## **1 | High-Level Agent Roles**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Discord Handle** | **Mission** | **Typical Skills / Calls** |
| 1 | **@Orchestrator** | Parse every post, choose specialist(s), merge replies, handle safety & rate-limits. | LangChain Router / Autogen Coordinator |
| 2 | **@Idea-Coach** | Structured brainstorming: market gaps, problem framing, “How-Might-We” prompts. | Creativity prompt library, trend APIs |
| 3 | **@Validation-Mentor** | TAM/SAM analysis, competitor scans, persona sketches, lean-canvas fill-ins. | SimilarWeb API, Crunchbase scrape, custom Python |
| 4 | **@Product-Manager** | Draft PRD, write user stories, map acceptance criteria; tags each story by complexity. | Story template bank, Markdown generator |
| 5 | **@MVP-Planner** | Filters stories to MVP, produces sprint backlog & Trello/Jira cards. | Vector search of past MVPs, Jira REST |
| 6 | **@Design-Guru** | Generates low-fi wireframes, color palettes, accessibility checks. | Figma API, DALL·E for quick mock screens |
| 7 | **@Prompt-Engineer** | Optimises prompts for Lovable.dev/Bolt.new so multiple features fit in one call. | Prompt compression heuristics |
| 8 | **@Tech-Architect** | Suggests stack, Supabase schema, auth flow, deployment plan. | Supabase CLI, Mermaid ERD diagrams |
| 9 | **@QA-Tester** | Produces test plans, edge-case checklists, and CI snippets. | Playwright template repo |
| 10 | **@Pitch-Coach** | Crafts slide deck outline, speaker notes, 90-sec elevator pitch. | Reveal.js template, rhetorical-device prompt set |
| ★ | **@Student-Profiler** (background) | Stores each student’s progress, knowledge gaps, agent usage. | Postgres / Supabase Edge |
| ☆ | **@KG-Archivist** *(optional)* | Logs chains-of-thought & tool outputs to a knowledge graph for audit/explainability. | Neo4j Aura / Memgraph |

## **2 | Interaction Flow (per Discord message)**

1. **Student** posts in #project-help: “Need ideas for hydration-tracking app features.”
2. **@Orchestrator** embeds message → routes to **@Idea-Coach**.
3. Idea-Coach returns three features; routes *internally* to **@Validation-Mentor** for quick feasibility scores.
4. Orchestrator merges & replies to Discord; **@Student-Profiler** logs the learner’s new concept tags.
5. If provenance research is enabled, **@KG-Archivist** stores the reasoning chain as triples.

## **3 | Memory Management**

|  |  |  |
| --- | --- | --- |
| **Layer** | **Purpose** | **Tech** |
| **Redis session cache** | Last few Discord turns per thread (fast). | Redis |
| **Vector store** | Slides, docs, past Q&As, exemplar PRDs. | Chroma + Instructor-XL embeddings |
| **Student DB** | Progress, mastery, deliverable links. | Supabase/Postgres |
| **(Optional) KG** | Agent thoughts & tool outputs for audit/eval. | Neo4j / Memgraph |

Agents call helper SDK functions: get\_context(thread\_id), write\_progress(student\_id, record), etc.

## **4 | Tools & Integrations**

* **Discord Bot** → FastAPI → Webhooks
* **LLMs** – vLLM cluster (DeepSeek-R 7B / Qwen-32B) with GPT-4 reserved for polishing/pitch generation
* **Embeddings** – Instructor-XL + E5-large
* **Design** – Figma REST, DALL·E image generation
* **Proj-Mgmt** – Jira/Trello API for card creation
* **DB / Auth** – Supabase templates suggested by Tech-Architect
* **Testing** – Playwright boilerplates dropped into repo by QA-Tester
* **Dashboards** – Superset on TimescaleDB for instructors

## **5 | Detailed Interaction Model (sequence diagram)**

Student ──▶ Orchestrator ──▶ Idea-Coach

│

├──▶ Validation-Mentor

│ ▲

│ └── scores + feature grades

▼

◀─────────────── combined suggestions

│

├──▶ Student-Profiler (async)

├──▶ KG-Archivist? (if logging ON)

└──▶ Discord reply (markdown + emoji)

* **Primary flow** (Student → Orchestrator → Idea-Coach → Validation → back)
* **Async side-effects** (Profiler update, optional KG logging)
* **Single return path** to the Student in Discord.

## **6 | Example Scenario – “Filter to MVP”**

1. Student uploads 15 user stories.
2. Student pings **@MVP-Planner** with “/mvp filter stories”.
3. Planner queries **@Product-Manager** for complexity tags, consults velocity heuristics, and outputs:
   1. “Auth, Profile, Water-log” as *must-have*
   2. “Community leaderboard” deferred.
4. Creates Trello cards via API and returns board link.
5. QA-Tester auto-generates a skeleton Playwright test for Auth + Profile.

## **7 | Key Design Tips**

1. **Keep handles memorable** – students type @Design-Guru not “Agent-6”.
2. **Slash-commands reduce token cost** – /mvp, /pitch, /wireframe.
3. **Structured JSON I/O** – enforce with Guardrails; Orchestrator rejects free-text that lacks required keys.
4. **Minimal cross-talk** – Orchestrator is the only hub; agents *never* DM each other in Discord (avoids spam).
5. **Adaptive responses** – Student-Profiler passes a *difficulty* tag (“novice / intermediate / advanced”) to every agent prompt.
6. **Model cost control** – open weights for drafting, GPT-4 only for final pitch deck or complex explanations.
7. **Pluggable KG** – add @KG-Archivist only if IRB or grading rubrics demand transparent reasoning.
8. **Continuous evaluation loop** – nightly script samples 20 threads, scores them with automated rubrics, triggers prompt adjustments.

### **3-Week Implementation Sprint**

|  |  |  |
| --- | --- | --- |
| **Week** | **Goal** | **Checks** |
| 1 | Discord bot + @Orchestrator skeleton; Redis cache | Bot responds “pong” and echoes role selection. |
| 2 | Add @Idea-Coach & @Product-Manager; seed vector store with Week 1-2 PDFs | Students can brainstorm & draft PRDs end-to-end. |
| 3 | Integrate Trello + Supabase; wire @MVP-Planner, @Tech-Architect | First cohort pilot (5 teams) completes MVP backlog creation. |