

암호화폐 자동매매 시스템 PRD (듀얼 버전)

Product Requirements Document - 최종 완전판

문서 정보

항목	내용
문서명	암호화폐 자동매매 시스템 PRD (듀얼 버전)
버전	v4.0 (최종 완전판)
작성일	2025년 8월 31일
대상 거래소	바이낸스 선물, 바이비트 선물/무기한
거래 타입	선물 거래 전용
배포 형태	EXE 버전 + 웹 대시보드 버전

1. 프로젝트 개요

1.1 프로젝트 목표

바이낸스 선물과 바이비트 선물/무기한 계약을 대상으로 하는 고도로 정교한 자동매매 시스템을 듀얼 버전으로 개발

1.2 듀얼 버전 전략



EXE 버전: 개인 PC에서 완전한 기능 제공



웹 대시보드 버전: 서버에서 24시간 원격 제어

1.3 핵심 특징

- 선물 거래 전용: 레버리지 활용한 고수익 추구
 - 멀티 거래소: 바이낸스 선물 + 바이비트 선물/무기한
 - 5가지 진입 조건: 정교한 진입 타이밍 포착
 - 4가지 청산 방식: 체계적인 리스크 관리
 - 12단계 시스템: 익절/손절 세밀 제어
 - 시간 제어: 요일별/시간별 정교한 운용 관리
 - 듀얼 인터페이스: 데스크톱 + 웹 동시 지원
-

2. 시스템 아키텍처

2.1 전체 아키텍처



2.2 공통 코어 모듈

모든 비즈니스 로직을 공통 모듈로 구현하여 두 버전에서 동일하게 사용

```
core/
├── trading_engine.py      # 거래 엔진 (진입/청산 로직)
├── risk_manager.py        # 리스크 관리 시스템
├── api_connector.py       # 거래소 API 통합 관리
├── config_manager.py      # 설정 관리 (암호화 포함)
├── time_controller.py     # 시간 제어 시스템
├── security_module.py     # 보안 모듈
├── data_processor.py      # 데이터 처리 및 분석
├── notification.py       # 알림 시스템
└── logger.py             # 로깅 시스템
```

2.3 버전별 특화 모듈

EXE 버전 (desktop/)

```
desktop/
├── main_gui.py            # PyQt5 메인 윈도우
├── tabs/
│   ├── entry_tab.py      # 진입 설정 탭
│   ├── exit_tab.py       # 청산 설정 탭
│   └── settings_tab.py   # 사용자 설정 탭
├── widgets/
│   ├── chart_widget.py   # 실시간 차트
│   ├── position_widget.py # 포지션 위젯
│   └── status_widget.py  # 상태 표시 위젯
└── utils/
    ├── gui_utils.py      # GUI 유틸리티
    └── theme_manager.py   # 테마 관리
```

웹 대시보드 버전 (web/)

```
web/
├── app.py                 # Flask 메인 애플리케이션
├── routes/
│   ├── api.py            # REST API 엔드포인트
│   ├── websocket.py      # WebSocket 핸들러
│   └── auth.py           # 인증 관리
├── templates/
│   ├── dashboard.html    # 메인 대시보드
│   ├── settings.html     # 설정 페이지
│   └── login.html        # 로그인 페이지
├── static/
│   ├── css/              # 스타일시트
│   ├── js/               # JavaScript
│   └── img/              # 이미지 자원
└── utils/
    ├── web_utils.py      # 웹 유틸리티
    └── session_manager.py # 세션 관리
```

3. EXE 버전 상세 설계

3.1 기술 스택

GUI Framework: PyQt5
패키징: PyInstaller
암호화: cryptography (Fernet)
차트: pyqtgraph 또는 matplotlib
실시간 통신: websocket-client
데이터 처리: pandas, numpy

3.2 UI 설계 (기존 설계 그대로 적용)

3개 탭 구조: 진입 설정 / 청산 설정 / 사용자 설정

메인 상태 표시줄:

연결: ☒ 바이낸스 선물 | 포지션: 매수 0.1 BTC (+1.5%) | 시간: 14:30:25
[🚨 긴급 포지션 청산] [⏸ 자동매매 일시정지] [▶ 자동매매 재시작]

3.3 EXE 버전 특징

- 완전한 오프라인 실행: 인터넷 연결 시에만 거래
- 로컬 설정 저장: 암호화된 로컬 파일
- 실시간 차트: 내장 차트 위젯
- 단일 실행 파일: .exe 파일 하나로 완전 실행
- Windows 최적화: Windows 10/11 완벽 지원


4. 웹 대시보드 버전 상세 설계

4.1 기술 스택

Backend: Flask + Flask-SocketIO
Frontend: HTML5 + CSS3 + JavaScript (ES6+)
실시간 통신: WebSocket (Socket.IO)
차트: Chart.js 또는 TradingView 위젯
UI Framework: Bootstrap 5 (반응형)
인증: Flask-Login + JWT
데이터베이스: SQLite (설정 저장용)

4.2 웹 대시보드 UI 설계

4.2.1 로그인 페이지


 자동매매 시스템 로그인

사용자명: [

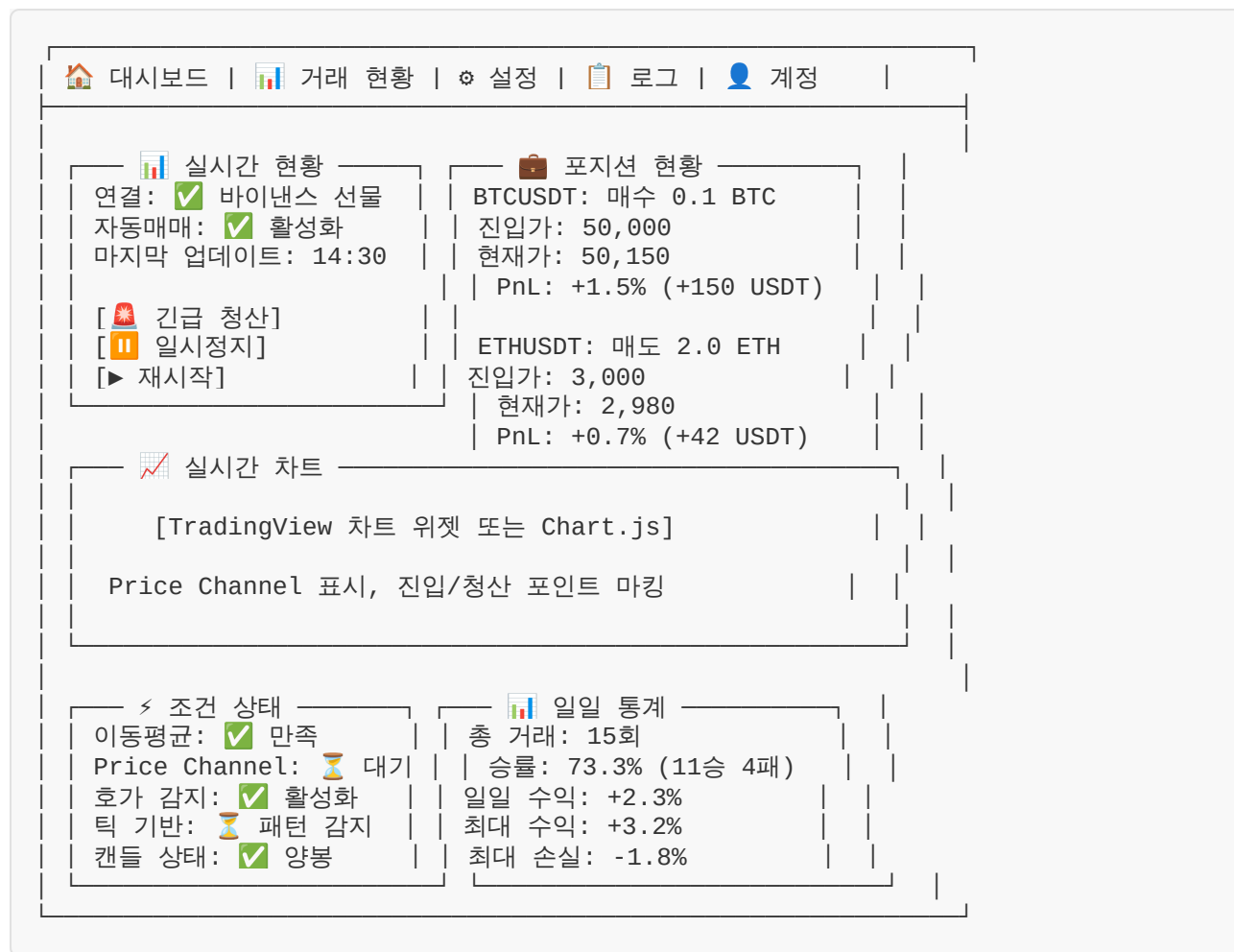
비밀번호: [

☐ 로그인 상태 유지 (7일)

[로그인] [비밀번호 변경]

 보안을 위해 30분 비활성 시 자동 로그아웃됩니다

4.2.2 메인 대시보드



4.2.3 거래 설정 페이지

거래 설정

🎯 진입 조건 설정

이동평균선 조건: ☒ 활성화

└ 시가 비교: [시가 < **이평선** → 매수 진입 ▼]

└ 현재가 비교: [선택 안함 ▼]

Price Channel: ☒ 활성화

└ 상단 돌파: [매수 진입 ▼] 기간: [20]일

└ 하단 돌파: [매도 진입 ▼]

✂ 호가 감지: ☒ 활성화

└ 상승 틱: ☒ [3]틱 → 매수

└ 하락 틱: ☒ [2]틱 → 매도

틱 기반 추가: ☒ 활성화

└ [5]틱 상승 후 [2]틱 하락 시 [30]% 추가 진입

캔들 상태: ☒ 활성화

└ 양봉 시: [매수 진입 ▼]

└ 음봉 시: [매도 진입 ▼]

[설정 저장] [초기화]

🛡 청산 조건 설정

PCS 청산: ☒ 활성화

└ 활성화 단계: ☒1단 ☒2단 ☐3단 ☐4단 ...

└ PCS 1단: [1STEP (즉시 100% 청산) ▼]

└ PCS 2단: [2STEP (50% + 50% 분할) ▼]

PC 트레이일링: ☒ 활성화

└ 매수 시: ☒ 하단선 하락 → 청산

└ 매도 시: ☒ 상단선 상승 → 청산

└ 손실중 청산: [손실중에만 청산 ▼]

✂ 호가 청산: ☒ 활성화

└ 매수 포지션: [5]틱 하락 시 청산

└ 매도 포지션: [5]틱 상승 시 청산

PC 본질: ☒ 활성화

└ 1단계: ☒ PC선 돌파 확인

└ 2단계: ☒ 진입가 복귀 시 청산

[설정 저장] [초기화]

4.2.4 시스템 설정 페이지

⚙ 시스템 설정

🏠 거래소 설정

거래소: [바이낸스 선물 ▼] 계약: [BTCUSDT ▼]
레버리지: [10]배 포지션: [단방향 ▼] 마진: [격리 ▼]

API 설정:
API Key: [*****] [변경]
Secret: [*****] [변경]
연결 상태: ☒ 연결됨 권한: 거래 가능

🕒 시간 제어

📅 요일별 가동시간:

월요일: ☒ 사용함
1차: [12:00] ~ [18:00] 2차: [00:00] ~ [12:00]
상태: ☒ 가동중 (1차 시간대)

화요일: ☐ 사용안함
1차: [09:00] ~ [12:00] 2차: [13:00] ~ [18:00]
상태: ☒ 비가동

📅 요일별 청산시간:
월요일: ☒ [12:00] 청산 화요일: ☐ [18:00] 청산

[전체 복사] [프리셋] [저장]

🛡 리스크 관리

포지션 크기: [1000] USDT | 최대 포지션: [3]개
레버리지 노출: [10000] USDT (10배)

익절/손절 12단계:
☒ 익절1: [+2.0]% ☒ 손절1: [-1.0]%
☒ 익절2: [+4.0]% ☒ 손절2: [-2.0]%
☐ 익절3: [+6.0]% ☐ 손절3: [-3.0]%
... (12단계까지)

일일 손실 한도: [-5.0]%
초기화 로직: ☒ 익절 후 손절 시 초기화

4.3 웹 대시보드 특징

- **반응형 디자인:** 모바일/태블릿/데스크톱 모두 지원
- **실시간 업데이트:** WebSocket으로 실시간 데이터 동기화
- **원격 접속:** 어디서든 웹 브라우저로 접속 가능
- **24시간 운영:** 서버에서 지속적 실행

- **다중 세션:** 여러 기기에서 동시 접속 가능
- **클라우드 설정:** 설정 자동 동기화

4.4 REST API 설계

4.4.1 인증 API

```
POST /api/auth/login          # 로그인
POST /api/auth/logout        # 로그아웃
POST /api/auth/change-password # 비밀번호 변경
GET  /api/auth/status         # 인증 상태 확인
```

4.4.2 거래 API

```
GET  /api/trading/status      # 거래 상태 조회
POST /api/trading/start       # 자동매매 시작
POST /api/trading/stop        # 자동매매 중지
POST /api/trading/emergency-close # 긴급 청산
```

4.4.3 설정 API

```
GET  /api/config/entry        # 진입 설정 조회
PUT  /api/config/entry        # 진입 설정 저장
GET  /api/config/exit         # 청산 설정 조회
PUT  /api/config/exit         # 청산 설정 저장
GET  /api/config/system       # 시스템 설정 조회
PUT  /api/config/system       # 시스템 설정 저장
```

4.4.4 모니터링 API

```
GET  /api/monitor/positions   # 포지션 현황
GET  /api/monitor/trades      # 거래 이력
GET  /api/monitor/stats       # 통계 데이터
GET  /api/monitor/logs        # 시스템 로그
```

4.5 WebSocket 이벤트

```
// 클라이언트 → 서버
socket.emit('subscribe_positions'); // 포지션 구독
socket.emit('subscribe_market_data'); // 시장 데이터 구독
socket.emit('emergency_close'); // 긴급 청산

// 서버 → 클라이언트
socket.on('position_update', data); // 포지션 업데이트
socket.on('market_data', data); // 시장 데이터
socket.on('trade_signal', data); // 거래 신호
socket.on('system_alert', data); // 시스템 알림
```



5. 거래소 지원 (기존과 동일)

5.1 바이낸스 선물 (Binance Futures)

지원 계약: BTCUSDT, ETHUSDT 등 USDT 마진 선물
레버리지: 1~125배 (종목별 상이)
포지션 모드: 단방향/양방향 선택
마진 모드: 격리/교차 마진 선택

5.2 바이비트 선물/무기한 (Bybit)

지원 계약:
- 무기한 계약: BTCUSDT, ETHUSDT 등
- 선물 계약: 분기별 만료 계약
레버리지: 1~100배 (종목별 상이)
포지션 모드: 단방향/양방향 선택
마진 모드: 격리/교차 마진 선택



6. 거래 로직 (기존과 동일)

6.1 진입 조건 (5가지)

1. 이동평균선 조건 (8가지 선택)
2. Price Channel 조건 (상단/하단 돌파)
3. 호가 감지 진입 (0호가 즉시 진입 포함)
4. 틱 기반 추가 진입 (패턴 감지)

5. 캔들 상태 조건 (현재 봉 기준)

6.2 청산 조건 (4가지)

1. PCS 청산 (1단~12단, 1STEP/2STEP)
2. PC 트레이딩 청산 (PCT 손실중 청산)
3. 호가 청산 (틱 기반)
4. PC 본절 청산 (2단계 시스템)

6.3 시간 제어 시스템

- 요일별 가동시간 (2개 시간대)
 - 요일별 청산시간 (강제 청산)
-



7. 보안 시스템

7.1 공통 보안 기능

- Fernet 암호화: 모든 설정 파일 암호화
- SHA256 해시: 비밀번호 안전한 저장
- API 키 암호화: 메모리에서 자동 삭제

7.2 EXE 버전 보안

- 로컬 인증: 프로그램 시작 시 비밀번호 입력
- 자동 로그아웃: 30분 비활성 시 자동 종료
- 로그인 시도 제한: 3회 실패 시 프로그램 종료

7.3 웹 대시보드 보안

- JWT 토큰: 안전한 세션 관리
- HTTPS 강제: SSL/TLS 암호화 통신
- CSRF 보호: Cross-Site Request Forgery 방지

- **Rate Limiting:** API 호출 빈도 제한
 - **IP 화이트리스트:** 허용된 IP에서만 접속 가능
-

8. 개발 계획

8.1 개발 단계 (총 10주)

1단계 (1-2주): 공통 코어 모듈 - 거래 엔진 구현 (5가지 진입 + 4가지 청산) - API 연동 (바이낸스 선물, 바이비트) - 보안 모듈 (암호화, 인증) - 설정 관리 시스템

2단계 (3-5주): EXE 버전 개발 - PyQt5 GUI 구현 (3개 탭) - 실시간 차트 위젯 - 로컬 설정 관리 - PyInstaller 패키징

3단계 (6-8주): 웹 대시보드 개발 - Flask 백엔드 구현 - 반응형 웹 프론트엔드 - WebSocket 실시간 통신 - REST API 구현

4단계 (9-10주): 통합 테스트 및 배포 - 두 버전 통합 테스트 - 성능 최적화 - 보안 점검 - 사용자 매뉴얼 작성

8.2 개발 우선순위

높음 (필수) - 공통 코어 모듈 (거래 로직) - EXE 버전 기본 기능 - 웹 대시보드 기본 기능 - 보안 시스템

중간 (중요) - 고급 차트 기능 - 알림 시스템 - 통계 및 분석 - 모바일 최적화

낮음 (선택) - 테마 시스템 - 다국어 지원 - 고급 백테스팅 - 플러그인 시스템

8.3 테스트 전략

단위 테스트 - 각 진입/청산 조건별 개별 테스트 - API 연동 테스트 - 보안 모듈 테스트

통합 테스트 - EXE 버전 전체 시스템 테스트 - 웹 대시보드 전체 시스템 테스트 - 두 버전 간 설정 동기화 테스트

실거래 테스트 - 테스트넷 환경에서 실제 거래 테스트 - 소액 실거래 테스트 - 스트레스 테스트 (고빈도 거래)



9. 시스템 요구사항

9.1 EXE 버전 요구사항

OS: Windows 10/11 (64bit)
RAM: 최소 4GB (권장 8GB)
Storage: 최소 200MB
Network: 안정적인 인터넷 연결
Display: 1920x1080 이상 권장

9.2 웹 대시보드 서버 요구사항

OS: Ubuntu 20.04+ / CentOS 8+ / Windows Server
RAM: 최소 4GB (권장 8GB)
Storage: 최소 1GB
Network: 안정적인 인터넷 연결 (업로드 속도 중요)
Python: 3.8+ (권장 3.9)

9.3 클라이언트 요구사항 (웹 대시보드)

브라우저: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+
JavaScript: ES6+ 지원
WebSocket: 지원 필수
해상도: 모바일 360px ~ 데스크톱 1920px+



10. 성능 목표

10.1 응답 시간

- API 응답: 평균 100ms 이하
- UI 반응: 50ms 이하
- WebSocket 지연: 10ms 이하
- 차트 업데이트: 실시간 (1초 이내)

10.2 리소스 사용량

- EXE 버전 메모리: 200MB 이하

- 웹 서버 메모리: 500MB 이하
- CPU 사용량: 평균 5% 이하
- 네트워크 대역폭: 1Mbps 이하

10.3 안정성

- 업타임: 99.9% 이상
 - 오류 발생률: 0.1% 이하
 - 자동 복구: 네트워크 장애 시 자동 재연결
 - 데이터 무결성: 100% 보장
-

11. 리스크 관리

11.1 기술적 리스크

API 변경 리스크 - 대응: 버전 관리, 호환성 레이어 - 모니터링: 거래소 공지사항 자동 확인

서버 장애 리스크 - 대응: 자동 재시작, 헬스 체크 - 백업: 다중 서버 구성 (선택사항)

11.2 보안 리스크

웹 보안 위협 - 대응: HTTPS, CSRF 보호, Rate Limiting - 모니터링: 비정상 접속 패턴 감지

API 키 탈취 - 대응: 암호화 저장, 권한 최소화 - 제한: IP 화이트리스트, 거래량 제한

11.3 거래 리스크

네트워크 지연 - 대응: 타임아웃 설정, 재시도 로직 - 모니터링: 지연 시간 실시간 추적

급격한 시장 변동 - 대응: 긴급 청산 버튼, 자동 손절 - 제한: 최대 포지션 크기, 레버리지 제한



12. 모바일 지원

12.1 반응형 웹 디자인

```
/* 모바일 (360px ~ 768px) */
@media (max-width: 768px) {
  .dashboard-grid { grid-template-columns: 1fr; }
  .chart-container { height: 300px; }
  .control-buttons { flex-direction: column; }
}

/* 태블릿 (768px ~ 1024px) */
@media (min-width: 768px) and (max-width: 1024px) {
  .dashboard-grid { grid-template-columns: 1fr 1fr; }
  .chart-container { height: 400px; }
}

/* 데스크톱 (1024px+) */
@media (min-width: 1024px) {
  .dashboard-grid { grid-template-columns: 1fr 1fr 1fr; }
  .chart-container { height: 500px; }
}
```

12.2 모바일 최적화 기능

- 터치 친화적 UI: 버튼 크기 44px 이상
- 스와이프 제스처: 차트 확대/축소, 페이지 전환
- 오프라인 모드: 네트워크 끊김 시 기본 기능 유지
- 푸시 알림: 중요한 거래 신호 알림 (PWA)



13. 알림 시스템

13.1 알림 채널

EXE 버전 - 시스템 트레이 알림 - 소리 알림 (사용자 정의 가능) - 이메일 알림

웹 대시보드 - 브라우저 푸시 알림 - 이메일 알림 - 텔레그램 봇 알림 (선택사항) - SMS 알림 (선택사항)

13.2 알림 유형

- 정보: 거래 시작/중지, 설정 변경
- 경고: 네트워크 지연, API 오류
- 위험: 긴급 청산, 시스템 오류
- 성공: 진입/청산 완료, 수익 달성

13.3 알림 설정

- 알림 레벨: 전체/중요/위험만
- 시간 제한: 거래 시간에만 알림
- 빈도 제한: 동일 알림 1분 간격
- 개인화: 사용자별 알림 선호도



14. 분석 및 리포팅

14.1 실시간 분석

- 포지션 분석: 현재 수익/손실, 보유 시간
- 성과 분석: 일/주/월별 수익률
- 위험 분석: 최대 낙폭, 샤프 비율
- 거래 패턴: 승률, 평균 수익/손실

14.2 리포트 생성

일일 리포트



2025-08-31 거래 리포트

총 거래: 15회
승률: 73.3% (11승 4패)
일일 수익률: +2.3%
최대 수익: +3.2%
최대 손실: -1.8%
평균 보유시간: 45분

주간/월간 리포트 - PDF 형태로 자동 생성 - 이메일 자동 발송 - 상세 거래 이력 포함 - 성과 차트 및 분석

14.3 백테스팅 (선택사항)

- 과거 데이터로 전략 검증
 - 다양한 시장 상황 시뮬레이션
 - 최적 파라미터 찾기
 - 위험 시나리오 분석
-



15. 유지보수 및 업데이트

15.1 자동 업데이트

EXE 버전 - 시작 시 업데이트 확인 - 백그라운드 다운로드 - 사용자 확인 후 설치

웹 대시보드 - 서버 측 자동 배포 - 무중단 업데이트 - 롤백 기능

15.2 로깅 시스템

로그 레벨

DEBUG: 상세한 디버깅 정보
INFO: 일반적인 정보 (거래 실행 등)
WARN: 경고 상황 (지연, 재시도 등)
ERROR: 오류 상황 (API 실패 등)
FATAL: 치명적 오류 (시스템 종료)

로그 관리 - 일별 로그 파일 분리 - 자동 압축 및 보관 (30일) - 원격 로그 수집 (웹 버전) - 로그 분석 도구

15.3 모니터링





- 시스템 상태: CPU, 메모리, 네트워크
 - 거래 상태: API 응답시간, 오류율
 - 사용자 활동: 접속 현황, 기능 사용률
 - 알림 발송: 임계치 초과 시 자동 알림
-







16. 결론

본 PRD는 바이낸스 선물과 바이비트 선물/무기한 계약을 대상으로 하는 듀얼 버전 자동매매 시스템의 완전한 개발 명세서입니다.

16.1 핵심 혁신점

-  **듀얼 버전:** EXE + 웹 대시보드로 최고의 유연성
-  **공통 코어:** 동일한 거래 로직으로 일관성 보장
-  **완전한 원격 제어:** 24시간 서버 운영 + 웹 접속
-  **모바일 지원:** 언제 어디서나 모니터링 가능

16.2 경쟁 우위

-  **정교함:** 5가지 진입 + 4가지 청산 + 12단계 리스크 관리
-  **유연성:** 개인 PC 실행 + 서버 24시간 운영 선택 가능
-  **안전성:** 강화된 보안 + 긴급 청산 + 다중 안전장치
-  **확장성:** 모듈화 설계로 기능 추가 용이

16.3 개발 준비도

100% 개발 준비 완료 - 이 PRD를 바탕으로 개발팀이 즉시 코딩을 시작할 수 있으며, 10주 내에 완전한 듀얼 버전 시스템 구축이 가능합니다.

16.4 비즈니스 가치

- **개인 사용자:** EXE 버전으로 완전한 제어권
- **전문 트레이더:** 웹 대시보드로 24시간 원격 운영
- **확장 가능성:** API 제공으로 타사 연동 가능
- **시장 차별화:** 듀얼 버전 제공하는 유일한 솔루션

문서 끝

본 PRD는 모든 이해관계자의 검토를 거쳐 최종 승인된 완전한 개발 명세서입니다.