1. **Class Diagram**

The class diagram illustrates the structure of the system by showing the system's classes, their attributes, methods, and the relationships between the classes. Here’s a detailed overview of the classes involved in the Speech-to-Text Converter project:

**1. UserInterface**

* **Attributes**:
  + window: Tk
  + record\_button: Button
  + stop\_button: Button
  + transcription\_textbox: Text
* **Methods**:
  + init\_ui()
  + start\_recording()
  + stop\_recording()
  + display\_transcription(transcription: str)

**2. AudioRecorder**

* **Attributes**:
  + is\_recording: bool
  + audio\_file\_path: str
* **Methods**:
  + start()
  + stop()
  + save\_audio()

**3. TranscriptionService**

* **Attributes**:
  + audio\_file\_path: str
  + transcription: str
* **Methods**:
  + transcribe\_audio() -> str

**4. SpeechRecognitionAPI**

* **Attributes**:
  + api\_key: str
* **Methods**:
  + send\_audio\_for\_transcription(audio\_file\_path: str) -> str
  + handle\_api\_response(response: dict) -> str

**5. LanguageModel**

* **Attributes**:
  + supported\_languages: List[str]
  + current\_language: str
* **Methods**:
  + set\_language(language: str)
  + get\_supported\_languages() -> List[str]

**6. Settings**

* **Attributes**:
  + language: str
  + save\_transcriptions: bool
* **Methods**:
  + load\_settings()
  + save\_settings()

**7. DataManager**

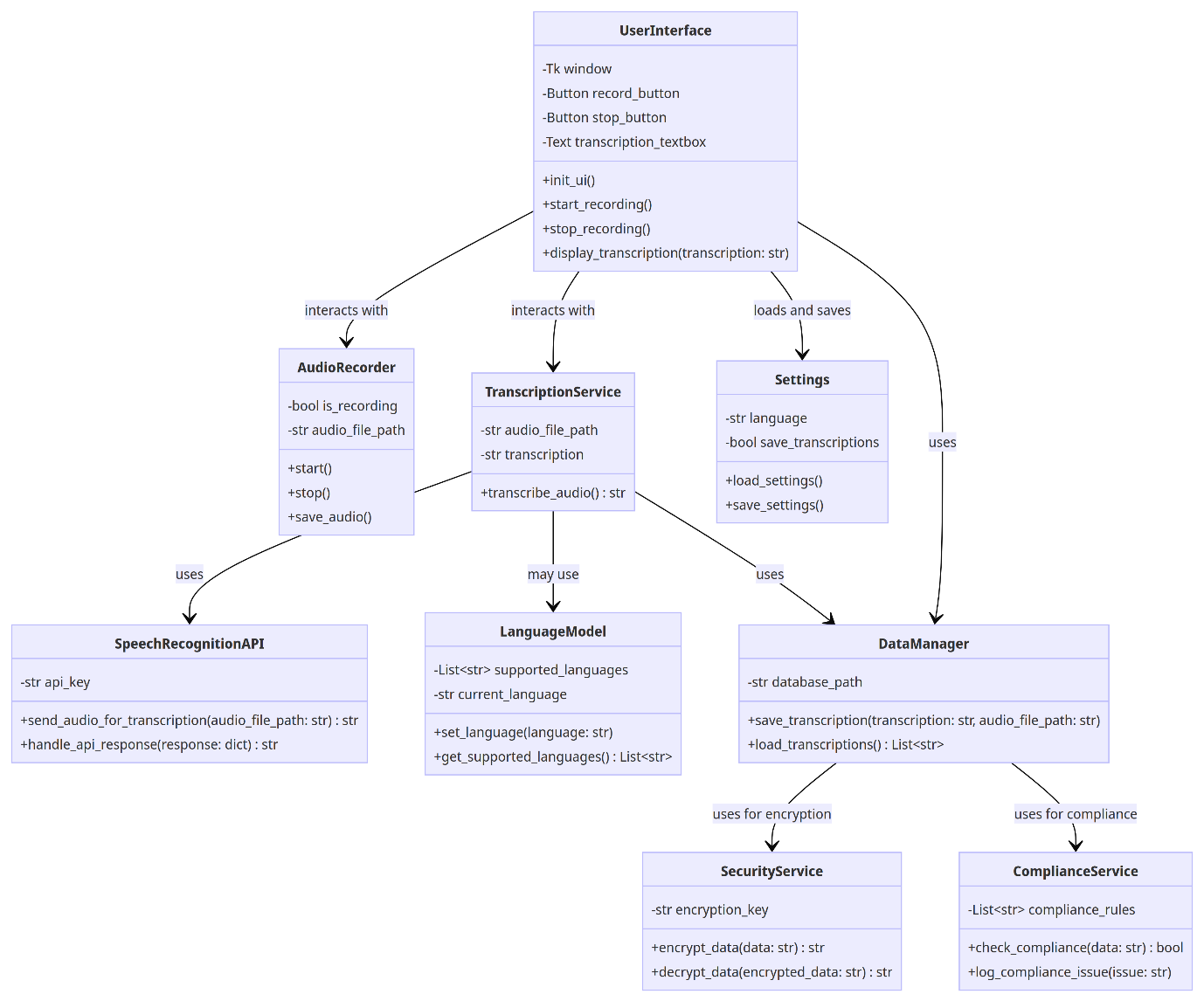
* **Attributes**:
  + database\_path: str
* **Methods**:
  + save\_transcription(transcription: str, audio\_file\_path: str)
  + load\_transcriptions() -> List[str]

**8. SecurityService**

* **Attributes**:
  + encryption\_key: str
* **Methods**:
  + encrypt\_data(data: str) -> str
  + decrypt\_data(encrypted\_data: str) -> str

**9. ComplianceService**

* **Attributes**:
  + compliance\_rules: List[str]
* **Methods**:
  + check\_compliance(data: str) -> bool
  + log\_compliance\_issue(issue: str)
  1. **Relationships**
* **UserInterface** interacts with **AudioRecorder** for starting and stopping the recording.
* **UserInterface** interacts with **TranscriptionService** to get the transcription and display it.
* **TranscriptionService** uses **SpeechRecognitionAPI** to send audio for transcription.
* **TranscriptionService** may use **LanguageModel** to set the appropriate language model.
* **Settings** class manages user preferences such as language and whether to save transcriptions.
* **DataManager** handles saving and loading transcriptions to/from the database.
* **SecurityService** is used to encrypt and decrypt sensitive data.
* **ComplianceService** ensures that the data handling is compliant with relevant regulations.
  1. **Class Diagram**

**