

FTEC5660 HW2 Part 2 Report

Name: Chan Chun Siu (1155250674)

Agent ID: Anson_6934324

1. Objective

This report details the design, architecture, and execution logic of an autonomous AI agent built to interact with Moltbook, a simulated social network. The agent is designed to navigate the platform via its REST API, authenticate securely, and perform social actions such as subscribing to communities, reading posts, upvoting, and generating context-aware comments.

2. Agent Design and Architecture

The agent is built using a tool-calling architecture, leveraging the LangChain framework and powered by the `gemini-2.5-flash` Large Language Model (LLM).

2.1 Core Components

- **The "Brain" (LLM):** The Gemini 2.5 Flash model processes system instructions, human commands, and the conversation history to determine the necessary sequence of actions.
- **The "Hands" (Tool Abstraction):** Moltbook's public REST endpoints are encapsulated into Python functions wrapped with LangChain's `@tool` decorator. These tools include explicit docstrings that teach the LLM how and when to use them.
- **The "Memory" (History):** A sequential list of messages (`SystemMessage` , `HumanMessage` , `AIMessage` , and `ToolMessage`) that maintains the state of the agent's operations during the execution loop.

2.2 Tool Set Implementation

Instead of relying on hardcoded scripts, the agent is equipped with a specific toolset mapped to the Moltbook API capabilities:

- `subscribe_submolt` : Subscribes to targeted communities (e.g., `/m/ftec5660`).
- `get_post` : Retrieves the content of a specific post ID to establish context before social interaction.
- `upvote_post` : Sends an HTTP POST request to increase the post's engagement score.
- `comment_post` : Generates and publishes a response to a post.
- `verify_challenge` : A dedicated security tool designed to solve obfuscated mathematical CAPTCHAs required by the Moltbook anti-spam system.

3. Decision Logic and Autonomy Level

The agent operates with a **High Level of Autonomy**, utilizing a dynamic Reason-and-Act (ReAct) loop rather than a predefined, step-by-step procedural script.

3.1 The ReAct Execution Loop

1. **Observe:** The agent parses the system prompt and the human instruction (e.g., "Subscribe to ftec5660, read post 47ff50f3..., upvote, and comment").
2. **Thought:** The LLM internally determines the first logical step (calling `subscribe_submolt`) and halts text generation to issue a `tool_call` .
3. **Act:** The Python backend executes the requested tool and appends the JSON response (e.g., `{"success": true}`) back into the agent's memory.
4. **Iterate:** The agent observes the successful result and automatically decides on the next sequence (e.g., calling `get_post`). This loop continues until all tasks are satisfied.

3.2 Contextual Awareness

To avoid generating generic or hallucinated responses (spam), the agent is explicitly instructed to execute `get_post` *before* attempting to comment. By pulling the post's title and content into its working memory, the agent synthesizes a highly relevant, professional comment that adds genuine value to the discussion.

3.3 Self-Correction and Dynamic Verification

A defining feature of this agent's autonomy is its ability to handle Moltbook's "AI Verification Challenges."

- If a `comment_post` API call returns a `"verification_required": true` status, the agent does not crash.
- Instead, its system prompt instructs it to autonomously extract the `challenge_text` (an obfuscated math problem), calculate the answer to two decimal places, and invoke the `verify_challenge` tool.
- Once the verification is successful, the agent inherently knows to resume its social operations.

4. Execution Logs and Moltbook Interactions

(Below are the execution logs demonstrating the agent's autonomous tool-calling sequence, as well as visual proof of the successful interactions on the Moltbook platform.)

4.1 Terminal Execution Logs

Description: The logs below illustrate the agent's step-by-step execution, including tool calling, parameter mapping, and successful API responses.

```
[09:54:41] [INIT] Starting Moltbook agent loop
[09:54:42] [HUMAN]
You must complete the following mandatory tasks on Moltbook:
1. Subscribe to the submolt named 'ftec5660'.
2. Retrieve and read the post with ID '47ff50f3-8255-4dee-87f4-2c3637c7351c'.
3. Upvote that specific post.
4. Leave a thoughtful, professional comment on that post based on what you just read.

[09:54:42] [TURN] Turn 1/10 started
/tmp/ipython-input-1213938906.py:9: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use
```

```

timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
    ts = datetime.utcnow().strftime("%H:%M:%S")
[09:54:44] [T00L] [1] Calling `subscribe_submolt`
[09:54:44] [T00L.ARGs] {
    "submolt_name": "ftec5660"
}
[09:54:45] [T00L.RESULT] subscribe_submolt finished (success) in 0.53s
[09:54:45] [TURN] Turn 1 completed in 3.27s
[09:54:45] [TURN] Turn 2/10 started
[09:54:46] [T00L] [1] Calling `get_post`
[09:54:46] [T00L.ARGs] {
    "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
}
[09:54:46] [T00L.RESULT] get_post finished (success) in 0.1s
[09:54:46] [TURN] Turn 2 completed in 0.93s
[09:54:46] [TURN] Turn 3/10 started
[09:54:47] [T00L] [1] Calling `upvote_post`
[09:54:47] [T00L.ARGs] {
    "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c"
}
[09:54:47] [T00L.RESULT] upvote_post finished (success) in 0.12s
[09:54:47] [TURN] Turn 3 completed in 1.06s
[09:54:47] [TURN] Turn 4/10 started
[09:54:49] [T00L] [1] Calling `comment_post`
[09:54:49] [T00L.ARGs] {
    "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c",
    "content": "Thank you for creating this dedicated space for FTEC5660. I look forward to engaging with the community and sharing insights related to the course material."
}
[09:54:49] [T00L.RESULT] comment_post finished (success) in 0.17s
[09:54:49] [TURN] Turn 4 completed in 2.41s
[09:54:49] [TURN] Turn 5/10 started
[09:54:51] [T00L] [1] Calling `verify_challenge`
[09:54:51] [T00L.ARGs] {
    "verification_code": "moltbook_verify_5664223469b418aee88d43fa2a48ad4e",
    "answer": "39.00"
}
[09:54:51] [T00L.RESULT] verify_challenge finished (success) in 0.14s
[09:54:51] [TURN] Turn 5 completed in 2.11s
[09:54:51] [TURN] Turn 6/10 started
[09:54:52] [STOP] No tool calls – final answer produced in 0.99s

=== Agent Final Output ===
[{'type': 'text', 'text': "All mandatory tasks have been successfully completed. I have subscribed to 'ftec5660', read the specified post, upvoted it, and left a thoughtful comment, which has also been successfully verified."}]

```

```
d and published.", 'extras': {'signature': 'CtoCAb4+9vt+ED2B8JkLQiiMTfbXXa
WBsGDLZv+wqBg8Q7JZeN8gP1bM/susBqi30Z0DwxMFotP4cpu00EkuqXju0fGKADlcutdJnyq
t/zyAKQfQeiDIaVLJo762tcwBd17Drdd7H8lbrBtF4ljpuF7KqWss/ezIx1JXeynluNg39jim0
4yZphLhvMphfHE03z2/64CVdQyMRjoRJ8So2EE7m6hdzdpI4I/ajcMttJfIWioTxZ4xFkwHEGk
cj3ucxgNIN8jithqnx3K/rHU8skfctiXYrmp0oSU8p5/dzH6lL8BIXC1oN7Td10LdZzfeZJrCU
Idj8JB2Zv06PIbwPnVuEHTW+usc1eDRH0WmpTYq4Rk/qNhucXee/aU+oC3CJm9b0fs++wWthjK
0YY49Q9r7rAB4PQqdYYcNXAsB06EUR10+xdXh3QIG644WHvXpQi/3t1o3QXuhixcIwg=='}}]
```

