PROJECT REPORT
CSCI305 - DATABASE SYSTEMS

20/4/2025

EMOTICA

MENTAL HEALTH & MOOD JOURNAL SYSTEM

Presented To:
Dr Ahmed Qandil

By:

Shahd El-Kazzaz	221000923
Samer El-Hossany	221001697
Omar Tawfik	221000307
Mostafa El-Sheikh	221000362
Nada Zaki	221000829
Mona El-Sayed	221000415

GITHUB REPO

Nile University

TABLE OF CONTENTS

1. Introduction
2. System Requirements and Objectives
3. Database Design
4. Implementation
5. Key Features in Action
6. Testing and Evaluation
7. Challenges and Solutions
8. Conclusion
9. Appendices
10. References

INTRODUCTION

Emotica is a privacy-respecting personal mental health and mood journal system developed to support individuals in tracking their emotional wellbeing. The application allows users to log daily mood entries, receive tailored advice and music recommendations, connect with therapists and support groups, and trigger alerts when signs of emotional distress are detected. The motivation behind Emotica stems from the growing awareness of mental health issues and the need for digital tools to help individuals and professionals monitor and respond to emotional health trends.

SYSTEM REQUIREMENTS AND OBJECTIVES

Functional Requirements:

- Allow users to log mood entries and receive insights.
- Schedule and manage therapist appointments.
- Enable users to join and create support groups.
- Alert therapists or emergency contacts if mood scores remain low for consecutive days.
- Provide therapists with dashboards to track patient progress.

Non-functional Requirements:

- Ensure user data privacy and security.
- Maintain high availability and integrity of records.
- Offer a responsive and user-friendly GUI (using PyQt).
- Scalable relational database schema (PostgreSQL).

DATABASE DESIGN

Entity-Relationship Diagram (ERD)

The ERD includes primary entities such as Person, User,

Therapist, MoodEntry, Appointment, Alert, SupportGroup, MCQQuestion, MCQResponse, Notes, Motivator, Advice, Music, and relationship tables such as GroupMembership, AdviceGiven, and MusicGiven. Relationships show many-to-many and one-to-many connections, with attributes like assignedTID, createdGroup, joinedGroup, and loggedBy.

Entity Descriptions

- Person: Basic personal information, including name, email, DOB, gender, and emergency contact.
- User: A person who uses the app to track moods.
- Therapist: A person who manages therapy appointments and provides feedback.
- MoodEntry: Daily mood logs, including moodScore, moodColor, and MCQ form data.
- Appointment: Stores appointment details like status, type, and timestamps.
- Alert: Captures emergency triggers with alertType and messageSent.
- SupportGroup: Group chats with members, messages, and admin assignment.
- Motivator: Stores streak data and notifications for user motivation.
- Advice / Music: Recommendations linked to user mood and behavior.

Schema Design

Based on the schema PDF and SQL code, the database is normalized to 3NF:

- Tables have primary keys.
- Foreign keys ensure referential integrity.
- No redundancy due to well-defined entity separation.
- Composite entities like GroupMembership handle many-to-many relationships.

IMPLEMENTATION

Technologies Used:

- Frontend: Python with PyQt for GUI.
- Database: PostgreSQL.
- Libraries: psycopg2 for DB connectivity, matplotlib for charting, SQLAlchemy for ORM.

SQL Snippets: Example (simplified):

```
CREATE TABLE Person (
  personID SERIAL PRIMARY KEY,
  firstName VARCHAR(50),
  lastName VARCHAR(50),
  email VARCHAR(100) UNIQUE,
  password TEXT,
  dateOfBirth DATE,
  gender VARCHAR(10),
  phoneNum VARCHAR(15),
  contactID INT
);
```

```
READE TABLE MoodEntry (
 entryID SERIAL PRIMARY KEY,
 moodDate DATE,
 moodScore INT,
 moodColor VARCHAR(20),
 adviceID INT,
 musicID INT,
 userID INT,
  FOREIGN KEY (userID)
REFERENCES User(userID)
);
```

GUI Integration:

- User logs mood from PyQt interface.
- Data is inserted into PostgreSQL using psycopg2.
- Therapist dashboard fetches patient summaries.
- Mood trend visualizations are plotted using matplotlib.

KEY FEATURES IN ACTION

- **Mood Logging:** Users input mood details, MCQ responses, and optional notes.
- **Therapist Appointment Booking:** Users can book sessions; therapists accept/reject via dashboard.
- **Safe-word Trigger:** Words like "help" or "I'm scared" trigger alerts via the Alert table.
- **Support Groups:** Users join groups and chat via GroupMembership and Message.
- **Low Mood Alert:** If moodScore < threshold for 3 days, an auto alert is generated.
- **Dashboard:** Therapists view charts and summaries of patients' mood trends and appointment history.

TESTING AND EVALUATION

- **Unit Tests:** Insert, update, and query for MoodEntry, Appointment, Alert.
- **Validation:** Ensured foreign key constraints prevent orphan records.
- **Security Tests:** Passwords hashed, input sanitized.
- **Performance:** Indexes added on userID, entryID, and appointmentID.

CHALLENGES AND SOLUTIONS

- **Complex Relationships:** Resolved with intermediate tables like GroupMembership.
- **Data Privacy:** Implemented access controls and hashing for sensitive fields.
- **Alert Logic:** Used scheduled checks to evaluate mood trends.
- **GUI Synchronization:** Used threading to keep GUI responsive during DB ops.

APPENDICES

A. Full ERD: See attached ERD diagram (emoticaERD.pdf)
B. Full Schema Code: See SQL file (emoticaSQL.sql)
C. Screenshots: (Attach GUI screenshots if applicable)