

# Design Document: EMT Patient Simulation AI

## Project Overview

The **EMT Patient Simulation AI** is an interactive tool designed to help EMT trainees practice patient assessment. The AI simulates a patient with specific symptoms, allowing the user to ask questions, analyze responses, and make a diagnosis. Key features include:

1. AI-generated responses based on user questions.
  2. Performance tracking with point-based grading.
  3. Final evaluation of the user's diagnosis.
- 

## System Architecture

The project is composed of four main scripts, each with a specific role:

### 1. File Reader (file\_reader.py)

This script stores all data related to patient conditions and symptoms, including:

- **Patient Database:** A structured collection of conditions, symptoms, and vital signs.
- **Keyword Synonyms:** A dictionary of alternative phrases for symptom-related keywords, used for improving user interaction.

Example functionality:

- Allows the simulation to access data like "Heart Attack" symptoms or associated vital signs.
- 

### 2. Intent Classification (bert\_pipeline.py)

This script is responsible for identifying the intent behind user questions using a fine-tuned **DistilBERT** model. It includes:

- **Intent Labels:** A predefined list of possible user intents (e.g., "Ask about blood pressure").
- **Intent Classification:** Processes user queries and maps them to the most relevant intent based on the model's predictions.

Key points:

- Fine-tuning involves training DistilBERT on a custom dataset of example questions and intents.
  - After fine-tuning, the model is saved and used in the simulation.
-

### 3. Grading System (grading.py)

This script manages user performance by:

- **Tracking Actions:** Maintains a record of whether specific questions (e.g., "PPE precautions") have been asked.
  - **Awarding Points:** Assigns points for correct actions and deducts points for critical errors.
  - **Processing Questions:** Handles user input by determining the intent, fetching relevant data, and updating the score.
- 

### 4. Simulation Script (patient\_simulation.py)

The main simulation script ties all components together:

1. Randomly selects a patient condition from the database.
  2. Displays the patient's symptoms to the user.
  3. Allows the user to ask up to 15 questions.
  4. Provides responses based on intent classification and patient data.
  5. Scores the user's performance and evaluates their final diagnosis.
- 

#### Interaction Flow

1. **Simulation Start:**
    - The AI selects a condition and displays the patient's symptoms.
  2. **User Interaction:**
    - The user asks questions to gather information about the patient's condition.
    - The AI classifies each question, retrieves relevant data, and provides responses.
  3. **Final Diagnosis:**
    - The user guesses the patient's condition.
    - The AI evaluates the guess and displays the total score.
- 

#### Performance Metrics

- **Scoring:** The user's performance is evaluated based on:
  - Actions like checking for PPE precautions, scene safety, and asking about vital signs.

- Correctly identifying the system affected by the condition.
    - Providing the correct final diagnosis.
  - **Critical Errors:** Certain actions (e.g., failing to check scene safety) result in an immediate fail.
- 

## Future Enhancements

### 1. Improved Intent Recognition:

- Enhance the model's accuracy by expanding the dataset or using a more robust pre trained model.

### 2. Dynamic Symptoms:

- Allow the AI to simulate variable patient conditions, such as worsening symptoms over time.

### 3. Detailed Feedback:

- Provide the user with a detailed report on their performance, highlighting strengths and areas for improvement.

### 4. UI:

- a more immersive UI with a window and question count instead of everything being ran in the terminal.