



CSE 404

Artificial Intelligence and Expert Systems Lab

Project Report - 1: Hospital Management System using Rule-Based Prolog.

Submission Date: 2nd August, 2025

Submitted by

Name: Noor Mohammed Priom

Registration ID: 22101134

Section: C-2

Semester: 4-1

Session: Spring 2025

Submitted to

Bidita Sarkar Diba

Lecturer

Department of Computer

Science & Engineering

University of Asia Pacific

Table of Contents

Table of Contents.....	2
I. Problem Title.....	3
II. Problem Description.....	3
III. Tools and Languages Used.....	3
IV. Diagram/Figure.....	4
V. Sample Input/output.....	5
VI. Conclusion.....	9
VII. Challenges.....	9
VIII. GitHub Link.....	10

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

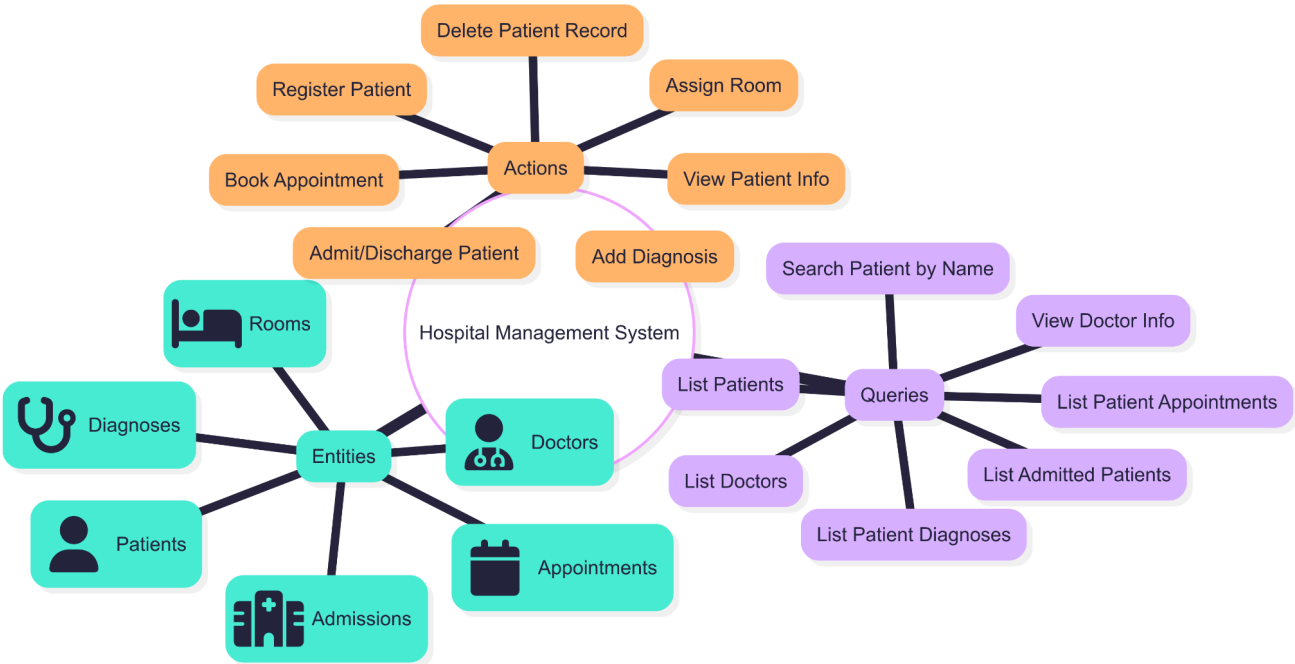
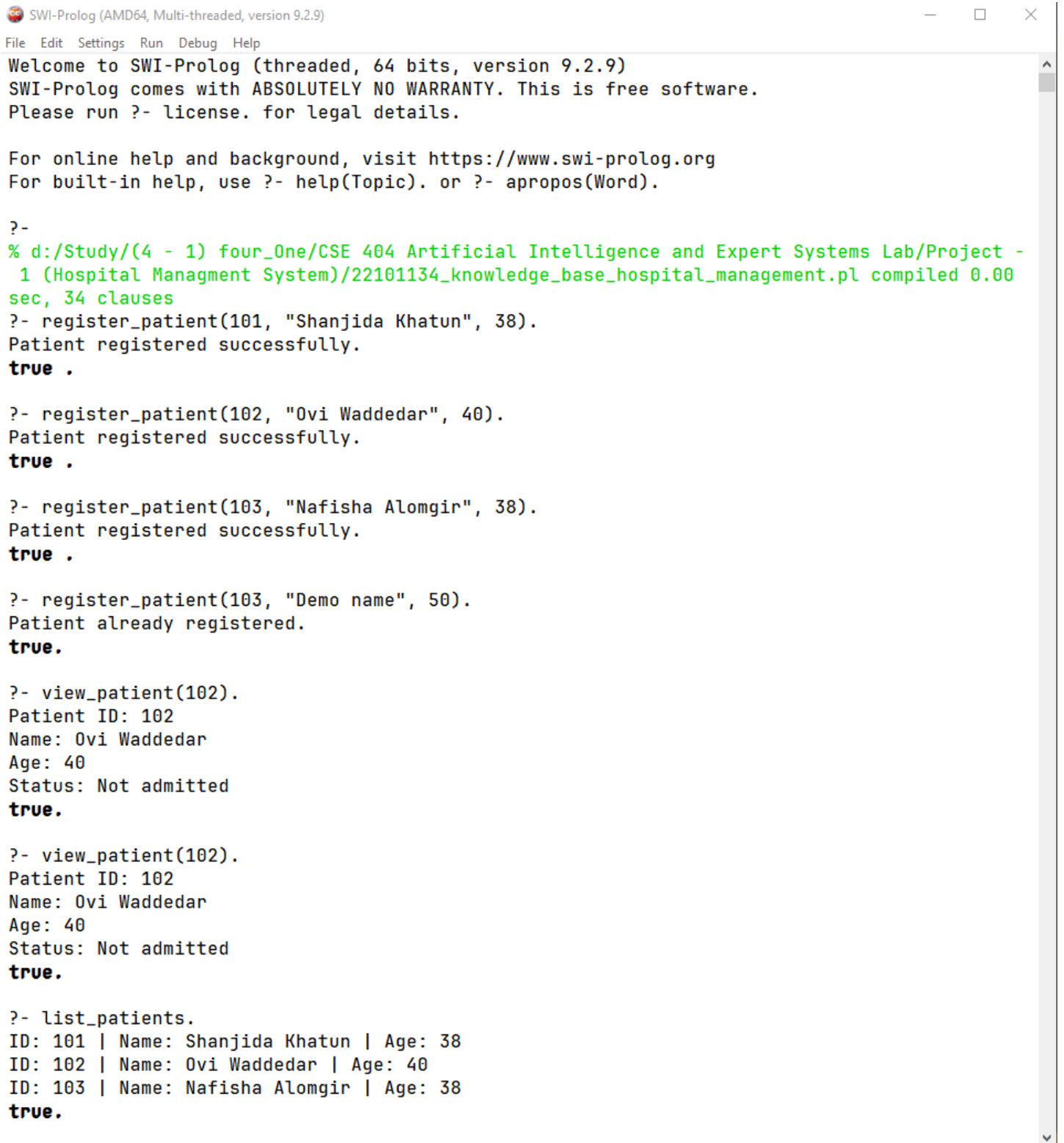


Diagram 1: Management Activity, Entity & Queries

V. Sample Input/output

Patient Register-

The image is a screenshot of the SWI-Prolog IDE window. The title bar reads "SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)". The menu bar includes "File", "Edit", "Settings", "Run", "Debug", and "Help". The main text area contains the following text:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/Study/(4 - 1) four_One/CSE 404 Artificial Intelligence and Expert Systems Lab/Project -
1 (Hospital Managment System)/22101134_knowledge_base_hospital_management.pl compiled 0.00
sec, 34 clauses
?- register_patient(101, "Shanjida Khatun", 38).
Patient registered successfully.
true .

?- register_patient(102, "Ovi Waddedar", 40).
Patient registered successfully.
true .

?- register_patient(103, "Nafisha Alomgir", 38).
Patient registered successfully.
true .

?- register_patient(103, "Demo name", 50).
Patient already registered.
true.

?- view_patient(102).
Patient ID: 102
Name: Ovi Waddedar
Age: 40
Status: Not admitted
true.

?- view_patient(102).
Patient ID: 102
Name: Ovi Waddedar
Age: 40
Status: Not admitted
true.

?- list_patients.
ID: 101 | Name: Shanjida Khatun | Age: 38
ID: 102 | Name: Ovi Waddedar | Age: 40
ID: 103 | Name: Nafisha Alomgir | Age: 38
true.
```

Diagram 5.1: Patient Register I/O

Patient diagnostic & room assignments-

SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)

File Edit Settings Run Debug Help

```
?- add_diagnosis(101, "hypertension", "Low-sodium diet and medication").
Diagnosis added.
true.

?- add_diagnosis(102, "fever", "Paracetamol and rest").
Diagnosis added.
true.

?- add_diagnosis(103, "fractured arm", "X-ray, plaster cast, and pain killers").
Diagnosis added.
true.

?- list_diagnosis(101).
Diagnosis: hypertension | Treatment: Low-sodium diet and medication
true.

?- list_diagnosis(102).
Diagnosis: fever | Treatment: Paracetamol and rest
true.

?- assign_room(103, "Ward-2, Room-5").
Patient must be admitted to assign a room.
true .

?- admit_patient(103).
Patient admitted.
true .

?- assign_room(103, "Ward-2, Room-5").
Room assigned.
true .

?- view_patient(103).
Patient ID: 103
Name: Nafisha Alomgir
Age: 38
Status: Admitted
Room: Ward-2, Room-5
Diagnosis: fractured arm | Treatment: X-ray, plaster cast, and pain killers
true.

?- list_admitted_patients.
ID: 103 | Name: Nafisha Alomgir | Room: Ward-2, Room-5
true.

?-
```

Diagram 5.2: Patient Treatment I/O

Patient booking appointment-

SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)

File Edit Settings Run Debug Help

```
?- book_appointment(103, 1, '2025-08-03').
Appointment booked.
true.

?- book_appointment(102, 1, '2025-09-02').
Appointment booked.
true.

?- book_appointment(101, 1, '2025-10-01').
Appointment booked.
true.

?- list_appointments(101).
Appointment on 2025-10-01 with Dr. Khairul Sharif (cardiology)
true.

?- list_appointments(102).
Appointment on 2025-09-02 with Dr. Khairul Sharif (cardiology)
true.

?- list_appointments(103).
Appointment on 2025-08-03 with Dr. Khairul Sharif (cardiology)
true.

?- book_appointment(101, 2, '2025-10-04').
Appointment booked.
true.

?- list_appointments(101).
Appointment on 2025-10-01 with Dr. Khairul Sharif (cardiology)
Appointment on 2025-10-04 with Dr. Sourav Korimuddin (surgery)
true.

?- view_patient(101).
Patient ID: 101
Name: Shanjida Khatun
Age: 38
Status: Not admitted
Diagnosis: hypertension | Treatment: Low-sodium diet and medication
Appointment on 2025-10-01 with Dr. Khairul Sharif (cardiology)
Appointment on 2025-10-04 with Dr. Sourav Korimuddin (surgery)
true.

?-
```

Diagram 5.3: Patient Appointment I/O

Patient discharge & deletion-

SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)

File Edit Settings Run Debug Help

```
?- list_doctors.  
ID: 1 | Name: Dr. Khairul Sharif | Specialty: cardiology  
ID: 2 | Name: Dr. Sourav Korimuddin | Specialty: surgery  
ID: 3 | Name: Dr. Solimuddin Faisal | Specialty: pediatrics  
true.  
  
?- view_doctor(1).  
Doctor ID: 1  
Name: Dr. Khairul Sharif  
Specialty: cardiology  
true .  
  
?- search_patient_by_name("Ovi Waddedar").  
ID: 102 | Name: Ovi Waddedar | Age: 40  
Search complete.  
true.  
  
?- list_admitted_patients.  
ID: 103 | Name: Nafisha Alomgir | Room: Ward-2, Room-5  
true.  
  
?- discharge_patient(103).  
Patient discharged.  
true .  
  
?- list_admitted_patients.  
No admitted patients.  
true .  
  
?- list_patients.  
ID: 101 | Name: Shanjida Khatun | Age: 38  
ID: 102 | Name: Ovi Waddedar | Age: 40  
ID: 103 | Name: Nafisha Alomgir | Age: 38  
true.  
  
?- delete_patient(103).  
Patient record deleted.  
true.  
  
?- list_patients.  
ID: 101 | Name: Shanjida Khatun | Age: 38  
ID: 102 | Name: Ovi Waddedar | Age: 40  
true.  
  
?-
```

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-world 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

VIII. GitHub Link

Hospital Management System Prolog Knowledgebase: [Hospital Management System](#)