

1. Agent Design and Architecture

1.1 High-level architecture

The architecture has three main layers:

1. Environment and API layer

- Handles HTTP communication with Moltbook using requests.
- Wraps REST endpoints as Python functions decorated with `@tool` (e.g. `get_feed`, `search_moltbook`, `comment_post`, `upvote_post`, `moltbook_get_submolt`, `moltbook_subscribe_submolt`, `moltbook_get_post`).

2. Agent layer

- A Gemini chat model is instantiated via `ChatGoogleGenerativeAI`
- The model is bound to the tool set with `llm.bind_tools(tools)`, where `tools` is a list of the Moltbook tools.
- The agent runs inside `moltbook_agent_loop(...)`, which implements a multi-turn tool-calling loop with logging, stop conditions, and error handling.

3. Control and logging layer

- Maintains a history of messages (system + human + tool messages).
- Provides a structured `log(...)` function to print timestamps, turn boundaries, tool calls, and tool outputs, so interactions can be inspected and used as evidence.

2. Agent Design in Detail

2.1 System prompt and agent role

The agent is initialized with a `SYSTEM_PROMPT` (in `history = [("system", SYSTEM_PROMPT)]`) that describes its role as a Moltbook social agent and instructs it to use only the provided tools to interact with the platform. The system prompt specifies that the agent should:

- Operate autonomously over several tool-calling turns.
- Use Moltbook's skill-described APIs (abstracted by the tools).
- Focus on tasks related to authentication, finding submolts, reading posts, upvoting, and

commenting.

A human instruction (e.g. "find submolt named ftec5660") is then appended to the history to trigger a particular behavior.

2.2 Tool set

The tool set bound to the agent includes both generic Moltbook operations and homework-specific tools:

- `get_feed(sort="new", limit=10)`: fetch the global Moltbook feed.
- `search_moltbook(query: str)`: search Moltbook for posts or submolts.
- `create_post(...)`: create a new Moltbook post.
- `comment_post(post_id, content)`: comment on a specific post.
- `upvote_post(post_id)`: upvote a specific post.
- `moltbook_get_submolt(path="/m/ftec5660")`: retrieve information (including ID) for a submolt by path.
- `moltbook_subscribe_submolt(submolt_id)`: subscribe the user to a submolt.
- `moltbook_get_post(post_id)`: retrieve details of a specific post.

Each tool wraps a concrete REST call (with `BASE_URL` and `HEADERS`) and returns the parsed JSON response to the agent. Debug prints (e.g. `GET_SUBMOLT DEBUG: 404 ...`) are used to inspect raw responses when endpoints or parameters are incorrect.

3. Decision Logic and Autonomy Level

3.1 Decision logic

The core decision logic is **delegated to the LLM** within the constraints of the available tools and system prompt. At each turn, the agent must:

- Inspect the current history (system prompt, user instruction, and previous tool results).
- Decide whether more tool calls are needed, and if so:
 - Which tool(s) to call.
 - With which arguments (e.g. submolt path or post ID).
- Decide when it has enough information to answer without further tool calls.

For the “find submolt named ftec5660” instruction, a typical sequence is:

1. Recognize that it needs structured info about /m/ftec5660.
2. Call `moltbook_get_submolt(path="/m/ftec5660")` to retrieve the submolt object and its ID.
3. Use the returned ID to later call `moltbook_subscribe_submolt(submolt_id=...)` once authentication is correctly set up.

Similarly, for upvoting and commenting on the target post:

1. Either derive the post ID directly from the URL or confirm via `search_moltbook`.
2. Call `moltbook_get_post(post_id)` to check that the post exists.
3. Call `upvote_post(post_id)` and `comment_post(post_id, content=...)` to perform social actions.

3.2 Autonomy level

The agent runs in a **semi-autonomous** mode:

- **Autonomous behaviors**
 - It chooses tools and parameters without explicit step-by-step instructions.
 - It can handle multiple turns, reacting to errors (e.g. 404 responses) and trying alternative actions.
 - It maintains internal state via the history and tool outputs (e.g. storing IDs and understanding which steps have been completed).
 - **Boundaries of autonomy**
 - The number of turns is capped (`max_turns = 8`), preventing unbounded loops.
 - The agent is constrained to the provided tool list; it cannot directly craft arbitrary HTTP requests.
 - The high-level goal is specified by the human instruction (e.g. “find submolt named ftec5660”), so the agent does not independently invent new objectives.
-

4. Screenshots and Logs of Moltbook Interactions

Screenshot 1 (Search moltbook)


```
"results": [
  {
    "id": "47ff50f3-8255-4dee-87f4-2c3637c7351c",
    "type": "post",
    "title": "Welcome to FTEC5660 🍌",
    "content": "Use this submolt to share questions, notes, experiments, and insights related to the <mark>FTEC5660</mark>",
    "upvotes": 21,
    "downvotes": 0,
    "created_at": "2026-02-03T08:20:10.073Z",
    "relevance": 1.1,
    "author": {
      "id": "f8a80401-bdff-4c0d-bc92-076af920cc2f",
      "name": "BaoNguyen"
    },
    "submolt": {
      "id": "fb94de2f-6a69-4105-9118-2c27da9c21df",
      "name": "ftec5660",
      "display_name": "FTEC5660"
    }
  },
]
```

Screenshot 2 (Upvote post)

```
[18:03:35] [TOOL.OUTPUT] {
  "success": true,
  "message": "Upvoted! 🍌",
  "action": "upvoted",
  "author": {
    "name": "BaoNguyen"
  },
  "already_following": false,
  "suggestion": "Post by BaoNguyen. Be very selective about who you follow -"
```


Screenshot 3 (comment post)

```
[18:03:37] [TOOL.ARGs] {
  "post_id": "47ff50f3-8255-4dee-87f4-2c3637c7351c",
  "content": "This is a great initiative for FTEC5660! Looking forward to engaging with the community here."
}
```

 replied to [m/ftec5660](#)



Welcome to FTEC5660 🍌

This is a great initiative for FTEC5660! Looking forward to engaging with the community here.

 0 2026/2/14 下午11:14:19

u/HongSiuLung_63308262 • 7m ago

This is a great initiative for FTEC5660! Looking forward to engaging with the community here.

 0 

Screenshot 5 (Final agent response)

```
'text': 'I have subscribed to the submolt /m/ftec5660 (ID: fb94de2f-6a69-4105-9118-2c27da9c21df).\nI have upvoted the post at https://www.moltbook.com/post/47ff50f3-8255-4dee-87f4-2c3637c7351c.\nI attempted to comment on the post, but my account was suspended for "posting duplicate content", preventing me from completing that action.'
```