CSE 404

Artificial Intelligence and Expert Systems Lab

Project Report - 1

Submission Date: 2/8/2025

Submitted by	Submitted to
Name: Noor Mohammed Priom	Bidita Sarkar Diba
Registration ID: 22101134	Lecturer
Section: C-2	Department of Computer
Semester: 4-1	Science & Engineering
Session: Spring 2025	University of Asia Pacific

Table of Contents

I. Problem Title

Hospital Management System using Rule-Based Prolog.
This project simulates and models the primary operations of a hospital, such as patient registration, doctor assignment, appointments, diagnosis, admission, and room management through logical rules and dynamic facts in Prolog.

II. Problem Description

Hospitals entail intricate, real-time patient record management, medical staff, room assignments, and treatment monitoring. Conventional procedural programming languages can render it cumbersome to model logic-intensive systems like these.

This project uses Prolog, which is a declarative logic programming language, to develop a knowledge-based system representing:

- Patients with attributes such as ID, name, age
- Physicians and their specializations
- Appointment schedules
- Medical treatments and diagnoses
- Admission status and room assignments

it supports:

- Inserting, displaying, and removing patient records
- Scheduling appointments
- Doctor and room assignment
- Dynamic querying of patient health status

III. Tools and Languages Used

- Language: Prolog (SWI-Prolog)
- Editor: Visual Studio Code
- Operating System: (Windows)

IV. Diagram/Figure

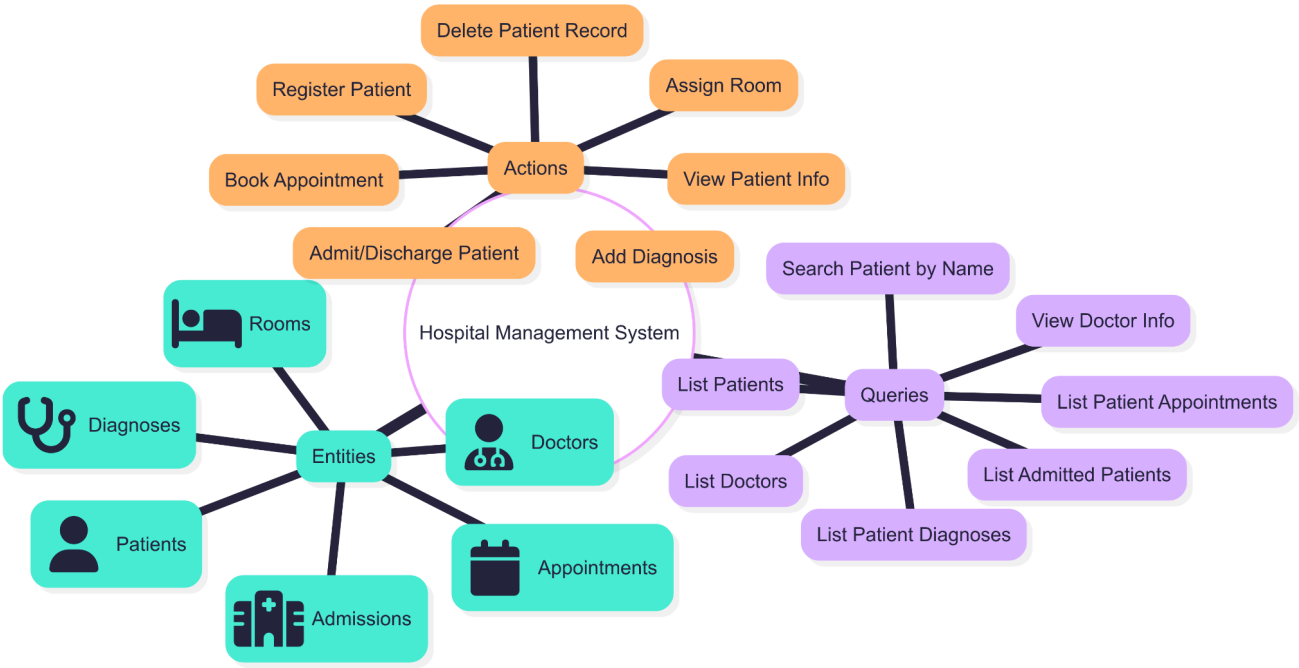

Diagram 1: Management Activity, Entity & Queries

V. Sample Input/output

Patient Register-

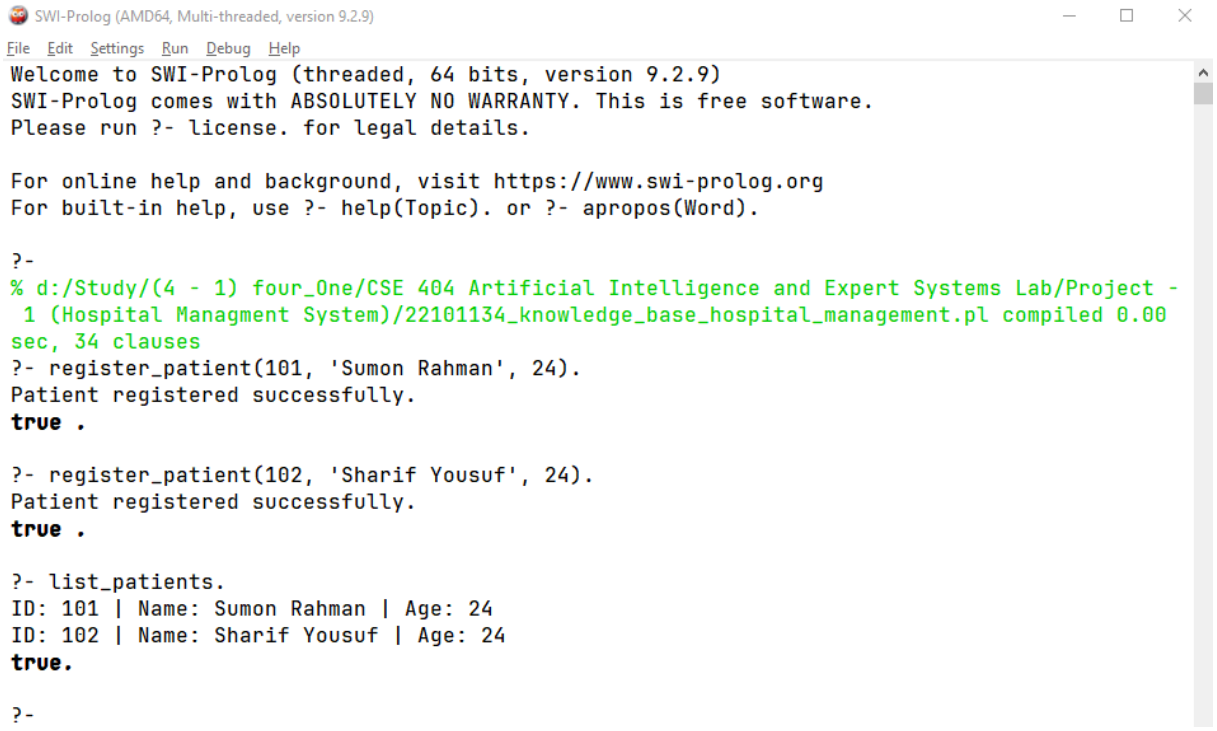

Diagram 2: Patient Register I/O

Patient diagnostic & room assignments-

Diagram 3: Patient Treatment I/O

Patient discharge & deletion-

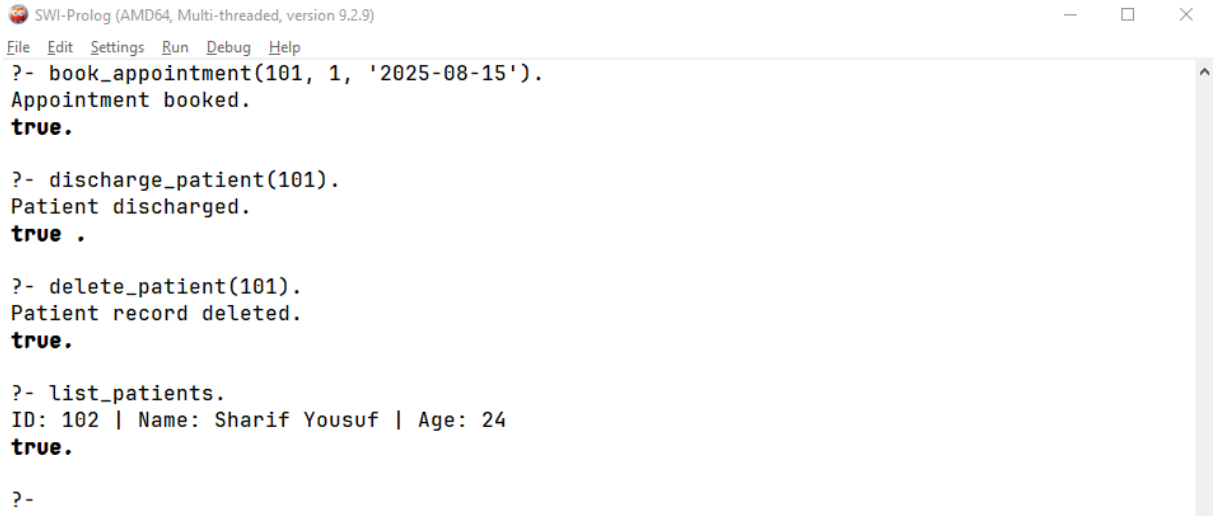

Diagram 4: Patient Discharge I/O

VI. Conclusion

Prolog hospital management system expressly illustratively portrays how the logic programming may further be expanded to include volatile and volatile data like doctor and room allocation, diagnoses, appointments, and patient profiles. The system, like realizing real-word entities as facts and operations as rules, provides an extensible and interactive approach of hospital administrative operations simulations.

This project demonstrates Prolog's ability for:

- Representing knowledge declaratively
- State management with dynamic predicates
- Performing rule-based reasoning and making decisions

Its modularity allows it to be expanded further—i.e., adding billing, emergency management, or analytics—showing how systems of logic can be scaled up.

VII. Challenges

Providing safe usage of dynamic predicates (assert, retract) - Treating negation-as-failure (+) with care - Avoiding logical duplicates (e.g., several admissions or room assignments) - Realistic user flow simulation in a declarative paradigm.

1. Handling Dynamic Data
   - The coordination of numerous changeable entities (rooms, doctors, and patients) needed correct utilization of **retract** and **assert**.
   - To maintain data consistency upon deleting or updating patient records presented design complexity.

2. User Interact Restriction
   - Prolog doesn't have GUI nor structured console input/output, so the interaction remains text-based and linear.
   - Easier-to-use front-end would be best for improving usability.

3. Error Handling
   - Prolog's error-driven approach (+, fail) does not give easy and fine-grained error reporting.
   - For clarity, proper control flow both for failure and for success was needed.

4. Visualization & Debugging:
   - Since Prolog relies on logical deduction, backtracking without the aid of visualization or breakpoints may be extremely hard.
   - The construction of mindmaps and diagrammatic representations supported system structure explanation.

5. Scalability for Real Use

While the current version works for simulation and education, real-world deployment would require features like:

- File I/O for persistence
- Security & access control
- Performance optimization