Phase 1: Problem Definition and Design Thinking

Problem Definition:

The problem is to build an AI-powered spam classifier that can accurately distinguish between spam and non-spam messages in emails or text messages. The goal is to reduce the number of false positives (classifying legitimate messages as spam) and false negatives (missing actual spam messages) while achieving a high level of accuracy.

Design Thinking:

Data: Obtain labeled spam and non-spam datasets, e.g., Kaggle.

Preprocessing: Clean and tokenize text (lowercase, remove special characters).

Feature Extraction: Convert tokens to numerical features using TF-IDF.

Models: Experiment with Naive Bayes, SVM, deep learning.

Evaluation: Assess accuracy, precision, recall, F1-score.

Improvement: Fine-tune models and hyperparameters iteratively.

Solution Architecture:

- **Data Flow:** Collect, preprocess, extract features, train, evaluate, iterate.
- **Feedback Loop:** User feedback for model refinement.
- **User Interface:** Intuitive feedback interface.
- **Integration:** Collaborate with ESPs for seamless implementation.

Key Metrics:

- **Accuracy:** Correct spam/non-spam classification.
- **Precision:** Accurate spam identification.
- **Recall:** Capturing all spam messages.
- **F1-Score:** Balance of precision and recall.
- **False Positives/Negatives:** Misclassified messages.

Implementation and Deployment:

- **Model Deployment:** Launch in production.
- **User Training:** Educate end users.
- **Continuous Monitoring:** Adapt to evolving spam patterns.

Feedback and Iteration:

- **Model Refinement:** Based on user feedback.
- **UI Enhancements:** Improve user experience.
- **Integration Improvements:** Ensure smooth ESP integration.

This design thinking document outlines the approach to create an AI-powered spam classifier, emphasizing high accuracy while minimizing false positives and false negatives.