Musashi MVP

Product Requirements Document

Prediction Market Platform

Version 1.0 | January 8, 2026

Product	Musashi - Prediction Market Platform (Kalshi clone)
Target Markets	Hong Kong, Singapore, China (pending regulatory review)
Funding	50M CNY secured
Team	CEO (Michael), CMO (Han), Quant Dev (■■)
Model	Peer-to-peer binary outcome contracts (YES/NO)

Table of Contents

1. Executive Summary

Goal: Build a minimum viable product to test with 10-15 users using play money to validate core prediction market mechanics before scaling to real money.

Core Model: Users trade binary outcome contracts (YES/NO) on future events. The platform operates as a peer-to-peer exchange where users trade against each other, not against the house.

2. Product Overview

2.1 What is a Prediction Market?

Users buy/sell shares that pay out based on whether a future event occurs:

• Each market has YES and NO shares

• If event happens: YES = ¥1, NO = ¥0

• If event doesn't happen: YES = ¥0, NO = ¥1

• Prices range from ¥0.01 to ¥0.99, representing probability

Example:

• Market: 'Will Bitcoin hit $100k by Jan 15?'

• YES trading at 65¢ means market thinks 65% chance

• User buys 100 YES shares for ¥65

• If Bitcoin hits $100k: User receives ¥100 (profit: ¥35)

• If Bitcoin doesn't: User receives ¥0 (loss: ¥65)

2.2 Key Differences from Traditional Trading

NOT:

• Stock/futures/options trading

• Aggregating existing exchanges

• Quantitative trading strategies platform

IS:

• Binary outcome betting on events

• Self-contained order books per event

• Peer-to-peer exchange (users trade against each other)

• Similar to Betfair in architecture, Kalshi in business model

3. MVP Scope

3.1 Core Features (MUST HAVE)

User-Facing Features:

1. User registration and authentication
2. Play money system (¥10,000 starting balance)
3. Browse active markets
4. View market details (order book, price chart, event description)
5. Place market orders (buy/sell YES or NO)
6. View user positions and P&L;
7. Real-time price updates
8. Settlement notifications when markets resolve

Admin Features:

1. Create new markets
2. Resolve markets (set outcome to YES or NO)
3. Monitor system health
4. View all user positions

System Features:

1. Order matching engine (separate order book per market)
2. Market maker functionality (manual or simple bot)
3. Real-time WebSocket updates
4. Settlement system

3.2 Out of Scope for MVP

• Real money deposits/withdrawals

• Advanced order types (limit orders, stop-loss)

• Mobile app (web-responsive only)

- Complex KYC/AML
- Sophisticated market maker algorithms
- Social features (chat, following users)
- Referral program
- API access for traders

4. Technical Architecture

4.1 System Components

The system consists of three main layers:

• **Frontend (React):** Market pages, trading interface, user dashboard, admin panel

• **Backend (Node.js/Python):** Order matching engine, market management, user authentication, market maker bot

• **Database (PostgreSQL):** Users, markets, orders, trades, positions, settlements

4.2 Key Database Tables

• **users:** id, email, password_hash, phone, created_at, is_admin

• **wallets:** user_id, balance, locked_balance, updated_at

• **markets:** id, question, description, resolution_criteria, end_date, status, outcome

• **orders:** id, user_id, market_id, side, type, price, quantity, status

• **trades:** id, market_id, buyer_id, seller_id, side, price, quantity, timestamp

• **positions:** user_id, market_id, side, quantity, avg_price, status

• **settlements:** market_id, user_id, side, quantity, payout, settled_at

• **transactions:** user_id, type, amount, status, reference_id, timestamp

5. Core User Flows

5.1 User Registration Flow

1. User lands on homepage and clicks 'Sign Up'

2. Enters email, password, phone, age confirmation (18+)

3. Backend validates and creates account

4. System auto-credits ¥10,000 play money

5. User redirects to dashboard

5.2 Browse Markets Flow

Users see a list of active markets with current YES/NO prices, trading volume, and number of traders. Clicking on a market shows detailed information.

5.3 Place Order Flow (CRITICAL)

1. User selects YES or NO side

2. Chooses order type (Market Order for MVP)

3. Enters quantity of shares

4. Reviews estimated cost

5. Clicks 'Buy' or 'Sell' button

6. **Order matching engine** matches against existing orders

7. User sees confirmation and updated position

5.4 Order Matching Engine Logic

The matching engine is the core of the platform:

• Validates user has sufficient balance

• Fetches opposite side orders from the order book

• Matches orders based on price-time priority

• Creates trade records for each fill

• Updates user positions and balances atomically

• Broadcasts real-time updates via WebSocket

• Handles partial fills and remaining quantities

5.5 View Positions Flow

Users can view their dashboard showing all open positions with real-time P&L; calculations based on current market prices.

5.6 Market Resolution Flow (Admin)

1. Admin reviews markets past their end date

2. Verifies actual outcome against resolution criteria

3. Clicks 'Resolve as YES' or 'Resolve as NO'

4. **Settlement engine** processes all positions:

• Winners (correct side): Receive ¥1 per share

• Losers (wrong side): Receive ¥0 per share

5. System credits wallets and closes positions

6. Users receive settlement notifications

6. Market Maker Strategy

6.1 Purpose

Provide initial liquidity so retail users always have someone to trade against. This prevents the 'dead market' feel that kills trading platforms.

6.2 Simple Bot Approach

A basic market maker bot continuously:

• Monitors the order book for each market

• Posts orders on both YES and NO sides

• Maintains target spread (e.g., 5¢)

• Adjusts prices based on fair value (midpoint)

• Manages inventory limits to avoid excessive risk

• Cancels stale orders and re-posts fresh ones

6.3 Manual Market Making (MVP Fallback)

If bot implementation is delayed, admins can manually provide liquidity by placing initial orders on both sides of each market, creating a reasonable spread (e.g., 48¢ / 52¢).

7. API Endpoints Reference

Authentication

• POST /api/auth/register - Create new user account

• POST /api/auth/login - User login

• POST /api/auth/logout - User logout

• GET /api/auth/me - Get current user info

Markets

• GET /api/markets - List all markets

• GET /api/markets/:id - Get market details

• POST /api/markets - Create market (admin only)

• POST /api/markets/:id/resolve - Resolve market (admin only)

• GET /api/markets/:id/orderbook - Get current order book

• GET /api/markets/:id/trades - Get recent trades

• GET /api/markets/:id/chart - Get price history

Orders

• POST /api/orders - Place new order

• GET /api/orders - Get user's orders

• DELETE /api/orders/:id - Cancel order

User

• GET /api/user/profile - Get user profile

• GET /api/user/balance - Get wallet balance

• GET /api/user/positions - Get open positions

• GET /api/user/settlements - Get settled positions

• GET /api/user/transactions - Get transaction history

8. Testing Plan

8.1 MVP Testing Phases

Phase 1: Internal Testing (Week 1)

• 3-5 team members test basic flows

• Focus on registration, markets, order placement

• Identify and fix critical bugs

Phase 2: Closed Beta (Week 2-3)

• 10-15 external users

• Test with simple events (weather, crypto prices)

• Monitor trades per day, average spread, user feedback

Phase 3: Evaluation (Week 4)

• Analyze user retention and engagement

• Measure liquidity quality (spread sizes)

• Gather detailed feedback on UX and market selection

8.2 Key Metrics to Track

• **Engagement:** Daily Active Users, Trades per user, Time on platform

• **Liquidity:** Average spread, Time to fill orders, Market maker profitability

• **Technical:** Order matching latency, WebSocket stability, Database performance

• **User Feedback:** NPS score, Feature requests, Pain points

9. Go-to-Market Strategy (Post-MVP)

9.1 Initial User Acquisition Targets

• **Crypto traders:** They already understand prediction markets

• **Finance professionals:** Familiar with derivatives and risk

• **Sports bettors:** Understand probability and odds

9.2 Acquisition Channels

• **Crypto Twitter/X:** Target Chinese crypto influencers, share interesting outcomes

• **Reddit/Forums:** Finance and crypto communities, post predictions

• **WeChat Groups:** Trading groups, crypto communities, university clubs

• **Referral Program:** Both referrer and referee get bonus credits (later phase)

9.3 Market Selection Strategy

Start with HIGH INTEREST + CLEAR RESOLUTION:

Good Markets:

• Crypto prices ('Will Bitcoin hit $X by date Y?')

• Economic data releases ('Will China GDP exceed X%?')

• Tech events ('Will Apple announce new iPhone?')

• Sports outcomes (if legally permissible)

Avoid Initially:

• Political events (high regulatory risk)

• Ambiguous outcomes (hard to resolve)

• Long-term predictions (low engagement)

10. Budget Allocation (50M CNY)

Category	Amount	Breakdown
Development	20M CNY	Engineering team (10M), Infrastructure (3M), Security (2M), Contingency (5M)
Marketing	15M CNY	User acquisition (10M), Liquidity subsidies (3M), PR/branding (2M)
Legal/Compliance	10M CNY	Regulatory licenses (5M), Legal consultation (3M), Compliance (2M)
Operations	5M CNY	Customer support (2M), Admin/overhead (3M)

11. Development Timeline

Phase	Duration	Key Activities
Development	Weeks 1-2	Database setup, Backend API, Order matching engine, Basic frontend
Internal Testing	Week 3	Team testing, Bug fixes, Deploy to staging
Closed Beta	Week 4	Invite 10-15 users, Create test markets, Monitor closely
Iteration	Weeks 5-6	Fix issues from beta, Improve UX, Add missing features
Prepare Launch	Weeks 7-8	Legal review, Payment integration, KYC system, Security audit
Launch	Week 9+	Start with small markets, Gradual user growth, Marketing ramp-up

12. Critical Success Factors

1. Liquidity is Everything - Users leave if they can't trade. Market making is absolutely critical.

2. Clear Resolution - If users don't trust outcomes, the platform dies. Resolution criteria must be crystal clear.

3. Fast Execution - Orders must fill instantly. Latency kills trading platforms.

4. Regulatory Compliance - One wrong move = platform shutdown. Michael must nail this.

5. User Trust - Play money testing builds trust before asking for real money.

6. Market Selection - Pick events people care about AND can verify easily.

13. Risks & Mitigation

Risk	Impact	Mitigation Strategy
Regulatory shutdown	HIGH	Start in Singapore with proper licenses
Low liquidity	HIGH	Subsidize market making, start with small markets
Resolution disputes	MEDIUM	Clear criteria, use multiple data sources
Technical failures	HIGH	Extensive testing, monitoring, backups
High CAC	MEDIUM	Start with organic growth, referrals
Competitor copies	LOW	Move fast, build brand loyalty
Wash trading	MEDIUM	Fraud detection, trading limits

14. Next Steps

1. Share PRD with quant dev (■■) - Ensure complete understanding of architecture

2. Set up development environment - Database, backend framework, frontend scaffolding

3. Create technical specifications - Detailed API contracts, data models, schemas

4. Build MVP iteratively - Start with order matching engine, then add features

5. Test internally - Team uses platform daily to identify issues

6. Launch closed beta - Invite 10-15 external users for real testing

7. Iterate based on feedback - Fix bugs, improve UX, add requested features

8. Prepare for real money - Legal compliance, payment integration, security hardening

Appendix A: Glossary

Order Book: List of all open buy and sell orders for a specific market

Spread: Difference between best bid (buy) and best ask (sell) price

Liquidity: How easily users can trade without significantly moving the price

Market Maker: Entity that provides liquidity by continuously posting orders on both sides

Settlement: Process of paying out winners after an event resolves

Position: User's current holdings (shares) in a specific market

P&L; (Profit & Loss): Gain or loss on a position, either realized or unrealized

Binary Outcome: Event with only two possible results (YES or NO)

Resolution Criteria: Specific rules for determining the outcome of an event

Appendix B: Example Markets for MVP Testing

1. 'Will Bitcoin close above $100,000 on January 15, 2026?'

Clear data source (Coinbase), daily updates, high interest topic

2. 'Will it rain in Beijing on January 10, 2026?'

Clear resolution (weather data), short timeframe, local interest

3. 'Will AAPL stock close above $200 on January 9, 2026?'

Clear data source (NASDAQ), finance-oriented users, daily market

4. 'Will Shanghai Composite close higher than Shenzhen on Jan 9?'

Relative prediction (interesting twist), local relevance, clear resolution

5. 'Will Elon Musk tweet more than 10 times on January 9, 2026?'

Fun/entertainment value, easy to verify (Twitter/X), viral potential

End of Document

Version 1.0 | January 8, 2026

Musashi Prediction Market Platform

For questions or clarifications, start with Section 4 (Technical Architecture) and Section 5 (User Flows) for development work.