Implementation of a Multi-Agent System for Emergency Response

Team 05

January 11, 2025

Overview of the Project

Agents and Tasks

Crews Module

Data Models

Database and Tools

GPS Tools

Scripts

Test Script

Test Folder

emergency_planner/main.py

```python
class EmergencyPlannerFlow(Flow[EmergencyPlannerState]):
    @start()
    def get_call_transcript(self): ...

    @listen(get_call_transcript)
    def emergency_services(self): ...

    @listen(emergency_services)
    def firefighters(self): ...

    @listen(emergency_services)
    def medical_services(self):
        if not self.state.call_assessment.medical_services_required:
            return
        ...

    @listen(or_(and_(firefighters, medical_services), "
        retry_public_communication"))
    def public_communication(self): ...

    @router(public_communication)
    def check_approval(self): ...
```

data/

```
@start()
def get_call_transcript(self):
    with open(EMERGENCY_CALL_TRANSCRIPTS_FILENAME, "r"
        ) as f:
        self.state.call_transcript = f.readlines()[
            TRANSCRIPT_INDEX]
```

```
@listen("save full emergency report")
def save_full_emergency_report(self):
    full_emergency_report = f"""
# Emergency Report

## Call Transcript
{self.state.call_transcript}
...
"""
with open(EMERGENCY_REPORT_FILENAME, "w") as f:
    f.write(full_emergency_report)
```

- ∨ data
  - ∨ inputs
    - ◈ .gitkeep
    - ≡ call_transcripts.txt
    - ≡ lloretDeMar.graphml
  - ∨ outputs
    - ◈ .gitkeep
    - ⬇ emergency_report_2.md
    - ⬇ emergency_report.md
    - ≡ logs1.txt
    - ≡ logs2.txt

Thank you!

Questions?