# Project Plan

1. **Timeline**

A detailed timeline will be created using a **Gantt chart** to track the project's progress. The timeline will outline key phases, tasks, and deadlines for each delivery.

We will use **ClickUp** to manage the timeline, assigning tasks to team members and setting dependencies between different project stages

<https://app.clickup.com/9012675172/v/l/8ck50k4-252>

1. **Milestones & Deliverables**

| **Milestone** | **Deliverables** | **Due Date** |
| --- | --- | --- |
| Project Initiation | Project Proposal, Initial Research | Week 1 |
| Data Collection | Gather and preprocess medical documents | Week 3 |
| Model Selection | Choose suitable NLP models and techniques | Week 5 |
| Model Development | Train and fine-tune the NLP model | Week 7 |
| Evaluation & Testing | Model evaluation, testing, and validation | Week 9 |
| Deployment | Deploying the system on a cloud platform | Week 11 |
| Final Report & Review | Documentation and final project review | Week 12 |

1. **Resource Allocation**

Each team member will have specific responsibilities to ensure efficient workflow.

| **Team Member** | **Responsibility** |
| --- | --- |
| Esraa Said, Asmaa Hassan | Data Collection & Preprocessing |
| Esraa said, Mohamed Shehata | Model Development (NLP, Training) |
| Asmaa Hassan, Mohamed Elkholy | Model Evaluation & Optimization |
| Marwa Mansour, Mohamed Elkholy | System Deployment & Integration |
| Mohamed Shehata, Marwa Mansour | Documentation & Final Report |

Each member will collaborate closely to ensure smooth execution and project success.

# Literature Review

**Deep Learning for Medical Information Extraction**

A breakthrough study by Cireşan et al. (2013) introduced deep learning for medical information extraction. The study applied Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to extract medical concepts from clinical notes, focusing on identifying metastatic breast cancer from imaging and text data. Their work demonstrated how deep learning can be effectively used in clinical text processing and automated diagnosis.

Reference: Cireşan, D. C., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2013). Deep Learning for Identifying Metastatic Breast Cancer. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013. DOI: 10.1109/ICCV.2013.69 (https://doi.org/10.1109/ICCV.2013.69)

Another important contribution to medical information extraction using LSTM networks was made by Lee et al. (2020). Their study focused on the use of deep learning models like LSTMs to extract key medical entities such as diseases, symptoms, and medications from clinical texts. This research has significant implications for improving decision support systems in healthcare by extracting structured information from unstructured data.

Reference: Lee, J., Yoon, W., Kim, S., Kim, D., & So, C. H. (2020). Clinical Text Analysis and Applications in Real-World Healthcare. Journal of Healthcare Informatics Research. DOI: 10.1007/s41666-020-00081-5 (https://doi.org/10.1007/s41666-020-00081-5)

# Requirements Gathering

## 1. Stakeholder Analysis

1. Healthcare Professionals (Doctors, Nurses, and Medical Staff): They will interact with the system for extracting medical information from clinical notes.

* Needs: Quick, accurate extraction of health-related data such as diagnoses, treatments, and medications. User-friendly interface to make interaction efficient

1. Healthcare Institutions (Hospitals, Clinics): Administrators, system maintainers, and data managers.

* Needs: A reliable system for extracting and managing patient data from large volumes of unstructured clinical notes. Security of patient data is a priority.

1. Patients: They may not directly interact with the system but will benefit from improved healthcare through accurate data extraction.

* Needs: Ensuring that their health information is handled securely and that errors in diagnosis or treatment are minimized through better data extraction.

1. Software Developers & Data Scientists: The team responsible for building and maintaining the system.

* Needs: Clear guidelines for system functionalities, integration with existing tools, and flexibility to train the system on new medical data.

## 2. User Stories & Use Cases

**• User Story 1**

As a doctor, I want the system to automatically extract diagnoses and medications from clinical notes so that I can make faster, more accurate treatment decisions.

• Acceptance Criteria:

- The system can accurately identify diagnoses and medications from the provided text.

- Extracted data is displayed in an easy-to-read format.

- The doctor can edit the extracted data if needed.

**• Use Case 1: Extract Diagnosis Information from Clinical Notes**

Actor: Doctor

Preconditions: Clinical notes have been input into the system.

Flow of Events:

1. The doctor uploads a new clinical note into the system.

2. The system processes the text and identifies relevant health information.

3. The system highlights the extracted diagnoses and medications.

4. The doctor reviews and confirms the extracted data.

5. The system stores the extracted data in the database.

## 3. Functional Requirements

1. Text Extraction:

* The system should automatically extract health-related entities (diagnoses, medications, allergies, etc.) from unstructured clinical notes.

1. User Interface:

* The system should have an intuitive, user-friendly interface to allow healthcare professionals to review and edit extracted data.

1. Data Integration:

* The system should integrate with hospital Electronic Health Records (EHR) to retrieve patient data and update medical records.

1. Reporting:

* The system should generate reports summarizing extracted medical information for further analysis or decision-making.

1. Error Handling:

* The system should flag any inconsistencies or uncertainties in extracted data and allow users to verify or edit the information

## 4. Non-functional Requirements

## • Performance

The system should process clinical notes and return extracted data within 5 seconds for files containing up to 5000 words.

**• Security**

The system must ensure data encryption at rest and during transmission to protect sensitive medical data.

Access to data must be restricted based on user roles, ensuring that only authorized personnel can access certain types of data.

**• Usability**

The user interface should be designed to minimize cognitive load and be easy to navigate for medical professionals, even those with limited technical expertise.

Support for multiple languages and medical terminologies is required.

**• Reliability**

The system should maintain 99.9% uptime to ensure it is always available for healthcare professionals.

The system should be fault-tolerant and handle errors gracefully without losing data or crashing.

# System Analysis & Design

## Problem Statement & Objectives

Problem Statement: The goal of the project is to create an automated medical information extraction system using Natural Language Processing (NLP) to process and extract meaningful data from unstructured clinical notes. Healthcare professionals spend a significant amount of time manually reviewing clinical notes, which can lead to errors, delays, and inefficiencies in patient care.

**Project Objectives:**

1.Develop an NLP model to accurately extract medical entities such as diagnoses, medications, treatments, allergies, and procedures from clinical texts.

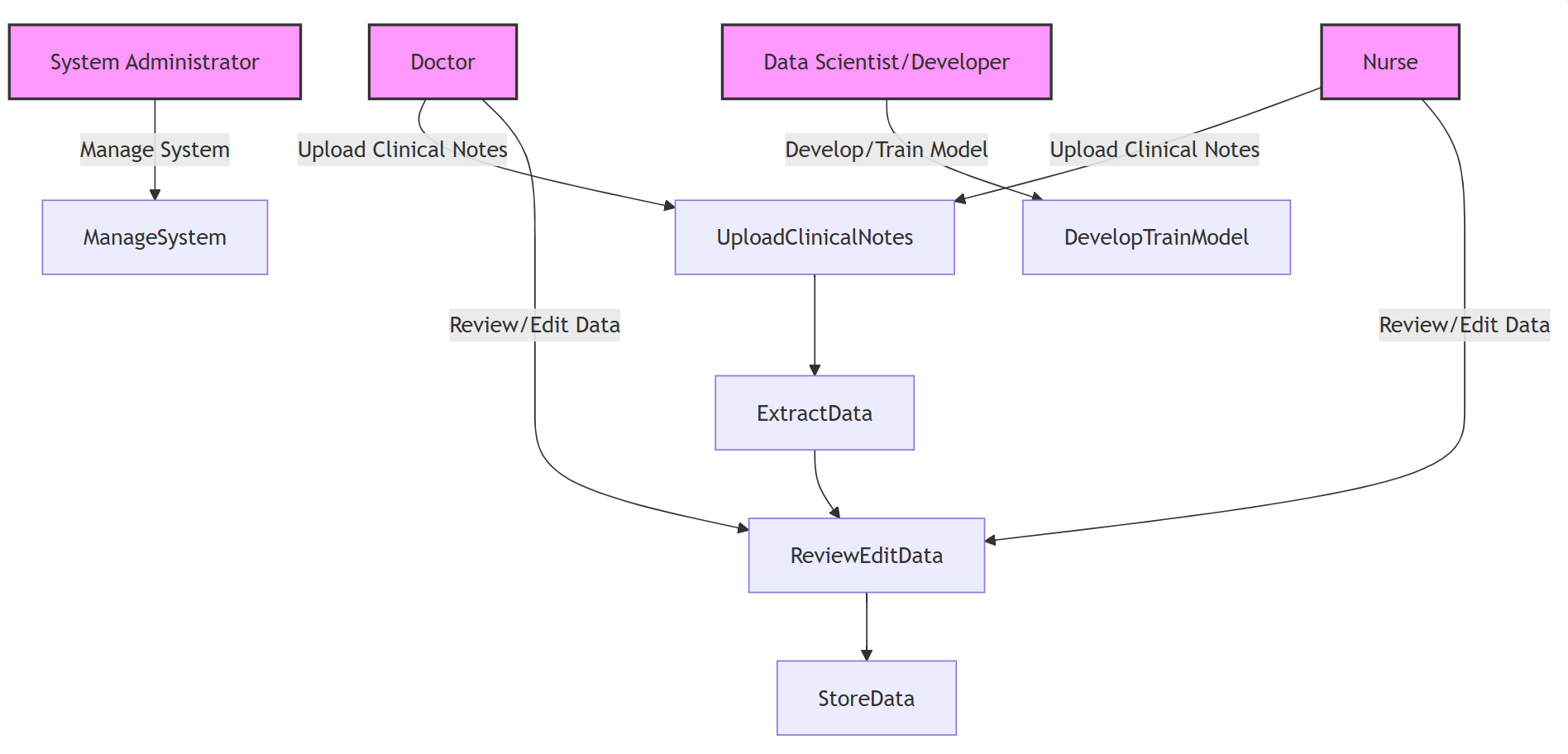
2.Create a user-friendly interface for healthcare professionals to interact with the system, review extracted data and ensure accuracy.

3.Integrate the system with hospital Electronic Health Records (EHR) for seamless data updates.

4.Improve efficiency and accuracy in clinical decision making by reducing manual data entry and potential errors.

## Use Case Diagram & Descriptions

#### 2.1 Use Case Diagram



**Actors:**

**1.Doctor** – Interacts with the system to upload clinical notes and review extracted medical data.

**2.Nurse** – Assists in uploading clinical notes and verifies extracted data.

**3.System Administrator** – Manages system settings, access permissions, and user roles.

**4.Patient** – The source of clinical data, benefiting indirectly from improved care.

**5.Data Scientist/Developer** – Works on the model development, training, and system maintenance.

#### 2.1 Use Case Descriptions

Use Case 1: Upload Clinical Note

- Actor: Doctor, Nurse

- Description: User uploads clinical notes into the system.

- Preconditions: User must be logged in.

- Postconditions: Clinical notes are uploaded to the system.

Use Case 2: Process Clinical Note

- Actor: System

- Description: System processes the uploaded clinical notes and extracts relevant information.

• **Review and Edit Extracted Data**: Doctors or nurses review and edit the extracted data.

**• Store Extracted Data:** The system stores the data in the database and updates the patient's record.

## 3. Functional & Non-Functional Requirements

#### 3.1 Functional Requirements

1. Automatically extract medical entities (e.g., diagnoses, medications) from clinical notes.
2. Allow users to review and edit the extracted data.
3. Integrate with Electronic Health Records (EHR) systems for data synchronization.
4. Generate reports based on the extracted medical data.

#### 3.2 Non-Functional Requirements

• Performance: System should process data within 5 seconds for files up to 5000 words.

• Security: All data should be encrypted at rest and in transit.

• Reliability: The system should maintain 99.9% uptime.

## 4. Software Architecture

#### 4.1 High-Level Design

Overview: A high-level description of the system components and how they interact.

Architecture Style: Microservices / Modular Design (Based on your architecture decision).

## 5. Database Design & Data Modeling

#### 5.1 Entity-Relationship Diagram (ERD)

A screenshot of a computer

AI-generated content may be incorrect.

#### 5.2 Logical & Physical Schema

• Tables: List out key tables such as `Patient`, `Clinical Note`, and `MedicalEntity`.

• Attributes: Describe attributes of each table (e.g., `PatientID`, `NoteID`, `Text`, etc.).

• Keys: Primary and foreign key relationships.

## 6. Data Flow & System Behavior

#### 6.1 Data Flow Diagram (DFD)

**1. DFD (Data Flow Diagram)**

* **Context-Level Diagram:**
  + The user (Doctor/Nurse) uploads clinical notes into the system. The system processes these notes using the NLP engine and stores the extracted data in the database.
* **Detailed-Level Diagram:**
  + Shows the internal processes for text extraction, data storage, and user interaction.

A diagram of a data flow

AI-generated content may be incorrect.

## A diagram of a system AI-generated content may be incorrect.6.2 Sequence Diagram

## 7. System Deployment & Integration

#### 8.1 Technology Stack

• Backend: Python (Flask/FastAPI), NLP libraries (SpaCy, NLTK)

• Frontend: React, Angular

• Database: PostgreSQL, MongoDB

# Risk Assessment

## Identified Risks & Mitigation

|  |  |
| --- | --- |
| **Risk** | **Mitigation Strategy** |
| Data Privacy Issues | Use anonymized datasets |
| Model Accuracy Concerns | Implement multiple evaluation metrics |
| Time Constraints | Set buffer periods in the timeline |
| Technical Challenges | Allocate research time for troubleshooting |

# KPIs (Key Performance Indicators)

|  |  |
| --- | --- |
| KPI | Measurement |
| Model Accuracy | Above 85% F1-score |
| Processing Time | Less than 2 seconds per document |
| User Adoption Rate | At least 80% positive feedback |
| System Uptime | 99% uptime |

# Task Assignment

|  |  |
| --- | --- |
| **Task** | **Assigned Member** |
| Data Collection & Cleaning | Esraa Said, Asmaa Hassan |
| Model Training & Optimization | Esraa said, Mohamed Shehata |
| Backend API Development | Marwa Mansour |
| Frontend & UI/UX | Mohamed Elkholy |
| Testing & Documentation | Asmaa Hassan |