I. Table of Contents

* Introduction
* Technical Design
  + Microservices Architecture
  + Input Microservice
  + Audio/Video File Processing Microservice
  + Audio/Speech Recognition Microservice
  + Question Detection Microservice
  + Text Pre-Processing Microservice
  + Question Comparison Microservice
  + Output Microservice
  + Data Storage Service
  + Database Connectivity Service
  + Model Training Microservice
  + User Interface Service
* System Requirements
  + Operating System
  + Container Management
  + Programming Language
  + Web Framework
  + API for Microservice Communication
  + Testing Library
  + Pretrained Models
  + Data Privacy Compliance
* User Roles and Authentication
* Conclusion

II. Introduction This software requirements specification outlines the requirements for a microservice architecture application for extracting medically relevant information from medical interviews. The application will use automatic speech recognition and artificial intelligence to transcribe and analyze the audio or video data, and store the results in a secure and compliant manner.

III. Technical Design

1. Microservices Architecture: The application will be implemented as a set of microservices, each responsible for a specific task. The microservices will communicate with each other through APIs.
2. Input Microservice: The input microservice will provide a web-based interface for
3. Audio/Video File Processing Microservice: The audio/video file processing microservice will extract the audio or video data from the file and store it in a buffer.
4. Audio/Speech Recognition Microservice: The audio/speech recognition microservice will use an Automatic Speech Recognition (ASR) engine to transcribe the audio data into text. The ASR engine will be trained on medical interviews to improve its accuracy.
5. Question Detection Microservice: The question detection microservice will use the transformer neural network to identify medically relevant questions in the transcribed text.
6. Text Pre-Processing Microservice: This microservice will perform any necessary pre-processing on the input text, including stop words, stemming, lemmatization, and any other relevant tasks.
7. Question Comparison Microservice: The question comparison microservice will compare the extracted questions to a database of pre-defined medical questions. The closest matching question will be selected and its corresponding answer will be recorded.
8. Output Microservice: The output microservice will store the questions and answers locally in a JSON file. The file will include the date, time, and type of medical interview.
9. Data Storage Service: This module will store the questions and answers detected in the medical interview in a secure and compliant manner, in accordance with European Union GDPR regulations. The data will be stored locally and will not be allowed to be shared over the internet.
10. Database Connectivity Service: This module will connect to the database and perform any SQL statements required by the Data Storage Service. It will read a configuration file to determine the database to use, the username and password, and any other required parameters.
11. Model Training Microservice: This module will allow the administrator to create and train new models for the application.
12. User Interface Service: This module will provide a web-based user interface for accessing and managing the results of the medical interview processing. The user will be able to view, search, sort, and filter the results by date, time, type, and other criteria. The user will also be able to export the results to a CSV or JSON file for further analysis.

IV. System Requirements

Operating System: The application will run under Ubuntu Linux.

* Container Management: Each microservice will run in its own Docker container, managed via Kubernetes.
* Programming Language: All code will be written in Python.
* Web Framework: The Django web framework will be used.
* API for Microservice Communication: The microservices will communicate with each other through ReSTful APIs.
* Testing Library: Unit tests and integration tests for each module will be written in Python using the PyUnit testing library.
* Pretrained Models: The application will initially use the BERT-ClinicalQA and BioBERT-mnli-snli-scinli-scitail-mednli-stsb pretrained models.
* Data Privacy Compliance: The application will comply with European Union GDPR regulations for data privacy.

V. User Roles and Authentication

* User: The user will be allowed to upload a video or audio file and receive as final output all relevant medical questions. The user must have a login and password authentication.
* Administrator: The administrator will be allowed to create, modify, or delete users to the application. Additionally, the administrator will be allowed to train and upload new versions of the models.

VI. Conclusion This software requirements specification outlines the requirements for a microservice architecture application for extracting medically relevant information from medical interviews. The application will use automatic speech recognition and artificial intelligence to transcribe and analyze the audio or video data, and store the results in a secure and compliant manner.