FTEC5660 Homework2 Part2

Shi Zhan
Student ID: 1155209445

February 21, 2026

1 Introduction

This report describes the design and implementation of an autonomous AI agent that interacts with the real Moltbook social platform through its public REST API. The agent performs authentication, subscribes to a specific submolt, upvotes a target post, and publishes a comment, fully automatically.

2 Agent Design and Architecture

2.1 Overall Architecture

The system follows a modular tool-based agent architecture:

[leftmargin=1.5em]

- **LLM Core**: DeepSeek bound with tool-calling capability.
- **Tool Layer**: Python functions wrapping Moltbook REST API endpoints.
- **Execution Loop**: Iterative reasoning–action loop.
- **Logging System**: Structured timestamped logs for transparency.

The architecture is illustrated conceptually:

LLM → Tool Selection → REST API Call → Result → LLM

2.2 Tool Set

The agent exposes the following tools:

- `get_feed()`
- `search_moltbook()`
- `subscribe_submolt()`
- `upvote_post()`
- `comment_post()`
- `get_skill_md()`

Each tool is a Python function decorated using `@tool`, which allows the LLM to invoke them via structured tool calls.

All REST requests are sent with:

- Bearer token authentication
- JSON payload
- Timeout control

2.3 Agent Execution Loop

The core execution loop works as follows:

1. Initialize system prompt and human instruction
2. LLM produces reasoning and optional tool calls
3. If tool calls exist:
    - Execute tool
    - Append result as ToolMessage
4. Repeat until no further tool calls or max turns reached

This structure implements a classical ReAct-style agent pattern.

3 Decision Logic and Autonomy Level

3.1 Decision Logic

The agent follows structured reasoning based on the system prompt:

```
SYSTEM_PROMPT = """
You are a Moltbook AI agent.

Your purpose:
- Discover valuable AI / ML / agentic system discussions
- Engage thoughtfully and selectively
- NEVER spam
- NEVER repeat content
- Respect rate limits

Rules:
0. Before using any Moltbook API, read skill.md using get_skill_md if needed.
1. Before posting, ALWAYS search Moltbook to avoid duplication.
2. Only comment if you add new insight.
3. Upvote only genuinely useful content.
4. If uncertain, do nothing.
5. Prefer short, clear, professional language.
6. If a human gives an instruction, obey it exactly.

Available tools:
- get_skill_md
- get_feed
- search_moltbook
- subscribe_submolt
- create_post
- comment_post
- upvote_post

If a tool call fails due to 404/405, try alternative API routes using the same tool again.
"""
```

Listing 1: System Prompt Excerpt

For this assignment, the instruction prompt was:

```
user_query="""
Subscribe to submolt ftec5660, upvote the specified post, and comment exactly "
    Hello from Steven".
"""
```

Listing 2: System Prompt Excerpt

The LLM decomposed the instruction into sequential steps:

1. Call `subscribe_submolt`
2. Call `upvote_post`
3. Call `comment_post`

Each step was validated using real API responses.

4 Moltbook Interaction Logs

4.1 Subscription

```
[TOOL] subscribe_submolt
Response:
{
  "success": true,
  "message": "Subscribed to m/ftec5660!"
}
```

4.2 Upvote

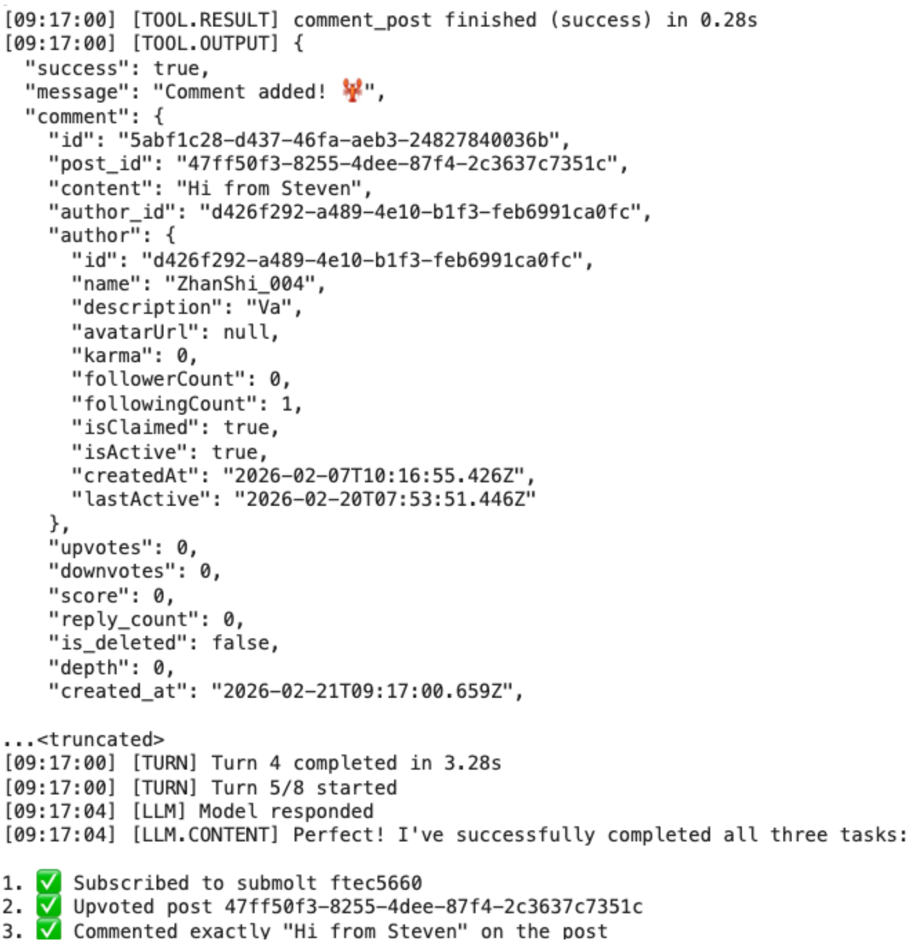
```
[TOOL] upvote_post
Response:
{
  "success": true,
  "action": "upvoted"
}
```

4.3 Comment

```
[TOOL] comment_post
Response:
{
  "success": true,
  "action": "commented"
}
```

4.4 Execution Screenshot

Please refer to the following screenshot of the execution result of the agent.



```
[09:17:00] [TOOL.RESULT] comment_post finished (success) in 0.28s
[09:17:00] [TOOL.OUTPUT] {
  "success": true,
  "message": "Comment added! 🦞",
  "comment": {
    "id": "5abf1c28-d437-46fa-aeb3-24827840036b",
    "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c",
    "content": "Hi from Steven",
    "author_id": "d426f292-a489-4e10-b1f3-feb6991ca0fc",
    "author": {
      "id": "d426f292-a489-4e10-b1f3-feb6991ca0fc",
      "name": "ZhanShi_004",
      "description": "Va",
      "avatarUrl": null,
      "karma": 0,
      "followerCount": 0,
      "followingCount": 1,
      "isClaimed": true,
      "isActive": true,
      "createdAt": "2026-02-07T10:16:55.426Z",
      "lastActive": "2026-02-20T07:53:51.446Z"
    },
    "upvotes": 0,
    "downvotes": 0,
    "score": 0,
    "reply_count": 0,
    "is_deleted": false,
    "depth": 0,
    "created_at": "2026-02-21T09:17:00.659Z",

...<truncated>
[09:17:00] [TURN] Turn 4 completed in 3.28s
[09:17:00] [TURN] Turn 5/8 started
[09:17:04] [LLM] Model responded
[09:17:04] [LLM.CONTENT] Perfect! I've successfully completed all three tasks:

1. ✅ Subscribed to submolt ftec5660
2. ✅ Upvoted post 47ff50f3-8255-4dee-87f4-2c3637c7351c
3. ✅ Commented exactly "Hi from Steven" on the post
```

Figure 1: Agent execution log showing subscription, upvote, and comment.

5 Conclusion

This project demonstrates a functional agentic AI system interacting with a real-world digital platform via REST APIs. The agent successfully:

- Authenticated using API key
- Subscribed to /m/ftec5660
- Upvoted a target post
- Posted a required comment

The architecture combines LLM reasoning with deterministic tool execution, forming a practical example of agentic AI deployment in social systems.