**System Architecture “mentis” final project**

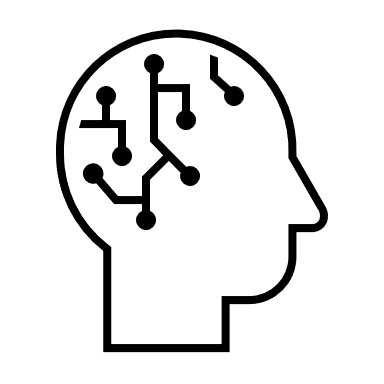
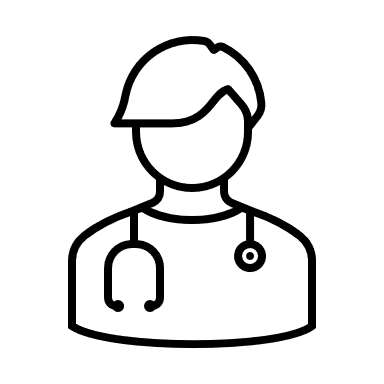
**Ohad Buskile- 207422080**

**Illana Gofman- 322766932**

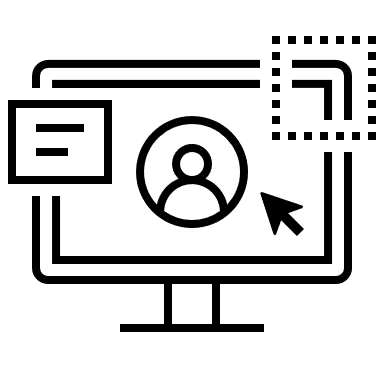
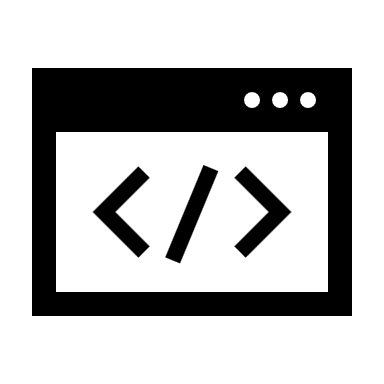
**High-Level Architecture Diagrams**

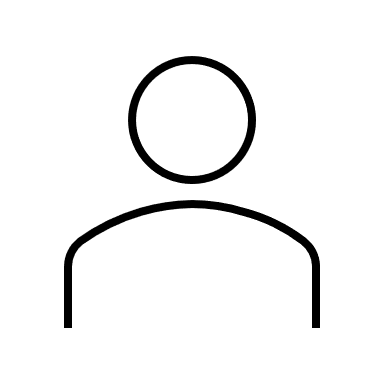
**Overview:** The system is designed as a client-server model with the following components:

1. **Client Layer:**
   * Mobile app interface for end-users and therapists.
   * Web interface for therapist dashboard.
2. **Server Layer:**
   * Authentication and user management module.
   * Journaling and sentiment analysis engine.
   * Notification service.
3. **Data Layer:**
   * Encrypted database for user data storage.
   * Analytics database for sentiment trends.

****

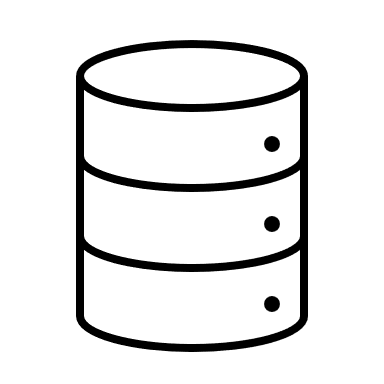
ML/AI Models

****

****

Front-End

GUI

****

Back-End

**UML Diagrams**

**Class Diagram**

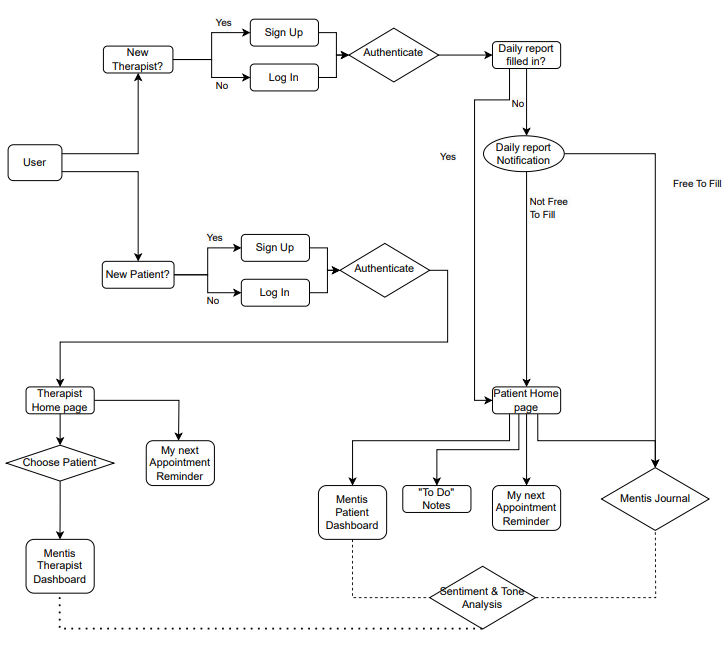
1. **User**: Attributes for ID, name, email, role (user/therapist).
2. **JournalEntry**: Attributes for ID, userID, timestamp, content, sentiment.
3. **TherapistDashboard**: Attributes for user summaries and visualizations.

**Sequence Diagram**

1. User logs in → Journaling interface loads → Entry saved → Sentiment analysis performed → Updates dashboard.
2. Therapist logs in → Dashboard retrieves user summaries → Visualizations displayed.

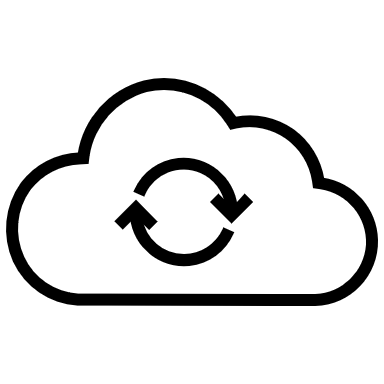
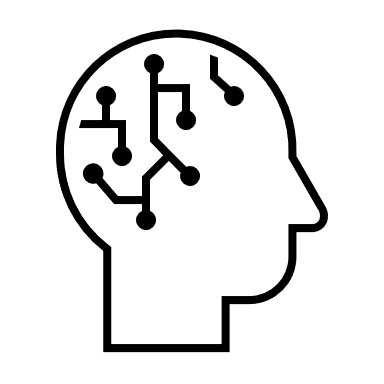
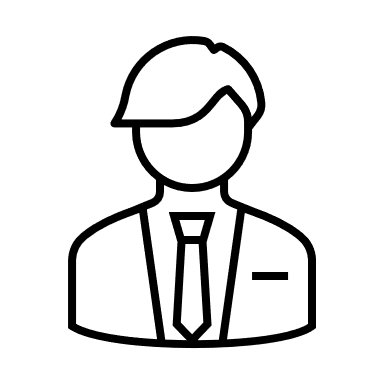
**Activity Diagram**

1. User writes journal entry.
2. Sentiment analysis and summarization executed.
3. Therapists notified of significant patterns.

****

**USE-CASE DIAGRAM**

User-Client server

****

**Authentication**

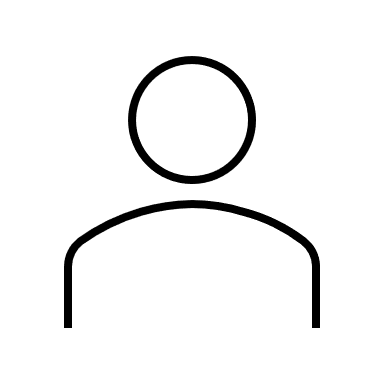
**Journaling**

**Dashboard**

**Notifications**

**S&T Analysis**

**Summarization**

****

System (AI-Powered Application)

**Data Main Tables**

1. **Users**
   * id (PK), name, email, passwordHash, role
2. **JournalEntries**
   * id (PK), userId (FK), timestamp, content, sentimentScore
3. **Notifications**
   * id (PK), userId (FK), type, message, timestamp
4. **Analysis**
   * id (PK), userId (FK), JournalEntries (FK), Sentiment score, LLM sum, timestamp

**Database Schema Design**

|  |
| --- |
| **User** |
| **ID** |
| **name** |
| **email** |
| **password** |
| **role** |

|  |
| --- |
| **Journal** |
| **ID** |
| **user Id** |
| **timestamp** |
| **content** |

|  |
| --- |
| **Notifications** |
| **ID** |
| **user Id** |
| **type** |
| **message** |
| **timestamp** |

|  |
| --- |
| **Analysis** |
| **ID** |
| **user Id** |
| **Journal Id** |
| **Sentiment score** |
| **LLM sum** |
| **timestamp** |

**API Specification**

* **POST /auth/login:** Authenticate users.
* **POST /journal/add:** Add new journal entry.
* **GET /dashboard/summary:** Retrieve user insights for therapists.

**Interface Design and Prototypes**

1. **Mobile App:** Simplistic journaling interface with guided prompts.
2. **Web Dashboard:** Chart visualizations for emotional trends and summaries.

**Design Patterns**

* **MVC Pattern:** For separating concerns between interface, business logic, and data.
* **Observer Pattern:** For sending notifications.

**Pseudocode**

**Sentiment Analysis**

function analyzeSentiment(entry):

sentiment = NLPModel.predict(entry.content)

return sentiment

**Modular Decomposition**

**Identifying Modules, Components, Services**

1. **Authentication Module:** User login and access control.
2. **Journaling Service:** Text input and storage.
3. **Sentiment Analysis Service:** NLP-based sentiment tracking.
4. **Notification Service:** Event-driven alerts.

**Module Interfaces and Contracts**

* Journaling service exposes APIs to create, update, and fetch entries.
* Sentiment service provides sentiment analysis and trend data.

**Cohesion and Coupling**

* High cohesion within modules (e.g., sentiment analysis encapsulates all NLP tasks).
* Loose coupling between services (e.g., RESTful APIs enable independent updates).

**Conway's Law**

The system reflects organizational roles:

* User experience prioritization aligns with mobile and dashboard interfaces.
* NLP tasks mirror AI team’s capabilities.
* Modular design facilitates parallel development by separate teams.

This document outlines the foundational design for the "Mentis" AI-powered therapy support application, ensuring scalability, security, and user-centric functionality.