Table of Contents

[Introduction of Design Document 2](#_Toc160032757)

[Exploratory Data Analysis 2](#_Toc160032758)

[Features Selection and Engineering 2](#_Toc160032759)

[Model Selection 3](#_Toc160032760)

[Image Generation for Training Data 3](#_Toc160032761)

[Training Process 4](#_Toc160032762)

[Model Evaluation and Validation 4](#_Toc160032763)

[Machine Learning Model Training 5](#_Toc160032764)

[Sequence Diagrams 6](#_Toc160032765)

[Data Flow and System Integration 6](#_Toc160032766)

[Streamlit Web Application 7](#_Toc160032767)

# Introduction of Design Document

For a project, a design document outlines the plan for development, highlighting the system's architecture, operation, and design in broad strokes.

Providing insight into the system's layout and functionality, the design document harnesses structure and brevity to dissect intricate concepts into digestible portions. By employing diagrams and visual cues, the document offers a clear illustration of the system's inner workings and data flow in order to facilitate comprehension. Furthermore, the document doubles as a roadmap that illuminates the project's objectives and path, effectively keeping all stakeholders in the surround.

# Exploratory Data Analysis

EDA is an initial step in understanding dataset for the Antibiotic Sensitivity Testing project using Machine Learning and image data. It involves exploring and visualizing data to gain insights that inform subsequent steps.

* Data Understanding:

EDA offers a deep understanding of image dataset's distribution, range, and anomalies. This understanding helps us in choices later on.

* Anomaly Detection:

EDA is helpful in identifying the outliers, ensuring that the data is clean and reliable before model training.

* Feature Relevance:

By uncovering significant image aspects, EDA informs effective model design and feature selection.

Here are the types of EDA for image data are mentioned below.

**Preprocessing:**

Preprocessing is a critical stage in preparing image data for analysis and modeling. It involves several essential tasks that enhance the quality and utility of our dataset.

Preprocessing matters in many elements or factors are as given below:

* Data Quality:

Preprocessing identifies and addresses data issues like missing values, outliers, and noise. This ensures that subsequent analyses and models are built on reliable data.

* Normalization and Scaling:

Techniques like normalization bring all pixel intensities to a consistent range. This removes biases caused by varying intensity levels and aids model training.

* Features Extraction:

Preprocessing can involve extracting key features from images, such as textures or edges, which help differentiate between image classes.

* Dimensionality Reduction:

Techniques like PCA simplify high-dimensional data, making it easier to understand and work with while retaining important information.

# Features Selection and Engineering

In this section, we outline the process of selecting and engineering features from the dataset to enhance the performance of our machine learning model for antibiotic sensitivity testing. The goal is to extract relevant information from the raw data that can significantly help us with the model's accuracy and generalization.

**Feature Selection:**

Here are some points we must keep in mind regarding features selection:

* Describe the criteria and rationale for selecting specific features from the dataset.
* Mention any domain knowledge or research that guided the selection process.
* Explain any feature exclusion decisions and their justifications.

**Feature Engineering:**

Detail the techniques applied to create new features from the existing ones, such as scaling, normalization, transformation, and aggregation.

Highlight any domain-specific feature engineering methods that enhance the model's ability to capture antibiotic sensitivity patterns.

# Model Selection

In this section, we justify the choice of machine learning models for predicting antibiotic sensitivity based on the pre-processed and engineered features. The model selection process is guided by research, evidence, and the nature of the problem.

Here are some points which are necessary while selecting a model:

* Provide an overview of the types of machine learning models considered for the task.
* Present research or literature supporting the feasibility of using machine learning for antibiotic sensitivity prediction.
* Explain the rationale behind the final model selection, highlighting its suitability for the dataset and the problem statement.
* Discuss any trade-offs or considerations made during the model selection process.
* Mention any hyperparameter tuning conducted to optimize the chosen model's performance.

After feature selection and engineering process as well as the model selection rationale, this design document gives a transparent and well-informed approach in handling the antibiotic sensitivity testing problem using machine learning.

# Image Generation for Training Data

The image generation module plays a pivotal role in the system, creating synthetic yet realistic antibiogram plate images for model training. This module employs advanced algorithms to simulate various scenarios of antibiotic resistance and sensitivity. Parameters such as dish radius, disc radius, antibiotic names, and concentrations are programmatically varied to generate a diverse set of images. This diversity is crucial for training the model to recognize a wide range of inhibition zones, thereby enhancing its accuracy and robustness. The generated images are annotated with precise measurements of the zones, forming a comprehensive dataset for training the U-Net model.

# Training Process

This part explains how we taught our model to understand the data and make accurate predictions about antibiotic sensitivity. Training is like teaching a computer to think and make smart decisions.

**1.Data Splitting:** We divided our dataset into parts: one to teach the model and another is to test how well it learned.

**2.Model Learning:** We fed the training data into the model and let it learn the patterns in the data. The model adjusted its settings to get better at predicting antibiotic sensitivity.

**3.Validation:** We checked the model's progress using the testing data. If it did well on this data, it means it's understanding and not just memorizing.

**4.Iteration:** We repeated the training and testing steps, adjusting the model's settings to improve its accuracy. This process helped us fine-tune the model.

By following these steps, our machine learning model gradually got smarter and better in understanding antibiotic sensitivity, making it a useful tool for healthcare applications.

# Model Evaluation and Validation

In this section, we will discuss how we made sure our machine learning model was doing a good job at predicting antibiotic sensitivity.

**Metrics for Evaluation:** We used specific measurements to understand how well the model was doing. Metrics like accuracy, precision, recall, and F1-score helped us judge its performance.

**Cross-Validation**: To make sure our model works well on different data, we used cross-validation. This is like giving the model different tests to see if it consistently gets good grades.

**Results:** We share the outcomes of our evaluations, showing how accurate our model is and where it might still need improvement.

**Fine-tuning:** If the model's performance wasn't great, we went back to the training step and adjusted things. This helps the model get better and better over time.

By evaluating and validating our model, we gained confidence in its ability to make accurate predictions about antibiotic sensitivity and making it a reliable tool for medical decision-making.

# Machine Learning Model Training

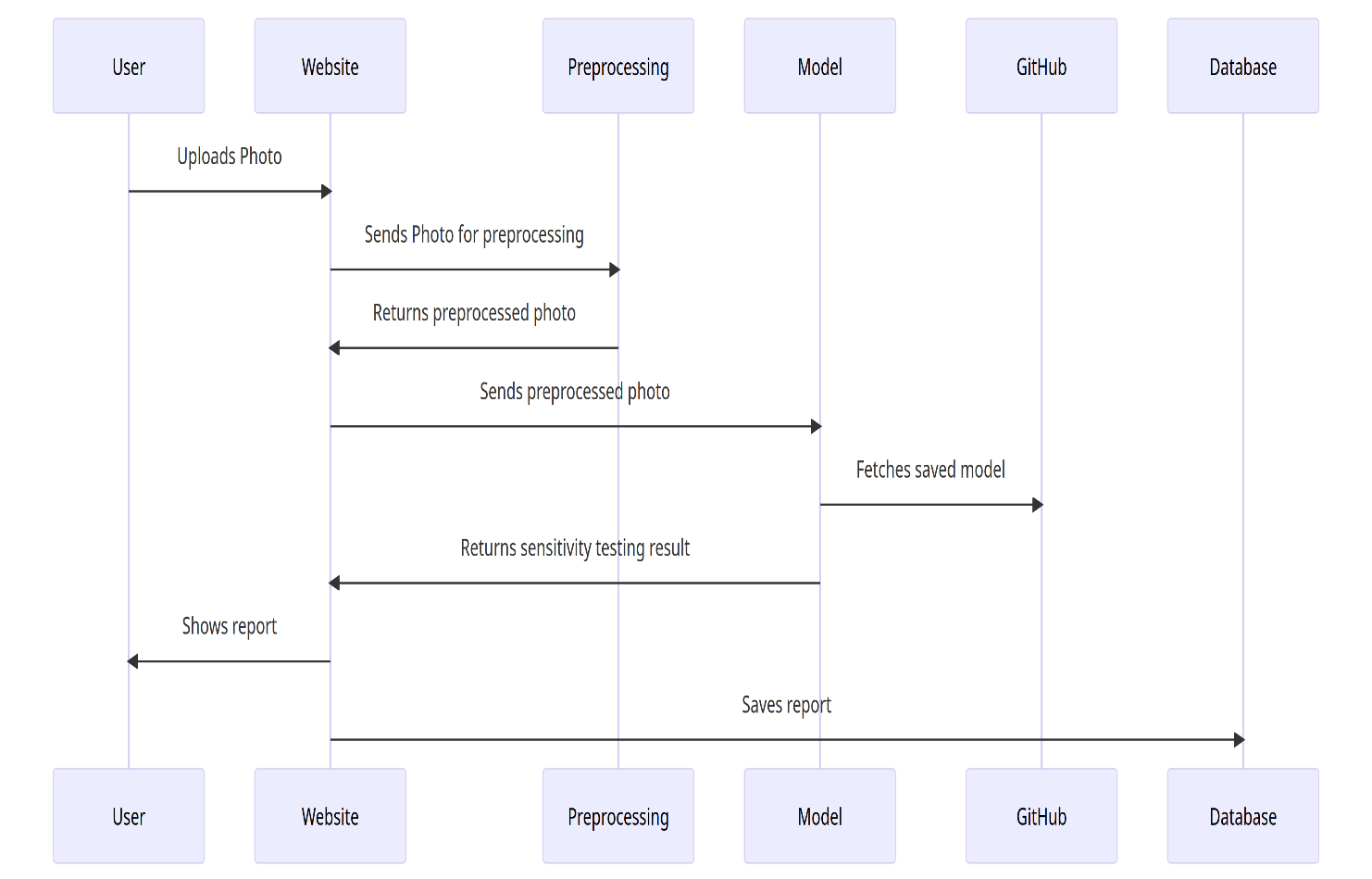
The machine learning model, employing a U-Net architecture, is meticulously trained for image segmentation tasks. The U-Net model, renowned for its effectiveness in medical image segmentation, is ideal for the precise identification of zones of inhibition in antibiogram images. The training process involves several steps, including data preprocessing, model training, and validation. Preprocessed images, generated through the image generation module, form the training dataset. These images are annotated with the zones of inhibition, providing the ground truth for model training. The model learns to identify and segment these zones accurately, a crucial step in automating the sensitivity testing process. Post-training, the model undergoes rigorous validation to ensure its accuracy and reliability.

A diagram of a computer system

Description automatically generated

# Sequence Diagrams

A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects. Sequence Diagrams are time focus, and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. They can be used by business and technical users but are more commonly used for in technical descriptions of a system.



# Data Flow and System Integration

The system exhibits a seamless integration of its core components, ensuring a smooth and efficient workflow. The data flow begins with the user uploading an antibiogram image via the Streamlit application. This image is then preprocessed to conform to the input requirements of the machine learning model. The preprocessed image is fed into the model, which performs segmentation to identify and measure the zones of inhibition. These measurements are then superimposed on the original image, providing a visual representation of the analysis. The processed image, along with the analysis results, is displayed back in the Streamlit application, offering users an immediate and clear understanding of the antibiotic sensitivity.

# Streamlit Web Application

The Streamlit web application is a core component of this system, serving as the user interface. It is designed to be intuitive and accessible, allowing users with varied technical backgrounds to interact with the system effortlessly. The application is developed using Python and the Streamlit framework, known for its ease of use in creating web applications. Key features of the application include image upload functionality, real-time image processing, and interactive result display. Users can upload antibiogram plate images, which are then processed by the backend machine learning model. The results, including the detected zones of inhibition, are displayed both on the original and processed images, providing a clear and understandable output.  
  
  
  
**The END**