
    - **Reporting**: Generating reports based on detected information.

# Data Design

## Database Structure

#### Data is stored in master tables, logs, and transaction tables.

**Database ER Diagram**: Provided in the document.

### Database Tables:

* **Table Name**: Description
* **Idx (Primary Key)**: Field Name, Data Type, Description

# Component Design

* **Data Ingestion**: Process and activity diagrams.
* **Preprocessing**: Steps followed in component implementation.
* **Model Training**: Activity and use case diagrams.
* **Detection**: Detailed explanation and diagrams

# 10.0 Workflow Summary

* **Data Collection**

## Activities:

* + Identify sources of data (databases, logs, files).
  + Extract data from these sources.
  + Ensure data is collected in a secure and compliant manner.

## Outputs:

* + Raw data sets ready for preprocessing.

# Data Preprocessing

## Activities:

* + Clean the data (remove duplicates, handle missing values).
  + Transform the data (normalize, anonymize).
  + Split the data into training and testing sets.

## Outputs:

* + Preprocessed data sets ready for training and testing.

# Model Selection and Training

## Activities:

* + Select appropriate machine learning algorithms (e.g., NLP models for text analysis).
  + Train models using the training data set.
  + Validate the models using cross-validation techniques.

## Outputs:

* + Trained machine learning models.
  + Performance metrics and validation results.

# Model Evaluation

## Activities:

* + Evaluate model performance on the testing data set.
  + Fine-tune model parameters to improve accuracy and precision.
  + Ensure the model meets predefined accuracy thresholds for sensitive information detection.

## Outputs:

* + Finalized machine learning model.
  + Evaluation reports detailing model performance.

# Integration and Deployment

## Activities:

* + Integrate the model into the existing system architecture.
  + Develop APIs for data input and output handling.
  + Deploy the model to a production environment.

## Outputs:

* + Deployed model accessible via APIs.
  + Integration documentation.

# Detection and Monitoring

## Activities:

* + Use the deployed model to scan new data for sensitive information.
  + Monitor the model's performance in real-time.
  + Implement alerts for detected sensitive information.

## Outputs:

* + Reports on detected sensitive information.
  + Alerts and notifications for stakeholders.

# Reporting and Visualization

## Activities:

* + Generate detailed reports on the detected sensitive information.
  + Visualize data and findings through dashboards and charts.
  + Provide insights and recommendations based on detection results.

## Outputs:

* + Detailed reports and visualizations.
  + Insights for improving data security practices.

# Continuous Improvement

## Activities:

* + Collect feedback from users and stakeholders.
  + Regularly update the model with new data and retrain as necessary.
  + Implement improvements based on feedback and new research.

## Outputs:

* + Updated and improved models.
  + Documentation of changes and improvements.
* Summary of Workflow Steps

1. **Data Collection**: Gathering data from various sources.
2. **Data Preprocessing**: Cleaning and preparing data for training.
3. **Model Selection and Training**: Choosing algorithms and training models.
4. **Model Evaluation**: Assessing model performance.
5. **Integration and Deployment**: Integrating models into systems and deploying them.
6. **Detection and Monitoring**: Using models to detect sensitive information.
7. **Reporting and Visualization**: Generating reports and visualizing findings.
8. **Continuous Improvement**: Updating models and processes based on feedback and new data.

# Interface Design

GUI layouts for each interface in the application, including user interaction diagrams.

# Any Specific Design Considerations

The system is designed for a limited number of simultaneous processing services provided by the APIs. Functional testing with required security controls through the API gateway is considered

# Design Test

<Testing Team need to Update this section>

# Cross Reference with System Requirement Specification

**Component Name**: Mapping with Functional Requirement ID, adherence details

* + **Appendix –** Detailed Information

Additional details and extended explanations relevant to the design document.