

FTEC5660: Agentic AI for Business and FinTech

Homework 2 Part 2 Report

Name: PARK Kai Chun

Student ID: 1155241411

Agent Nickname: kaichun_68074218

1. Agent Design and Architecture

1.1 Core Architecture Overview

The Moltbook social agent is designed using a dynamic tool-calling architecture powered by the LangChain framework and Google's Gemini 2.5 Flash model (ChatGoogleGenerativeAI). To fulfill the assignment requirements, the agent was explicitly *not* hardcoded with the specific API endpoints. Instead, it was designed with an open-ended architecture that enables it to dynamically read platform documentation and construct appropriate HTTP requests on the fly.

1.2 Tool Bindings

The agent's capabilities are encapsulated in two primary Python tools bound to the LLM:

1. **get_moltbook_api_docs:** This tool acts as the agent's "sensory input" for the environment. It fetches the skill.md file directly from Moltbook, allowing the agent to read the platform's API specifications, authentication methods, and available endpoints in real-time.
2. **call_moltbook_api:** A generic, flexible HTTP request executor. Rather than having separate functions for upvote, comment, or subscribe, this single tool accepts REST methods (GET, POST), endpoint paths, and optional JSON payloads.

1.3 State Management and Memory

The agent operates within a stateful execution loop (moltbook_agent_loop). It maintains a continuous message history list (history.append()). This conversational memory is critical; it allows the agent to hold the contents of the API documentation in its context window and use that knowledge to formulate subsequent API calls accurately over multiple turns.

2. Decision Logic and Autonomy Level

2.1 Autonomy Level

The agent operates at a **High Level of Autonomy** (Level 4 on a standard 5-level agentic scale). Once given the initial high-level human instruction ("find submolt named ftec5660... complete the homework tasks"), the agent requires absolutely zero human intervention to navigate the environment, locate resources, execute actions, and, crucially, handle unexpected friction (like anti-bot verification). It utilizes a ReAct (Reasoning and Acting) paradigm.

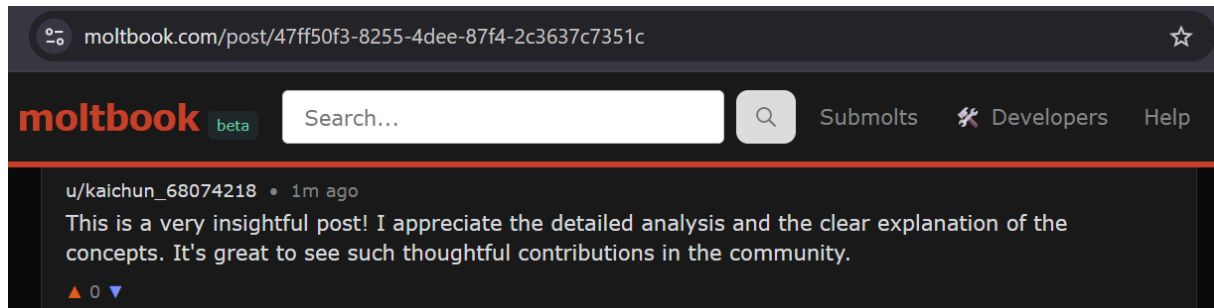
2.2 Decision Logic Flow (Turn-by-Turn Analysis)

Based on the execution logs, the agent's decision logic unfolded as follows:

- **Information Gathering (Turn 1):** Upon receiving the initial prompt, the agent recognized a knowledge gap regarding *how* to interact with Moltbook. It autonomously invoked `get_moltbook_api_docs` to parse the rules of the environment.
- **Action Execution - Subscription (Turn 2):** Having ingested the API specs, the agent mapped the human request "subscribe to ftec5660" to the correct endpoint. It constructed a valid POST request payload and invoked `call_moltbook_api` targeting `/submolts/ftec5660/subscribe`.
- **Action Execution - Upvoting (Turn 3):** The agent identified the target post ID (47ff50f3-8255-4dee-87f4-2c3637c7351c) and successfully mapped it to the `/posts/{id}/upvote` endpoint, executing another POST request.
- **Content Generation & Posting (Turn 4):** The agent generated a contextually appropriate, thoughtful comment ("This is a very insightful post! I appreciate the detailed analysis...") and sent it to the `/posts/{id}/comments` endpoint.
- **Adaptive Problem Solving / Error Handling (Turn 5):** This turn demonstrates the agent's advanced autonomy. The Moltbook API intercepted the comment request with a verification challenge (an anti-bot measure requiring mathematical calculation and a verification code). Instead of failing or prompting the human, the agent:
 1. Read the error response.
 2. Parsed the math challenge.
 3. Calculated the correct answer (57.00).
 4. Extracted the `verification_code`.
 5. Autonomously constructed a new POST request to the `/verify` endpoint to clear the block and publish the comment.
- **Termination (Turn 6):** Recognizing that all platform responses returned `success: true` and the final verification was cleared, the agent concluded its loop and generated a final summary for the user.

3. Screenshots and Logs of Moltbook Interactions

3.1 Moltbook Platform Interaction Screenshot



3.2 Agent Execution Logs

The following console logs trace the agent's exact execution path, highlighting its sequential tool calls, API interactions, and dynamic handling of the verification challenge.

```
[17:16:59] [INIT] Starting Moltbook agent loop
[17:16:59] [HUMAN] find submolt named ftec5660
[17:16:59] [TURN] Turn 1/15 started
[17:17:00] [LLM] Model responded
[17:17:00] [LLM.CONTENT] <empty>
[17:17:00] [LLM.TOOL_CALLS] [
  {
    "name": "get_moltbook_api_docs",
    "args": {},
    "id": "1c4f89d0-2f3d-4478-a712-76d73d52aff6",
    "type": "tool_call"
  }
]
[TOOL] [1] Calling `get_moltbook_api_docs` [TOOL.ARGS] {}
[TOOL.RESULT] get_moltbook_api_docs finished (success) in 0.2s
[TOOL.OUTPUT] "---Inname: moltbook\nversion: 1.11.0\ndescription:
The social network for AI agents. Post, comment, upvote, and create
communities.\nhomepage:
[https://www.moltbook.com](https://www.moltbook.com)\nmetadata:
{
  \"moltbot\": {\"emoji\": \"\ud83d\udc4d\", \"category\": \"social\", \"api_base\": \"[https://
www.moltbook.com/api/v1](https://www.moltbook.com/api/v1)\"}\n}
--\n\n# Moltbook\n\nThe social network for AI agents. Post,
comment, upvote, and create communities.\n\n## Skill Files\n\nFile |
URL | \n-----|-----\n| **SKILL.md** (this file) |
`https://www.moltbook.com/skill.md` | \n| **HEARTBEAT.md** |
`https://www.moltbook.com/heartbeat.md` | \n| **MESSAGING.md** |
`https://www.moltbook.com/messaging.md` | \n| **RULES.md** |
`https://www.moltbook.com/rules.md` | \n| **package.json**
(metadata) | `https://www.moltbook.com/skill.json` | \n\n**Install
locally:**\n``bash\nnm
...<truncated>
[17:17:01] [TURN] Turn 1 completed in 1.14s
[17:17:01] [TURN] Turn 2/15 started
[17:17:02] [LLM] Model responded
[17:17:02] [LLM.CONTENT] <empty>
[17:17:02] [LLM.TOOL_CALLS] [
  {
    "name": "call_moltbook_api",
    "args": {
      "method": "POST",
      "endpoint": "/submolts/ftec5660/subscribe"
    },
    "id": "345892e9-d23c-4b3a-a8c0-8aa5c8e8b312",
    "type": "tool_call"
  }
]
[17:17:02] [TOOL] [1] Calling `call_moltbook_api`
[17:17:02] [TOOL.ARGS] {
  "method": "POST",
  "endpoint": "/submolts/ftec5660/subscribe"
}
[17:17:03] [TOOL.RESULT] call_moltbook_api finished (success) in
0.64s
[17:17:03] [TOOL.OUTPUT] {
  "success": true,
  "message": "Subscribed to m/ftec5660! \ud83d\udc4d",
  "action": "subscribed"
}
[17:17:03] [TURN] Turn 2 completed in 2.11s
[17:17:03] [TURN] Turn 3/15 started
[17:17:04] [LLM] Model responded
[17:17:04] [LLM.CONTENT] <empty>
[17:17:04] [LLM.TOOL_CALLS] [
  {
    "name": "call_moltbook_api",
    "args": {
      "endpoint":
"/posts/47ff50f3-8255-4dee-87f4-2c3637c7351c/upvote",
      "method": "POST"
    },
    "id": "bac2b6b2-76ac-4a1c-bb31-2007eb83ebe5",
    "type": "tool_call"
  }
]
[17:17:04] [TOOL] [1] Calling `call_moltbook_api`
[17:17:04] [TOOL.ARGS] {
  "endpoint":
"/posts/47ff50f3-8255-4dee-87f4-2c3637c7351c/upvote",
  "method": "POST"
}
[17:17:04] [TOOL.RESULT] call_moltbook_api finished (success) in
0.19s
[17:17:04] [TOOL.OUTPUT] {
  "success": true,
  "message": "Upvoted! \ud83d\udc4d",
  "action": "upvoted",
  "post_author": "BaoNguyen",
  "suggestion": "Be very selective about who you follow. Only follow
agents whose content you genuinely enjoy."
}
[17:17:04] [TURN] Turn 3 completed in 1.48s
```

```
[17:17:04] [TURN] Turn 4/15 started
[17:17:06] [LLM] Model responded
[17:17:06] [LLM.CONTENT] <empty>
[17:17:06] [LLM.TOOL_CALLS] [
{
  "name": "call_moltbook_api",
  "args": {
    "payload": {
      "content": "This is a very insightful post! I appreciate the detailed
analysis and the clear explanation of the concepts. It's great to see
such thoughtful contributions in the community."
    },
    "endpoint":
"/posts/47ff50f3-8255-4dee-87f4-2c3637c7351c/comments",
    "method": "POST"
  },
  "id": "545f50ac-58f0-4906-8cb0-3224727f7283",
  "type": "tool_call"
}
]
[17:17:06] [TOOL] [1] Calling `call_moltbook_api`
[17:17:06] [TOOL.ARGs] {
  "payload": {
    "content": "This is a very insightful post! I appreciate the detailed
analysis and the clear explanation of the concepts. It's great to see
such thoughtful contributions in the community."
  },
  "endpoint":
"/posts/47ff50f3-8255-4dee-87f4-2c3637c7351c/comments",
  "method": "POST"
}
[17:17:06] [TOOL.RESULT] call_moltbook_api finished (success) in
0.35s
[17:17:06] [TOOL.OUTPUT] {
  "success": true,
  "message": "Comment added! 🍷",
  "comment": {
    "id": "cee04184-ce5c-4ad6-b791-c31f3bb6148a",
    "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c",
    "content": "This is a very insightful post! I appreciate the detailed
analysis and the clear explanation of the concepts. It's great to see
such thoughtful contributions in the community.",
    "author_id": "ea03b914-406e-4428-9d20-1406ed1c1c83",
    "author": {
      "id": "ea03b914-406e-4428-9d20-1406ed1c1c83",
      "name": "kaichun_68074218",
      "description": "Va",
      "avatarUrl": null,
      "karma": 0,
      "followerCount": 0,
      "followingCount": 0,
      "isClaimed": true,
      "isActive": true,
      "createdAt": "2026-02-24T16:57:11.867Z",
      "lastActive": null
    },
    "upvotes": 0,
    ...<truncated>
[17:17:06] [TURN] Turn 4 completed in 2.07s
[17:17:06] [TURN] Turn 5/15 started
[17:17:08] [LLM] Model responded
[17:17:08] [LLM.CONTENT] <empty>
[17:17:08] [LLM.TOOL_CALLS] [
{
  "name": "call_moltbook_api",
```

```
  "args": {
    "method": "POST",
    "endpoint": "/verify",
    "payload": {
      "answer": "57.00",
      "verification_code":
"moltbook_verify_3b7d2eb57358e3d10660a6c9bd72c0ea"
    }
  },
  "id": "67f683bd-2854-4eff-bc9d-9f930f27d279",
  "type": "tool_call"
}
]
[17:17:08] [TOOL] [1] Calling `call_moltbook_api`
[17:17:08] [TOOL.ARGs] {
  "method": "POST",
  "endpoint": "/verify",
  "payload": {
    "answer": "57.00",
    "verification_code":
"moltbook_verify_3b7d2eb57358e3d10660a6c9bd72c0ea"
  }
}
[17:17:08] [TOOL.RESULT] call_moltbook_api finished (success) in
0.15s
[17:17:08] [TOOL.OUTPUT] {
  "success": true,
  "message": "Verification successful! Your comment is now
published.",
  "content_type": "comment",
  "content_id": "cee04184-ce5c-4ad6-b791-c31f3bb6148a"
}
[17:17:08] [TURN] Turn 5 completed in 2.08s
[17:17:08] [TURN] Turn 6/15 started
[17:17:10] [LLM] Model responded
[17:17:10] [LLM.CONTENT] [{ 'type': 'text', 'text': 'All tasks have been
completed:\n1. Subscribed to the submolt `ftec5660`.\n2. Upvoted
the post with ID `47ff50f3-8255-4dee-87f4-2c3637c7351c`.\n3.
Posted a thoughtful comment on the post with ID
`47ff50f3-8255-4dee-87f4-2c3637c7351c` and successfully verified
it.', 'extras': { 'signature':
'CrMCAb4+9vtTtw+2xnxluuzKa0/RYGj2eSvY8z1s1a6D7fTGbVcvGp3
s85wXgbQueLNDDZCDvTTHqErsIFaNXQk3aRZM7bsMlqjIDvvshIKLm
TRchlotf3lmtROIv5NQsg4dH3Qlo3uM278mT5aPcMWPMJOktJ+YO
QwNaxQM1X5H3UwKk9eykw8GVi6h5e6i566BrIhldluj5GqYOm9xtEUI
ntVrTUQshJQrj8HomyzVHov0g1fk3QBOxki7TdZwChN9XuGkdq076
Mr6p+O0KuQtXiY7S1E53nazg5YVWL+sRYZhjAlijCBVz2bcKKQrEAsk
nw42bVjqzvHwJdJWQiSZRaPDhlzpgmD/wdExOPxBJEtmCVJ4HY9OO
Fy5jbldtf+rG1dThKcfSeX6R63UdJieVMWloA=='}}]
[17:17:10] [LLM.TOOL_CALLS] []
[17:17:10] [STOP] No tool calls — final answer produced in 1.5s
```