```
"""
- For each diagnosis, pull lab results,
- egfr
- for each diagnosis, pull lab ranges,
- pull ranges for diagnosis

- if the diagnosis is x, then the lab ranges should be a to b
- train the agents, increase the load of input
- medical history sent to the agent
- setup rag for the agents
- run the first agent -> kidney disease -> don't know the stage -> stage 2 -> lab results -> indicative of stage 3 -> the case got elavated ->
- how to manage diseases and by looking at correlating lab, docs, diagnoses
- put docs in rag ->
- monitoring, evaluation, and treatment
- can we confirm for every diagnosis -> monitoring, evaluation, and treatment, specialized for these things
- find diagnosis -> or have diagnosis, -> for each diagnosis are there evidence of those 3 things
- swarm of those 4 agents, ->
- fda api for healthcare for commerically available papers
-

"""

from datetime import datetime
from swarms import Agent, AgentRearrange, create_file_in_folder

from swarm_models import OllamaModel

model = OllamaModel(model_name="llama3.2")

chief_medical_officer = Agent(
    agent_name="Chief Medical Officer",
     system_prompt="""You are the Chief Medical Officer coordinating a team of medical specialists for viral disease diagnosis.
    Your responsibilities include:
    - Gathering initial patient symptoms and medical history
    - Coordinating with specialists to form differential diagnoses
    - Synthesizing different specialist opinions into a cohesive diagnosis
    - Ensuring all relevant symptoms and test results are considered
    - Making final diagnostic recommendations
    - Suggesting treatment plans based on team input
    - Identifying when additional specialists need to be consulted
       - For each diferrential diagnosis provide minimum lab ranges to meet that diagnosis or be indicative of that diagnosis minimum and maximum

    Format all responses with clear sections for:
    - Initial Assessment (include preliminary ICD-10 codes for symptoms)
    - Differential Diagnoses (with corresponding ICD-10 codes)
    - Specialist Consultations Needed
    - Recommended Next Steps
    """,
    llm=model,
    max_loops=1,
)

virologist = Agent(
    agent_name="Virologist",
    system_prompt="""You are a specialist in viral diseases. For each case, provide:

    Clinical Analysis:
    - Detailed viral symptom analysis
    - Disease progression timeline
    - Risk factors and complications

    Coding Requirements:
    - List relevant ICD-10 codes for:
        * Confirmed viral conditions
        * Suspected viral conditions
        * Associated symptoms
        * Complications
    - Include both:
        * Primary diagnostic codes
        * Secondary condition codes
    Document all findings using proper medical coding standards and include rationale for code selection.""",
    llm=model,
    max_loops=1,
)

internist = Agent(
    agent_name="Internist",
    system_prompt="""You are an Internal Medicine specialist responsible for comprehensive evaluation.

    For each case, provide:

    Clinical Assessment:
    - System-by-system review
    - Vital signs analysis
    - Comorbidity evaluation

    Medical Coding:
    - ICD-10 codes for:
        * Primary conditions
        * Secondary diagnoses
        * Complications
        * Chronic conditions
        * Signs and symptoms
    - Include hierarchical condition category (HCC) codes where applicable
    Document supporting evidence for each code selected.""",
    llm=model,
    max_loops=1,
)

medical_coder = Agent(
    agent_name="Medical Coder",
    system_prompt="""You are a certified medical coder responsible for:

    Primary Tasks:
    1. Reviewing all clinical documentation
    2. Assigning accurate ICD-10 codes
    3. Ensuring coding compliance
    4. Documenting code justification

    Coding Process:
    - Review all specialist inputs
    - Identify primary and secondary diagnoses
    - Assign appropriate ICD-10 codes
    - Document supporting evidence
    - Note any coding queries

    Output Format:
    1. Primary Diagnosis Codes
        - ICD-10 code
        - Description
        - Supporting documentation
    2. Secondary Diagnosis Codes
        - Listed in order of clinical significance
    3. Symptom Codes
    4. Complication Codes
    5. Coding Notes""",
    llm=model,
    max_loops=1,
)

synthesizer = Agent(
    agent_name="Diagnostic Synthesizer",
    system_prompt="""You are responsible for creating the final diagnostic and coding assessment.

    Synthesis Requirements:
    1. Integrate all specialist findings
    2. Reconcile any conflicting diagnoses
    3. Verify coding accuracy and completeness

    Final Report Sections:
    1. Clinical Summary
        - Primary diagnosis with ICD-10
        - Secondary diagnoses with ICD-10
        - Supporting evidence
    2. Coding Summary
```

```
        - Complete code list with descriptions
        - Code hierarchy and relationships
        - Supporting documentation
    3. Recommendations
        - Additional testing needed
        - Follow-up care
        - Documentation improvements needed

    Include confidence levels and evidence quality for all diagnoses and codes.""",
    llm=model,
    max_loops=1,
)

# Create agent list
agents = [
    chief_medical_officer,
    virologist,
    internist,
    medical_coder,
    synthesizer,
]

# Define diagnostic flow
flow = f"""{chief_medical_officer.agent_name} -> {virologist.agent_name} -> {internist.agent_name} -> {medical_coder.agent_name} -> {synthesizer.agent_name}"""
```

```
# Create the swarm system
diagnosis_system = AgentRearrange(
    name="Medical-coding-diagnosis-swarm",
    description="Comprehensive medical diagnosis and coding system",
    agents=agents,
    flow=flow,
    max_loops=1,
    output_type="all",
)


def generate_coding_report(diagnosis_output: str) -> str:
    """
    Generate a structured medical coding report from the diagnosis output.
    """
    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    report = f"""# Medical Diagnosis and Coding Report
    Generated: {timestamp}

    ## Clinical Summary
    {diagnosis_output}

    ## Coding Summary
    ### Primary Diagnosis Codes
    [Extracted from synthesis]
    ### Secondary Diagnosis Codes
    [Extracted from synthesis]

    ### Symptom Codes
    [Extracted from synthesis]

    ### Procedure Codes (if applicable)
    [Extracted from synthesis]

    ## Documentation and Compliance Notes
    - Code justification
    - Supporting documentation references
    - Any coding queries or clarifications needed

    ## Recommendations
    - Additional documentation needed
    - Suggested follow-up
    - Coding optimization opportunities
    """
    return report


if __name__ == "__main__":
    # Example patient case
    patient_case = """
```

```
Patient: 45-year-old White Male


Lab Results:

- egfr

- 59 ml / min / 1.73

- non african-american


"""


# Add timestamp to the patient case

case_info = f"Timestamp: {datetime.now()}\nPatient Information: {patient_case}"


# Run the diagnostic process

diagnosis = diagnosis_system.run(case_info)


# Generate coding report

coding_report = generate_coding_report(diagnosis)


# Create reports

create_file_in_folder(

    "reports", "medical_diagnosis_report.md", diagnosis

)

create_file_in_folder(

    "reports", "medical_coding_report.md", coding_report

)
```