



MIND CARE

*A Mental Health Patient
Management System*

Opting for Sanity with OOP

*A Project by
Syeda Khadija Hassan*



Mental Health Care – Patient Management System (MHC-PMS)

Project Proposal

1. Introduction

In recent years, mental health has gained significant awareness, breaking long-standing stigmas and encouraging individuals to seek timely help. However, mental health care often faces challenges in maintaining organized records, tracking patient progress, and ensuring regular communication between patients and therapists.

The **Mental Health Care – Patient Management System (MHC-PMS)** also named as **MindCare** is a C++-based software designed to streamline the process of managing patients, therapists, and administrative staff in a mental health care setting. In recent years, **mental health awareness** has significantly increased, highlighting the urgent need for effective, secure, and user-friendly management solutions.

This system integrates **role-based access**, **mood tracking**, **questionnaire assessments**, and **daily positive reminders** to not only manage patient data but also contribute actively to improving patient well-being.

2. Need for the Project

- **Growing Mental Health Awareness:** With increasing recognition of mental health issues, there is a higher demand for organized patient management systems.
- **Efficient Record-Keeping:** Manual systems are prone to errors, data loss, and inefficiency.
- **Personalized Care:** Patients benefit when their progress, mood patterns, and therapeutic needs are regularly monitored.

- **Bridging Communication Gaps:** A well-structured system improves coordination between patients, therapists, and administrative staff.

Traditional record-keeping methods are prone to errors, inefficiency, and data breaches. The **MHC-PMS** addresses these challenges by providing:

- **Secure** storage of patient information.
- **Automated** scheduling and reminders.
- Tools to monitor patient **emotional well-being** over time.

By merging administrative efficiency with mental health support tools, the system ensures both **better patient care** and **optimized clinical workflows**.

3. Core Functionalities

3.1 User Roles

- **Receptionist** – Add/edit patient records, manage appointments, generate invoices.
- **Therapist** – Access patient histories, update treatment plans, review questionnaire results, and monitor mood tracker data.
- **Patient** – View upcoming appointments, track mood, fill out questionnaires, receive daily reminders.

3.2 Shared Features

- **Login/Logout** functionality for all users.
 - **Data validation** to ensure correct and complete inputs.
 - **Role-based access control** to prevent unauthorized data access.
-

4. Additional Features

- **Daily Reminders** – Sends motivational quotes or reminders to patients to invoke positivity.
- **Mood Tracker** – Patients record their mood on a scale of 1–10; therapists can review trends.
- **Questionnaire Module** – Mental health assessment tool that calculates a risk score.
- **Report Generation** – For administrative and clinical purposes.
- **Search & Filter** – Quickly locate patient records or appointments.

5. Methods to Achieve the Features (C++ Implementation Plan)

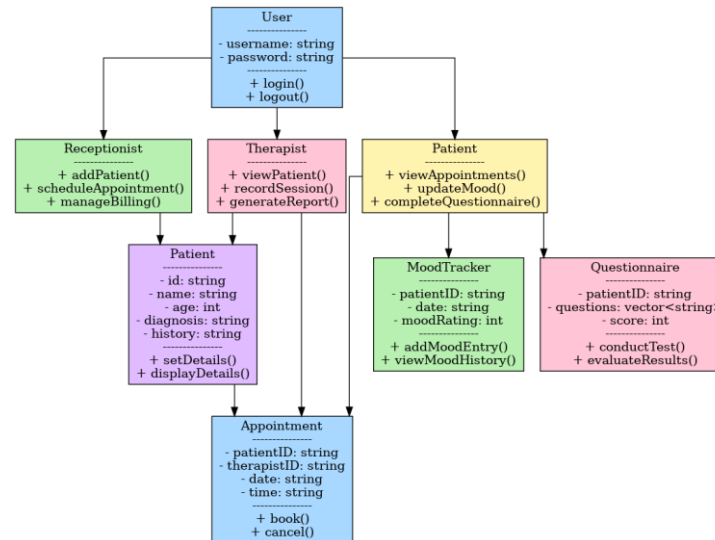
Feature	Implementation in C++
Role-Based Login	Base class <code>User</code> → Derived classes <code>Receptionist</code> , <code>Therapist</code> , <code>Patient</code> . Store credentials in a file, authenticate on login.
Patient Record Management	<code>Patient</code> class with private attributes (ID, name, age, diagnosis, history). Store in <code>patients.txt</code> using <code>fstream</code> .
Appointment Scheduling	<code>Appointment</code> class with date/time attributes. Conflict-checking done by reading existing appointments from file before adding.
Therapy Notes	<code>Treatment</code> or <code>Session</code> class linked to patient ID. Append notes to a file like <code>patientID_sessions.txt</code> .
Mood Tracker	<code>MoodTracker</code> class with <code>patientID</code> , date, mood rating. Append daily entries to file and show history on request.
Daily Positivity Reminders	Store a set of motivational quotes in a file or array; display a random one using <code>rand()</code> .
Questionnaire	<code>Questionnaire</code> class with predefined questions; calculate a score; trigger alert if score exceeds threshold.
Search & Filter	Functions to iterate through file data, applying keyword or condition checks.
Reports	Polymorphic <code>generateReport()</code> method — different details for receptionist, therapist, and patient.
Exception Handling	Try-catch blocks for invalid date formats, duplicate IDs, file errors.
Data Backup	Copy existing data files into a backup folder using file I/O operations.

6. Tools & Technologies

- **Language:** C++ (OOP principles: Encapsulation, Inheritance, Polymorphism, Abstraction, File Handling).
- **Data Storage:** Text files (.txt) using `fstream`.
- **Libraries:** `<iostream>`, `<fstream>`, `<string>`, `<vector>`, `<ctime>`, `<cstdlib>`.
- **Platform:** Console-based application (Windows/Linux).

7. System Design (UML Class Diagram)

The UML diagram illustrates the **class structure** and **relationships** in MHC-PMS.



8. Design Justification

1. **Inheritance for Role Management** – `User` is the base class; specialized classes extend it for unique features.
 2. **Encapsulation** – Private attributes secure sensitive data; getters/setters control access.
 3. **Polymorphism** – Functions like `generateReport()` work differently depending on user role.
 4. **Associations** – Relationships between `Therapist`, `Patient`, `Appointment`, and `MoodTracker` mimic real-world workflows.
-

9. Sample data for Implementation

Use the following data to login as:

	Username	Password
Patient	khadija_hassan	password123
Therapist	dr_mohsin	therapy123
Admin	reception	admin123

10. Conclusion

The MHC-PMS will provide a **secure, efficient, and supportive environment** for mental health care providers and patients. It integrates **administrative efficiency** with **well-being features**, making it an essential tool for modern mental health care facilities.

11. Future Enhancements

- Mobile application integration for remote access.
- AI-based mood analysis for more accurate mental health tracking.
- Secure cloud storage for real-time data access and backup.