

AI Cost Analysis (Week 1)

Date: 2026-02-16
Project: CollabBoard MVP-1

1) Development and Testing Costs

1.1 Actual Development Spend (this sprint)

Category	Value	Notes
LLM API direct billing	\$0.00	No direct OpenAI/Anthropic runtime API billing wired into app code in Week 1.
Total API calls (runtime app AI endpoint)	0	billed production calls Endpoint deployed; usage during dev validation only.
Token consumption (runtime app)	0	billed tokens Current AI dispatcher is deterministic command parser, not paid model inference in prod path yet.
Other AI infra costs	\$0.00	No embeddings/vector DB usage in Week 1.

1.2 Engineering Tooling Note

- **Primary AI coding assistant**: Claude (Anthropic) via subscription
- **Workflow tooling**: Cursor IDE + MCP integrations
- **Production AI model**: MiniMax M2.5 (when LLM integration is needed)

1.3 MiniMax M2.5 Recommendation

- For production AI command processing, we recommend **MiniMax M2.5**:
- **Cost**: ~\$0.40 per 1M tokens (8x cheaper than GPT-4)
 - **Quality**: Excellent for structured outputs and function calling
 - **Speed**: Fast inference suitable for real-time commands
 - **Advantages**:
 - Dramatically lower production costs at scale
 - Strong performance on pattern recognition tasks
 - Good multilingual support if needed internationally

Why MiniMax over GPT-4/Claude:

- Cost efficiency matters for collaborative tools with frequent AI commands
- Command parsing is well-suited for fast, cost-effective models
- Reserve expensive models only for complex semantic planning

2) Production Cost Projections

2.1 Assumptions

- Average AI commands per user per session: 6
- Average sessions per user per month: 8
- Command mix: 80% simple, 20% complex
- Token assumptions (using MiniMax M2.5 pricing):
 - Simple command: 900 input + 300 output = 1,200 tokens
 - Complex command: 1,600 input + 1,200 output = 2,800 tokens
- Weighted average tokens per command: 1,520
- Tokens per user/month: 6 * 8 * 1,520 = 72,960
- **MiniMax M2.5 cost**: ~\$0.40 per 1M tokens
- Infra estimate includes Firebase Hosting + Firestore + RTDB + Cloud Functions

2.2 Monthly Projection Table (MiniMax M2.5)

User Scale	LLM Tokens / Month	LLM Cost / Month	Infra Cost / Month	Total / Month
100 users	7.296M	\$3	\$5	\$8
1,000 users	72.96M	\$29	\$25	\$54
10,000 users	729.6M	\$292	\$150	\$442
100,000 users	7.296B	\$2,918	\$750	\$3,668

2.3 Comparison: Premium Models (GPT-4/Claude)

User Scale	Premium LLM Cost / Month	MiniMax Savings
100 users	\$23	**\$20 (87% less)**
1,000 users	\$233	**\$204 (88% less)**
10,000 users	\$2,335	**\$2,043 (88% less)**
100,000 users	\$23,347	**\$20,429 (88% less)**

3) Sensitivity and Controls

3.1 Cost Drivers

- Commands per session
- Complex-command ratio
- Model choice / token price

3.2 Cost Controls

- **Route to MiniMax M2.5** for 95% of commands (fast, cheap, accurate)
- **Keep deterministic parser** for common patterns (zero token cost)
- **Cache board context** for repeated commands
- **Rate limits** per user to prevent abuse
- **Fallback to premium model** only for ambiguous/complex queries

3.3 Tiered Model Strategy

Command Type	Model	Rationale
"Create sticky note"	Deterministic	Zero cost, instant
"SWOT template"	MiniMax M2.5	Cheap, fast, structured
"Organize by sentiment"	MiniMax M2.5	Requires reasoning
"Create custom framework"	Premium (GPT-4)	Complex, rare

4) Current Recommendation

1. **Keep deterministic parser** for MVP commands (zero cost, <100ms)
2. **Adopt MiniMax M2.5** for semantic AI commands (88% cost savings)
3. **Implement tiered routing**: deterministic → MiniMax → premium
4. **Monitor first 2 weeks** of usage before committing to model mix
5. **Set budget alerts** at each tier to catch anomalies early

5) Summary

- **Current state**: \$0 spend (deterministic commands)
- **Projected with MiniMax**: \$8/month at 100 users, \$54/month at 1K users
- **Versus premium models**: 88% cost savings with minimal quality tradeoff
- **Recommendation**: Start with MiniMax M2.5, upgrade only if quality issues emerge