Assignment: Knowledge Base Project & Expert System Development

Knowledge Engineering

Submission Deadline: 13 April 2025
Total Marks: 12

Contact:
seddikmedtakieddine@gmail.com
m.seddik@univ-batna2.dz

Objective

Design a knowledge base project and implement a **GUI-based expert system** using either the **CommonKADS** or **MASK** methodology. Focus on systematic modeling and translating domain knowledge into a functional system.

Phases of the Project

Phase 1: Project Selection

- Choose a real-world domain (e.g., medical diagnosis, academic advising, troubleshooting).
- Avoid overly simple domains (e.g., "tomato plant disease diagnosis" is better than "weather prediction").
- Select **CommonKADS** or **MASK** (no justification required).

Phase 2: Knowledge Modeling

- Model the domain using your chosen methodology.
- **Discover the required components** of the methodology (e.g., diagrams, workflows).
- Use flowcharts or UML diagrams to represent your model.

Phase 3: Expert System Implementation

- Develop a **GUI-based expert system** in Python (preferred) or another language (e.g., Java, C#).
- Required components:
  - **Knowledge Base**: Rules/facts (e.g., JSON/XML files or Python classes).
  - **Inference Engine**: Reasoning logic (e.g., forward/backward chaining).
  - **GUI Interface**: Interactive graphical interface (e.g., Tkinter, PyQt, JavaFX).

Phase 4: Testing & Validation

- Test the system with **5–10 real-world scenarios**.
- Refine the system based on test results.

Deliverables

- **Code**: Well-commented source code + instructions to run the GUI.
- **Presentation (10–15 minutes)**:
  - Explain the modeling process (methodology, diagrams, challenges).
  - Demo the GUI and discuss test results.

Grading Criteria

Criteria	**Marks**
Modeling accuracy	4
System functionality (GUI + logic)	4
Code quality & documentation	2
Presentation clarity & demo	2
Total	**12**

Tips

- Use tools like Draw.io for diagrams.
- Build the GUI incrementally alongside the logic.
- Python libraries like Experta can simplify rule-based systems or program the expert system from scratch(scan the code for an example).

Figure 1: Python Sample

Example Projects

- Plant Disease Diagnosis (e.g., tomato, wheat)
- Career Path Advisor
- PC Hardware Troubleshooter

Submission

Upload your **code** and **presentation slides** to my Email by **13 April 2025**. Late submissions will not be accepted.