

Forward Inheritance SaaS Platform - Product Requirements Document V1.3

Executive Summary

Forward is building a family-first inheritance and wealth transfer SaaS platform centered around Forward Family Circles (FFCs) for collaborative estate planning and wealth transfer transparency, featuring integrated referral systems and comprehensive asset management capabilities.

Table of Contents

1. [Project Context](#)
2. [Strategic Insights](#)
3. [Phase 1A Scope & Core Features](#)
4. [Epic Structure & Implementation](#)
5. [Technical Architecture](#)
6. [Database Schema](#)
7. [Success Metrics](#)
8. [Risk Mitigation](#)
9. [Development Phases](#)

Project Context

Forward is developing a SaaS platform that transforms traditional individual-focused wealth management into collaborative family-centered inheritance planning, providing comprehensive asset management and estate planning collaboration.

Vision

Create a single source of truth for inheritance planning with collaborative, living plans that evolve with family needs, enabling transparent wealth transfer across generations.

Mission

Empower families to make informed inheritance decisions through transparency, collaboration, and professional guidance within a secure, compliant platform.

Market Analysis

Total Addressable Market (TAM)

- 11.8 million high-net-worth households in the US (>\$1M investable assets)¹
- Additional 20.3 million mass-affluent households (250K – 1M)²
- **TAM: 32M households** ♦ *200/year average subscription* = **6.4B annual market**

Serviceable Addressable Market (SAM)

- Based on Cerulli Associates research, ~40% of HNW families actively use digital tools for financial planning³
- Focus on digitally-engaged families planning wealth transfer
- **SAM: 12.8M households (40% of TAM) = \$2.56B annual market**

Serviceable Obtainable Market (SOM)

- 0.5% market penetration target based on comparable fintech SaaS adoption curves⁴
- Conservative estimate accounting for 18-24 month sales cycle
- **SOM: 64,000 families = \$12.8M ARR by Year 3**

Competitive Landscape

Feature	Forward	Vanilla	EstateMap	Wealth.com
Family Collaboration	✓ Full	Basic	Partial	Limited
Marketing Independence	✓ Builder.io	Limited	Limited	None
Advisor Network	✓ Built-in	Partner-only	None	Limited
HEI Integration	✓ Native	None	None	None
Asset Diversity	✓ 13 types	5 types	7 types	9 types
Individual Asset Control	✓ Granular	Basic	Basic	Limited

User Research Insights

Based on market research and industry studies:

- **73%** of families report lack of transparency in inheritance planning⁵
- **62%** want better collaboration tools with their advisors⁶
- **Average estate settlement: 18 months** (opportunity to reduce by 50%)⁷
- **70%** of wealth transfers fail by the third generation due to poor communication and trust breakdown⁸

Sources:

1. Federal Reserve Survey of Consumer Finances, 2022
2. Spectrem Group Affluent Market Insights, 2023
3. Cerulli Associates: U.S. High-Net-Worth and Ultra-High-Net-Worth Markets 2023
4. Based on Bessemer Venture Partners SaaS adoption benchmarks for vertical fintech
5. Forward user research study (N=250 HNW families, Q3 2024)
6. Forward user research study (N=250 HNW families, Q3 2024)
7. EstateExec probate statistics, 2023
8. Williams Group wealth transfer research, 20-year longitudinal study

Strategic Insights

Target Market

- **Primary:** Mass-affluent/HNW families seeking tax-efficient wealth transfer and collaborative estate planning
- **Secondary:** Financial advisors specializing in estate planning and wealth transfer
- **Acquisition Strategy:** Direct marketing campaigns through [Builder.io](#) landing pages

Business Model

- **Revenue Streams:**
 - SaaS subscriptions (tiered family plans)
 - Advisor revenue sharing
 - Referral fees
 - Premium advisory services

- **Customer Journey:** Landing page ♦ registration ♦ FFC creation ♦ asset management ♦ advisor network expansion

Core Differentiation

- **Marketing Independence** through [Builder.io](#) CMS integration
- **Family-first transparency** vs individual portfolio focus
- **Individual asset control** - assets controlled by owners, not FFC owners
- **Comprehensive asset diversity** including alternative and digital assets
- **Enhanced security** with mandatory dual-channel verification
- **Living plans** that evolve with family circumstances
- **Database-first architecture** - all operations through stored procedures
- **Privacy-first documents** - automatic PII masking for document protection
- **Multi-language accessibility** - native Spanish support with expansion roadmap for underserved US communities

Scale Target

- Millions of families with event-driven interaction patterns
- Setup, periodic check-ins, life events (births, deaths, marriages, divorces)
- Seasonal tax planning and annual reviews

Phase 1A Scope & Core Features

Phase 1A: True MVP Core (Days 1-60)

Goal: Launch with marketing foundation, complete FFC onboarding, HEI integration, and comprehensive asset management

Key Features for Launch:

1. Forward Marketing Foundation

- **Forward Landing Page** with [Builder.io](#) CMS integration
- **CMS Access** for marketing team independence
- **A/B Testing Capability** for conversion optimization

2. Complete FFC Onboarding Flow

- **User Registration** with email verification
- **Secure Login** with session management
- **FFC Creation** with unique naming and optional family picture upload
- **Member Invitation System** (email-based)
- **Mandatory Dual-Channel Verification** (email + SMS)
- **Owner Approval Required** for all new members
- **Mandatory Account Creation** for invited members with optional profile picture upload
- **Role Assignment:**
 - FFC Owners (edit rights on FFC level)
 - Can add additional owners with edit rights
 - Other roles: Beneficiary, Non-beneficiary

3. Comprehensive Asset Management

- **All 13 Asset Categories** implemented from Day 1
- **Asset-Level Permissions:** Individual ownership rights per asset
- **Manual Asset Entry** for all asset types
- **HEI Integration** as part of Loan category (read-only from API)

4. Core Security & Infrastructure

- **Authentication & authorization**
- **Granular permissions** (FFC-level and asset-level)
- **Data encryption**
- **Audit logging for all changes**
- **Weekly audit reports** sent to FFC owners

Asset Categories (All 13 Types)

1. Personal Directives (POA, Healthcare Directive, Letter of Intent, HIPAA)
2. Trust
3. Will
4. Personal Property (Jewelry, Art, Pets, Furniture, etc.)
5. Operational Property (Vehicles, Boats, Equipment, Appliances)
6. Inventory
7. Real Estate
8. Life Insurance
9. Financial Accounts (Investments, Bank, Retirement, College)

10. Recurring Income (Royalties)
11. Digital Assets (IP, Digital Assets)
12. Ownership Interests (Business, Franchise)
13. **Loans (including HEI via API, Interfamily Loans)**

Key Permission Model

- **FFC Level:** Owners control FFC structure, invitations, approvals
- **Asset Level:** Each asset has independent ownership/permission structure
- **Critical Principle:** Asset owners control their assets, NOT FFC owners
- **Privacy First:** Assets only visible to personas explicitly granted permission

DEFERRED to Later Phases:

- Chat System (removed entirely from MVP)
- Referral Engine (moved to Phase 1B or later)
- AI Suggestions
- Document Intelligence

Epic Structure & Implementation Roadmap

Phase 1A: Foundation (Days 1-60)

- **Epic 1:** Marketing Foundation & Landing Page (2 weeks)
- **Epic 2:** FFC Onboarding Flow with Enhanced Security (3 weeks)
- **Epic 3:** Comprehensive Asset Management System (4 weeks)

Phase 1B: Enhancement (Days 61-90)

- **Epic 4:** Advanced Features & Integrations (3 weeks)
- **Epic 5:** Multi-Language Support (US Market) (2 weeks)

Phase 2: Enterprise & Compliance (Months 4-9)

- **Epic 6:** SOC 2 Compliance & Trust Management Platform (6 months, parallel)

Post-MVP: Performance & Scale (Year 1)

- **Epic 7:** React Performance Optimization & User Experience Enhancement (3 months)

Technical Architecture

Core Architecture Principles

Mandatory Database Access Layer

All database operations **MUST** use stored procedures and functions:

- No direct table access from application code
- All CRUD operations through stored procedures
- Complex queries through database functions
- Business logic enforcement at database level
- Type-safe operations with pgtyped
- Comprehensive audit logging at procedure level

Benefits:

1. **Security:** SQL injection prevention, access control
2. **Performance:** Optimized execution plans, reduced network traffic
3. **Consistency:** Centralized business rules and validation
4. **Maintainability:** Database API versioning, easier refactoring
5. **Compliance:** Automatic audit trails and data governance

Frontend Technology Stack

Core Framework and Tools

- **React 18+ with TypeScript**
 - Functional components with hooks
 - Strict TypeScript configuration for type safety
 - React Router v6 for client-side routing
 - React Query for server state management
- **Styling and UI Framework**
 - Tailwind CSS for utility-first styling
 - shadcn/ui component library for consistent design
 - CSS modules for component-specific styles
 - Design system implementation from existing mockups
- **Build and Development Tools**

- Vite for fast development and building
- ESLint and Prettier for code quality
- Husky for pre-commit hooks
- Testing with Jest and React Testing Library

Application Architecture

- **Mobile-First Responsive Design**
 - Progressive enhancement approach
 - Breakpoint-based responsive design
 - Touch-friendly interface optimization
 - Cross-browser compatibility (modern browsers)
- **Single Page Application (SPA)**
 - Client-side routing and navigation
 - Code splitting for optimal bundle sizes
 - Lazy loading for performance optimization
 - Progressive Web App (PWA) capabilities
- **Component Architecture**
 - Card-based modular component system
 - Reusable component library
 - Atomic design principles
 - Storybook for component documentation
- **State Management**
 - Context API for global application state
 - React Query for server state caching
 - Local storage for user preferences
 - Session management for authentication

Integration Features

- **Builder.io CMS Integration**
 - Headless CMS for marketing content
 - Dynamic content delivery
 - A/B testing capabilities
 - SEO optimization
- **File Management**
 - Document upload and preview
 - Image optimization and compression
 - Secure file sharing and access control

- Version control for documents

Backend Technology Stack

Core Server Framework

- **Node.js with Express and TypeScript**
 - RESTful API architecture
 - GraphQL for complex data queries (future consideration)
 - Middleware for authentication, logging, and error handling
 - Rate limiting and security headers
- **Database and ORM**
 - PostgreSQL for primary data storage
 - Slonik for safe PostgreSQL client operations
 - pgtyped for compile-time SQL type safety
 - Database migrations and version control
- **Business Logic Architecture**
 - Stored procedures and functions for complex operations
 - Transaction management for data consistency
 - Event-driven architecture for system integration
 - Background job processing with Bull/Agenda

Integration Infrastructure

- **Financial Data Integration**
 - RESTful API clients for financial service providers
 - Real-time webhook handling for account synchronization
 - Data transformation and mapping services
 - Conflict resolution and error handling
- **Email Service Integration**
 - SendGrid for transactional emails
 - Template management for referral campaigns
 - Bounce and complaint handling
 - Email analytics and tracking
- **SMS Service Integration**
 - Twilio for SMS delivery and phone verification
 - WhatsApp Business API for international support
 - Voice call backup for verification
 - Phone number validation and formatting

Security and Authentication

- **Authentication and Authorization**
 - JWT-based authentication
 - Role-based access control (RBAC)
 - Multi-factor authentication (2FA)
 - Session management and security
- **Data Protection**
 - Encryption at rest and in transit
 - PII masking and data anonymization
 - GDPR and CCPA compliance
 - Audit logging for all data access
- **API Security**
 - Rate limiting and DDoS protection
 - Input validation and sanitization
 - CORS and security headers
 - API versioning and deprecation management

AWS Cloud Migration Roadmap

Phase 1: Basic Cloud Infrastructure

- **Content Delivery**
 - CloudFront for global CDN
 - S3 for static asset storage
 - Route 53 for DNS management
 - SSL/TLS certificate management
- **Application Hosting**
 - Amplify for frontend CI/CD and hosting
 - Alternative: S3 + CloudFront for static site hosting
 - API Gateway for secure, scalable API endpoints
 - Lambda functions for serverless operations

Phase 2: Containerized Services

- **Container Orchestration**
 - EKS (Elastic Kubernetes Service) for container management
 - Fargate for serverless container execution
 - Docker containerization for all services
 - Auto-scaling based on demand

- **Database Services**
 - RDS PostgreSQL for development and testing
 - Aurora PostgreSQL for production scalability
 - ElastiCache for Redis caching layer
 - Database backup and disaster recovery

Security and Compliance

Compliance Frameworks

- **SOC 1 & SOC 2 Compliance**
 - Regular third-party audits
 - Control implementation and monitoring
 - Risk assessment and mitigation
 - Continuous compliance monitoring
- **Industry Standards**
 - PCI DSS for payment processing
 - GDPR for data protection
 - CCPA for California privacy rights
 - HIPAA considerations for health directives
- **Penetration Testing**
 - Regular security assessments
 - Vulnerability scanning and remediation
 - Third-party security audits
 - Bug bounty program considerations

Security Infrastructure

- **Vanta Integration**
 - Continuous compliance monitoring
 - Automated security assessments
 - Policy management and enforcement
 - Audit trail maintenance
- **Access Controls**
 - Two-factor authentication enforcement
 - Single sign-on (SSO) integration
 - Session management and timeout
 - IP whitelisting and geo-blocking
- **Data Protection**

- Context-aware access controls
- Dynamic PII masking
- Data loss prevention (DLP)
- Encryption key management
- **Network Security**
 - VPC segmentation and isolation
 - Network access control lists (NACLs)
 - Security groups and firewall rules
 - VPN access for administrative functions

Performance Requirements

API Performance Targets

- **Response Times**
 - p50: < 100ms
 - p95: < 200ms
 - p99: < 500ms
- **Throughput**
 - 10,000 concurrent users
 - 100 financial sync operations/second

Frontend Performance

- **Page Load Times**
 - First Contentful Paint (FCP): < 1.5s
 - Time to Interactive (TTI): < 3.5s
 - Cumulative Layout Shift (CLS): < 0.1
- **Mobile Performance**
 - Touch response: < 100ms
 - Smooth scrolling: 60fps
 - Offline capability for critical features

Database Performance

- **Query Performance**
 - Simple queries: < 10ms
 - Complex queries: < 50ms
 - Bulk operations: < 500ms
- **Connection Management**

- Connection pool: 100-500 connections
- Query timeout: 30s
- Transaction timeout: 60s

Database Schema

Complete database schema designs are documented in [architecture.md](#).

Key Components:

- **Multi-Tenant Core Schemas:** Tenants, Personas, FFCs, Contact/Communication
- **Enhanced Asset Management:** 13 asset categories with ownership and permissions
- **Invitation & Verification:** Dual-channel verification system
- **Audit & Compliance:** Comprehensive audit trails and security measures
- **Performance Indexes:** Optimized for multi-tenant operations

Reference Sections in [architecture.md](#):

- **Multi-Tenant Core Schemas:** Complete table definitions for tenants, personas, FFCs
- **Contact and Communication Schema:** Flexible address, phone, and email management
- **Enhanced Asset Management Schema:** All 13 asset categories with ownership models
- **Invitation and Verification Schema:** Dual-channel security verification
- **Enhanced Audit and Compliance Schema:** Comprehensive audit logging
- **Performance Indexes:** All database indexes for optimal performance

See [architecture.md](#) for complete table definitions, triggers, constraints, and indexes.

Success Metrics

Launch Metrics (Day 1-30)

- **Registration Completion Rate:** >85%
 - Measured from landing page to verified account
 - A/B test onboarding variations
- **FFC Creation Rate:** >70% of registered users
 - Time from registration to first FFC
 - Track abandonment points
- **First Asset Added:** >60% of FFCs

- Manual entry vs. import usage
- Category distribution analysis
- **Member Invites Sent:** Average 3 per FFC
 - Dual-channel verification completion rate
 - Acceptance rate tracking
 - Time to acceptance
- **Invite Acceptance Rate:** >40%
 - By relationship type
 - Re-invitation success
- **Phone Verification Success:** >90%
 - SMS delivery rate
 - Code entry success rate
 - International vs domestic performance

Growth Metrics (Day 31-90)

- **Weekly Active Family Circles (WAFCs):** 60%
 - Definition: 2+ members, 3+ actions/week
 - Growth target: 10% WoW
- **Assets per FFC:** Average 5+
 - Distribution by category
 - Value concentration analysis
- **Landing Page Performance:**
 - Conversion rate >2%
 - Page load time <2 seconds
 - A/B test winners implemented
- **Marketing Independence:** 100% of content updates by marketing team
- **Asset Permission Changes:** Track frequency and patterns
- **Security Metrics:** Zero successful phishing attempts

North Star Metric

Weekly Active Family Circles (WAFCs)

- **Definition:** FFCs with 2+ members and 3+ platform actions per week
- **Target:** 50% of all FFCs are WAFCs by Day 90
- **Growth:** 10% week-over-week increase in WAFCs
- **Why it matters:** Indicates true platform value and stickiness

Business KPIs

Platform Foundation Success

- **Target:** 95% successful family onboarding completion
- **Measurement:** End-to-end flow completion rate
- **Sub-metrics:**
 - Setup abandonment: <5%
 - Time to complete: <15 minutes
 - User satisfaction: >4.0/5

Security Performance

- **Verification Success Rate:** >90%
- **Phone Verification Completion:** >85%
- **False Positive Rate:** <1%
- **Security Incident Rate:** 0

Marketing Performance

- **Landing Page Load Time:** <2 seconds
- **Form Conversion Rate:** >2%
- **Content Update Frequency:** Daily capability
- **A/B Test Velocity:** 2+ tests per month

Risk Mitigation

Risk Scoring Matrix

Risk	Probability	Impact	Score	Priority	Mitigation Strategy
Phone Verification Delivery	Medium (40%)	High	16	P0	Multiple SMS providers, voice backup, WhatsApp integration
Builder.io Performance Impact	Medium (30%)	Medium	9	P1	CDN optimization, fallback content, performance monitoring
Asset Permission Complexity	High (60%)	Medium	18	P0	Clear UI, extensive testing, user education,

Risk	Probability	Impact	Score	Priority	Mitigation Strategy
					progressive disclosure
Invitation Fraud/Abuse	Low (20%)	High	12	P1	Rate limiting, phone verification, audit trails, manual review
International SMS Costs	High (70%)	Low	7	P2	WhatsApp backup, voice alternatives, cost monitoring

Security Risks

Phone Verification System

Risk: SMS delivery failures or phone number spoofing

Mitigation:

- Multi-provider SMS strategy (Twilio primary, backup providers)
- Voice call verification as backup
- WhatsApp Business API for international
- Rate limiting and abuse detection
- Complete audit trail for all verification attempts

Asset Permission System

Risk: Complex permissions leading to user errors or security gaps

Mitigation:

- Progressive disclosure of advanced features
- Clear visual indicators for permission levels
- Default to secure permissions
- Permission preview before saving
- Extensive user testing and iterative improvement

Invitation Security

Risk: Phishing or unauthorized access attempts

Mitigation:

- Dual-channel verification required

- Unique tokens with expiration
- Owner approval after verification
- Comprehensive audit logging
- Clear security messaging to users

Technical Risks

Builder.io Integration

Risk: Performance impact or service dependency

Mitigation:

- CDN caching for all [Builder.io](#) content
- Fallback static content for outages
- Performance monitoring and alerting
- Regular performance testing
- Alternative CMS evaluation

Database Performance

Risk: Complex permission queries causing slowdowns

Mitigation:

- Optimized indexes for permission checks
- Query performance monitoring
- Caching layer for frequently accessed data
- Database scaling plan
- Regular performance reviews

Business Risks

User Adoption of Security Features

Risk: Complex verification process reducing adoption

Mitigation:

- Clear explanation of security benefits
- Progressive onboarding
- Support team training
- User feedback collection
- Iterative UX improvements

Marketing Team **Builder.io** Learning Curve

Risk: Delayed marketing capability due to tool complexity

Mitigation:

- Comprehensive **Builder.io** training
- Template library for common use cases
- Dedicated support channel
- Documentation and video tutorials
- Gradual feature rollout

Early Warning System

Monitoring Triggers:

- SMS delivery rate <85% ⚠️ Switch providers
- Landing page load time >3s ⚠️ Performance investigation
- Permission errors >5/day ⚠️ UI review
- Failed verification attempts >10% ⚠️ Security review
- Asset creation abandonment >30% ⚠️ UX analysis

Epic Structure & Implementation

Epic 1: Marketing Foundation & Landing Page

Goal: Establish marketing independence through **Builder.io** CMS integration and high-converting landing pages.

Duration: 2 weeks (Sprints 1-2)

Story 1.1: **Builder.io** CMS Integration

As a marketing team member

I want to manage landing page content independently

So that we can iterate quickly without developer dependencies

Acceptance Criteria:

- **Builder.io** workspace configured with Forward branding
- Landing page templates created for different campaigns
- Visual editor accessible to marketing team

- Content changes reflect in production within 5 minutes
- A/B testing capabilities enabled

Story 1.2: Landing Page Performance & SEO

As a visitor

I want fast-loading, discoverable landing pages

So that I can quickly learn about Forward's services

Acceptance Criteria:

- Page load time <1.5 seconds (First Contentful Paint)
- Google PageSpeed score >90
- SEO meta tags and structured data implemented
- Mobile responsiveness across all devices
- Analytics tracking configured (Google Analytics, Hotjar)

Story 1.3: Lead Capture System

As a potential customer

I want to easily express interest and get follow-up

So that I can learn more about Forward's services

Acceptance Criteria:

- Contact form with progressive profiling
- Email validation and spam protection
- CRM integration (HubSpot/Salesforce)
- Automated follow-up email sequence
- Lead scoring and qualification workflow

Database Implementation:

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 1: Marketing Foundation" for complete stored procedure specifications including `capture_marketing_lead` function.

Epic 2: FFC Onboarding Flow with Enhanced Security

Goal: Enable secure family circle creation with mandatory dual-channel verification.

Duration: 3 weeks (Sprints 3-5)

Story 2.1: User Registration & Email Verification

As a new user

I want to create an account securely

So that I can start organizing my family's wealth

Acceptance Criteria:

- Strong password requirements (12+ chars, complexity)
- Email verification required before account activation
- Account lockout after 5 failed attempts
- Password reset functionality
- Terms of service and privacy policy acceptance

Database Implementation:

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 2: FFC Onboarding" for complete stored procedure specifications including `register_user` and `invite_ffc_member` functions.

Story 2.2: FFC Creation Wizard

As a verified user

I want to create my family forward circle easily

So that I can start collaborating with family members

Acceptance Criteria:

- Multi-step wizard with progress indicator
- FFC name uniqueness validation
- Family description and goals capture
- Initial role assignment (Owner)
- Success confirmation with next steps

Story 2.3: Enhanced Member Invitation System

As an FFC owner

I want to invite family members with dual-channel verification

So that we ensure maximum security for sensitive family data

Acceptance Criteria:

- Email invitation with secure token

- Mandatory phone number verification (SMS)
- Invite expiration after 7 days
- Role assignment during invitation
- Owner approval required for all invitations

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 2: FFC Onboarding" for complete `invite_ffc_member` stored procedure specification.

Story 2.4: Phone Verification System

As an invited family member

I want to verify my phone number

So that the FFC owner can be confident in my identity

Acceptance Criteria:

- SMS code sent via Twilio
- 6-digit numeric code format
- Code expires after 10 minutes
- Maximum 3 verification attempts
- Fallback to voice call option

Story 2.5: Owner Approval Workflow

As an FFC owner

I want to approve member invitations after verification

So that I maintain control over who joins my family circle

Acceptance Criteria:

- Notification to owner when verification complete
- Approval/rejection interface with reason capture
- Automatic approval option (configurable)
- Member onboarding email after approval
- Activity log for all approval decisions

Epic 3: Comprehensive Asset Management System

Goal: Enable secure tracking and management of all 13 asset categories with document support and PII protection.

Duration: 4 weeks (Sprints 6-9)

Database Implementation: All stored procedures for asset management, document handling, and PII processing are documented in [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 3: Asset Management & PII Protection"

Story 3.1: Asset Category Infrastructure

As a system architect

I want robust asset category support

So that users can track all types of family wealth

Acceptance Criteria:

- All 13 asset categories implemented
- Category-specific database schemas
- Dynamic form generation per category
- Validation rules per asset type
- Category icons and descriptions

Story 3.2: Asset-Persona Ownership Model

As an asset owner

I want direct ownership tracking

So that my assets are clearly attributed to me

Acceptance Criteria:

- Direct persona-to-asset ownership links
- Percentage ownership support (0.01% precision)
- Ownership type classification (direct, trust, beneficiary)
- Ownership transfer functionality
- Ownership history tracking

Database Implementation:

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 3: Asset Management" for complete stored procedure specifications including `create_asset_with_ownership` function.

Story 3.3: Document & Photo Management with PII Protection

As an asset owner

I want to upload supporting documents and photos

So that I can maintain complete asset records while protecting sensitive information

Acceptance Criteria:

- Multiple file uploads per asset
- PDF, JPG, PNG, DOC, XLS format support
- Files up to 10MB each, 100MB total per asset
- Document categorization (deed, statement, photo, etc.)
- Original document encryption in S3
- Automatic PII detection and masking
- Separate storage for PII-masked versions

PII Masking System Architecture:

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 3: PII Protection" for complete stored procedure specifications including `upload_asset_document` and `complete_pii_processing` functions.

PII Detection Pipeline:

1. **Upload:** Original document encrypted and stored in S3
2. **Text Extraction:** OCR/text parsing for searchable content
3. **PII Detection:** AWS Comprehend identifies SSNs, phone numbers, addresses, names
4. **Masking:** Sensitive data replaced with tokens (XXX-XX-1234)
5. **Dual Storage:** Original (encrypted) + masked version (encrypted)
6. **Access Control:** Role-based access to original vs masked versions

Supported PII Types:

- Social Security Numbers
- Phone Numbers
- Email Addresses
- Physical Addresses
- Financial Account Numbers
- Names and Personal Identifiers
- Date of Birth

- Driver's License Numbers

Story 3.4: Individual Asset Permissions

As an asset owner

I want granular control over who can see my assets

So that I maintain privacy within my family circle

Acceptance Criteria:

- Read/Edit/Admin permission levels
- Per-asset permission assignment
- Inheritance of FFC-level permissions (optional)
- Permission override capabilities
- Bulk permission management
- Permission audit trail

Story 3.5: Asset Dashboard & Visualization

As an FFC member

I want clear visualization of family assets

So that I understand our collective wealth picture

Acceptance Criteria:

- Category-based asset grouping
- Total value calculations by category
- Visual charts and graphs
- Filter by ownership, category, value
- Export capabilities (PDF, Excel)
- Mobile-responsive design

Story 3.6: HEI Integration (Read-Only)

As a family with HEI loans

I want to see loan information in our asset dashboard

So that we have a complete financial picture

Acceptance Criteria:

- HEI API integration for loan data
- Read-only display in loan category

- Automatic data synchronization
- Integration with permission system
- Loan performance metrics display

Epic 4: Advanced Features & Integrations

Goal: Enhance platform capabilities with advanced search, reporting, audit trails, and third-party integrations.

Duration: 3 weeks (Sprints 10-12)

Database Implementation: All stored procedures for search, reporting, audit trails, integrations, and Quillt API functions are documented in [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 4: Reporting, Analytics & Integrations" and "Integration Architecture > Quillt Integration Stored Procedures"

Story 4.1: Advanced Search & Filtering System

As an FFC member

I want powerful search capabilities across all assets and documents

So that I can quickly find specific information within our family wealth

Acceptance Criteria:

- Global search across all asset categories
- Advanced filters (category, value range, owner, date)
- Search within document content (OCR text)
- Saved search queries
- Search result sorting and pagination
- Search analytics and suggestions

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 4: Advanced Features" for complete stored procedure specifications including `search_family_assets` function.

Story 4.2: Comprehensive Reporting & Analytics

As an FFC owner

I want detailed reports on our family wealth

So that I can make informed financial decisions and track changes over time

Acceptance Criteria:

- Wealth distribution reports by category and person
- Asset growth tracking over time
- PDF report generation with charts
- Scheduled report delivery (email)
- Custom report builder
- Export to Excel/CSV formats

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 4: Reporting & Analytics" for complete stored procedure specifications including `generate_wealth_report` function.

Story 4.3: Comprehensive Audit Trail System

As a compliance officer

I want complete audit trails for all platform activities

So that we maintain regulatory compliance and security accountability

Acceptance Criteria:

- All CRUD operations logged with user context
- IP address and device tracking
- Data change history with before/after values
- Retention policy implementation (7 years)
- Audit log search and filtering
- Suspicious activity detection and alerts

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 4: Audit & Compliance" for complete stored procedure specifications including `log_audit_event` function.

Story 4.4: Bulk Operations & Data Management

As an FFC owner with many assets

I want bulk operations for efficient management

So that I can update multiple assets quickly and maintain data consistency

Acceptance Criteria:

- Bulk asset updates (category, permissions, values)

- Bulk document uploads with batch processing
- Data validation and error reporting
- Rollback capabilities for failed operations
- Progress tracking for long-running operations
- Bulk export and import functionality

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 4: Bulk Operations" for complete stored procedure specifications including `bulk_update_assets` function.

Story 4.5: Third-Party Integration Framework

As a platform administrator

I want extensible integration capabilities

So that we can connect with financial institutions and estate planning tools

Acceptance Criteria:

- Generic webhook system for external notifications
- OAuth 2.0 framework for secure API access
- Rate limiting and API key management
- Integration health monitoring
- Data synchronization job management
- Error handling and retry logic

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Epic-Specific Stored Procedures > Epic 4: Integration Management" for complete stored procedure specifications including `manage_integration` function.

Story 4.6: Quillt API Integration for Financial Accounts

As an FFC member

I want to connect my bank accounts through Quillt

So that my account balances update automatically without manual entry

Acceptance Criteria:

- Quillt Profile creation and management for FFC members
- Bank account connection flow with OAuth authentication
- Automatic balance refresh (weekly/monthly scheduled)

- Multiple accounts per connection support
- Webhook handling for balance updates
- Connection health monitoring and re-authentication prompts
- Support for 10,000+ financial institutions via Quillt

Integration Architecture:

FFC Member → Quillt Profile → Bank Connection → Multiple Accounts → Forward Assets

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Integration Architecture > Quillt Integration Stored Procedures" for complete stored procedure specifications including `manage_quillt_connection` and `process_quillt_webhook` functions.

Quillt Integration Workflow:

1. **Profile Creation:** Map FFC member to Quillt Profile with metadata
2. **Bank Connection:** User connects via Quillt's OAuth flow
3. **Account Discovery:** Quillt discovers multiple accounts per connection
4. **Asset Creation:** Each account becomes a Financial Account asset
5. **Balance Sync:** Webhook updates or scheduled refreshes update balances
6. **Error Handling:** Connection health monitoring triggers re-authentication

Supported Account Types:

- Checking and Savings Accounts
- Credit Cards and Lines of Credit
- Investment Accounts (401k, IRA, Brokerage)
- Loan Accounts (Mortgage, Auto, Personal)

Story 4.7: Real Estate Data Provider Integration

As an FFC member with real estate assets

I want automated property value updates

So that my real estate portfolio reflects current market values

Acceptance Criteria:

- Research and select optimal real estate data provider (Zillow API, CoreLogic, RentSpree, etc.)
- Property identification via address lookup

- Automated valuation model (AVM) integration
- Comparable sales (comps) data integration
- Property detail enrichment (square footage, lot size, etc.)
- Market trend analysis and alerts
- Scheduled value updates (monthly/quarterly)

Provider Evaluation Criteria:

- **Coverage:** National property database completeness
- **Accuracy:** AVM accuracy vs actual sales data
- **API Quality:** Rate limits, uptime, data freshness
- **Cost Structure:** Per-call pricing vs subscription models
- **Data Richness:** Property details, history, market trends

Database implementation details are documented in [architecture.md](#)

See [architecture.md](#) section "Integration Architecture > Real Estate Provider Integration" for complete stored procedure specifications including `manage_real_estate_integration` function.

Provider Research Tasks:

1. **API Comparison:** Zillow API vs CoreLogic vs RentSpree vs others
2. **Pricing Analysis:** Cost per property lookup and ongoing updates
3. **Data Quality Assessment:** Accuracy testing with known property values
4. **Integration Complexity:** Authentication, rate limits, webhook support
5. **Legal Compliance:** Terms of service and data usage restrictions

Story 4.8: Performance Optimization & Caching

As a system user

I want fast response times even with large datasets

So that the platform remains responsive as our family wealth grows

Acceptance Criteria:

- Redis caching for frequently accessed data
- Database query optimization and indexing
- API response time monitoring (<200ms p95)
- Pagination for large result sets
- Background job processing for heavy operations
- CDN integration for static assets

Epic 5: Multi-Language Support (US Market)

Goal: Expand market reach to underserved Spanish-speaking communities in the US with native language support and cultural adaptations.

Duration: 6 months (Post-MVP, Year 1 expansion)

Strategic Priority: High - \$170M ARR market opportunity with zero competition

Story 5.1: Spanish Language Foundation

As a Spanish-speaking user
I want complete Spanish language support
So that I can confidently manage my family's inheritance planning in my native language

Acceptance Criteria:

- User registration and profile language preferences (English/Spanish)
- Complete UI translation for all core user flows
- Asset category and document type translations
- Spanish legal document templates (professionally translated and legally reviewed)
- Email communications in user's preferred language
- Cultural UX adaptations for Hispanic family structures
- Professional translation quality with financial terminology accuracy

Market Validation Metrics:

- 15-25% of new signups from Spanish speakers within 6 months
- 85%+ monthly retention rate for Spanish users
- \$100K+ ARR from Spanish market by month 9
- 25%+ referral rate within Spanish-speaking communities

Future Language Expansion Roadmap:

Language	US Speakers	Market Potential	Timeline
Chinese	3.5M	\$56M ARR	Year 2
Korean	1.1M	\$19M ARR	Year 2-3
Vietnamese	1.7M	\$9M ARR	Year 3
Arabic	1.2M	\$7M ARR	Year 3-4

Language	US Speakers	Market Potential	Timeline
Russian	900K	\$6M ARR	Year 4+

Note: Additional languages will only be pursued after Spanish market success validation

Technical Implementation: All translation infrastructure, helper functions, and database schema extensions are documented in [architecture.md](#) section "Multi-Language Support Architecture".

Epic 6: SOC 2 Compliance & Trust Management Platform

Epic Goal: Establish enterprise-grade compliance, security transparency, and customer trust through comprehensive Vanta API integration, automated SOC 2 compliance monitoring, and real-time security posture visibility.

Business Value: Enable faster enterprise sales cycles, reduce security questionnaire burden by 90%, achieve SOC 2 Type II certification, and build customer trust through transparent security practices.

Market Impact: \$2.4M+ ARR opportunity from enterprise customers requiring SOC 2 compliance for vendor approval.

Story 6.1: Vanta API Foundation & Authentication

As a platform administrator
I want to integrate with Vanta's compliance automation platform
So that we can automate SOC 2 evidence collection and monitoring

Acceptance Criteria:

- 1. OAuth 2.0 authentication setup with Vanta API
- 2. Secure credential management with environment variables
- 3. API rate limiting compliance (5/min OAuth, 20/min Integration, 50/min Management)
- 4. Error handling and retry logic for API failures
- 5. Health check monitoring for Vanta API connectivity
- 6. Audit logging for all Vanta API interactions
- 7. Webhook endpoint setup for real-time compliance alerts
- 8. Integration testing suite for all API endpoints
- 9. Documentation for API configuration and maintenance
- 10. Monitoring dashboard for API usage and status

Technical Requirements:

- Node.js/Python SDK for Vanta API integration
- Redis caching for API rate limit management
- PostgreSQL logging for audit trails
- Environment-based configuration management

Story 6.2: Automated SOC 2 Evidence Collection

As a compliance officer

I want automated evidence collection for SOC 2 controls

So that audit preparation is streamlined and continuous

Acceptance Criteria:

1. PII Processing Evidence Integration:

- Automated submission of PII detection job results
- Masking strategy compliance evidence
- Data processing audit trail sync
- Confidence score and accuracy metrics

2. Database Security Evidence:

- User access review submissions
- Database encryption status reporting
- Row-level security validation evidence
- Stored procedure access audit logs

3. Multi-tenant Isolation Evidence:

- Tenant segregation verification
- Cross-tenant access prevention tests
- FFC-level data isolation proof
- Access control matrix validation

4. Asset Management Evidence:

- Document encryption compliance
- Access permission audit trails
- Asset ownership verification logs
- Financial data protection evidence

5. Integration Security Evidence:

- Third-party vendor assessments (Quillt, AWS, real estate APIs)
- API security validation
- Webhook signature verification
- External data handling compliance

6. Automated Scheduling:

- Daily evidence collection jobs
- Weekly compliance status reports
- Monthly vendor risk assessments
- Quarterly comprehensive audits

7. **Real-time Monitoring:**

- Failed control notifications
- Security incident alerts
- Compliance drift warnings
- Remediation task creation

Business Rules:

- Evidence collection runs continuously, not just during audits
- All submissions include tenant isolation verification
- Failed evidence collection triggers immediate alerts
- Evidence retention follows SOC 2 requirements (minimum 3 years)

Story 6.3: Customer-Facing Trust Center

As a prospective enterprise customer

I want to view Forward's real-time security posture

So that I can make informed vendor selection decisions

Acceptance Criteria:

1. **Public Trust Center Page:**

- Real-time SOC 2 Type II compliance status
- Current certification details and expiration dates
- Security framework adherence (ISO 27001, GDPR, CCPA)
- Incident-free days counter
- Overall security score display

2. **Compliance Artifacts:**

- Downloadable SOC 2 reports (gated behind contact form)
- Security questionnaire auto-responses
- Penetration testing summaries
- Compliance framework mappings

3. **Transparent Incident Communication:**

- Public incident status page
- Historical incident transparency
- Incident response time metrics

- Resolution communication timeline

4. **Security Metrics Dashboard:**

- Uptime statistics (99.9%+ target)
- Data encryption status
- Security training completion rates
- Vulnerability remediation SLAs

5. **Customer Access Features:**

- Authenticated customer portal for detailed reports
- Quarterly security reviews for enterprise clients
- Custom security questionnaire responses
- Direct compliance officer contact information

6. **SEO and Marketing Integration:**

- Search engine optimized trust content
- Integration with [Builder.io](#) marketing pages
- Social proof through compliance badges
- Case studies of security-conscious customers

Technical Requirements:

- Public API endpoints for trust center data
- CDN caching for fast loading
- Mobile-responsive design
- Analytics tracking for trust center engagement

Story 6.4: Continuous Risk Management & Monitoring

As a security team member

I want continuous monitoring of our security posture

So that compliance issues are identified and resolved proactively

Acceptance Criteria:

1. **Real-time Control Monitoring:**

- 1200+ automated hourly compliance tests
- Control failure immediate alerting
- Remediation task auto-creation
- SLA tracking for resolution times

2. **Vendor Risk Management:**

- Quarterly security assessments for all vendors
- Integration security posture monitoring

- Vendor compliance status tracking
- Risk score calculations and trending

3. **Asset and Resource Monitoring:**

- AWS infrastructure security compliance
- Database security configuration validation
- Application security scanning results
- Network security posture assessment

4. **Access Control Monitoring:**

- User access review automation
- Privileged account monitoring
- Session management compliance
- Failed login attempt analysis

5. **Incident Response Integration:**

- Automated incident creation in Vanta
- Security event correlation
- Response time tracking
- Post-incident review automation

6. **Compliance Drift Detection:**

- Configuration change monitoring
- Policy deviation alerts
- Unauthorized access detection
- Data handling compliance verification

7. **Management Reporting:**

- Executive security dashboards
- Board-level compliance reports
- Trend analysis and predictions
- ROI metrics for security investments

Alert Thresholds:

- Critical: Security control failures, data breaches
- High: Vendor risk changes, privileged access anomalies
- Medium: Configuration drifts, policy violations
- Low: Informational updates, scheduled reviews

Story 6.5: Automated Security Questionnaire Response

As a sales team member

I want automated security questionnaire responses

So that enterprise sales cycles are accelerated

Acceptance Criteria:

1. Questionnaire Automation:

- AI-powered questionnaire analysis
- Automated response generation from Vanta data
- 90%+ automation rate for standard questions
- Custom response library for specialized questions

2. Response Accuracy:

- Real-time data synchronization with Vanta
- Automated fact-checking against current compliance status
- Version control for response templates
- Legal review workflow for sensitive responses

3. Integration with Sales Process:

- CRM integration for questionnaire tracking
- Sales team notification for manual responses needed
- Customer portal for questionnaire submission
- Response time SLA tracking (24-48 hours)

4. Response Categories:

- Data security and encryption
- Access controls and authentication
- Incident response procedures
- Business continuity planning
- Vendor management practices
- Compliance certifications
- Data handling and privacy
- Infrastructure security

5. Quality Assurance:

- Legal team review for first-time questions
- Automated accuracy scoring
- Customer feedback integration
- Continuous improvement tracking

6. Competitive Advantage:

- Fastest response time in industry
- Most comprehensive documentation
- Real-time compliance verification
- Proactive security communication

Success Metrics:

- 90% questionnaire automation rate
- 50% reduction in sales cycle time for enterprise deals
- 95% customer satisfaction with security responses
- 24-hour average response time

Story 6.6: Advanced Compliance Analytics & Reporting

As an executive stakeholder

I want comprehensive compliance analytics and insights

So that I can make informed decisions about security investments

Acceptance Criteria:

1. Executive Dashboard:

- Overall compliance health score
- Trend analysis for security posture improvements
- Cost-benefit analysis of security investments
- Competitive compliance positioning

2. Operational Metrics:

- Mean time to remediation (MTTR) for security issues
- Control effectiveness measurements
- Audit preparation time reduction metrics
- Employee security training completion rates

3. Risk Analytics:

- Risk heat maps by control category
- Vendor risk scoring and trends
- Threat landscape impact assessment
- Business impact analysis for security events

4. Compliance ROI Tracking:

- Cost savings from automation (target: \$500K annually)
- Sales cycle acceleration impact
- Customer acquisition correlation with compliance
- Audit cost reduction measurements

5. Predictive Analytics:

- Compliance drift predictions
- Security incident probability modeling
- Resource allocation optimization

- Future compliance requirement planning

6. Custom Reporting:

- Board presentation auto-generation
- Regulatory filing support
- Customer-specific compliance reports
- Industry benchmark comparisons

Technical Requirements:

- Business intelligence integration
- Data visualization with charts and graphs
- Export capabilities (PDF, Excel, CSV)
- Automated report scheduling and distribution

Epic 6 Success Criteria:

- ☐ SOC 2 Type II certification achieved within 6 months
- ☐ 90% reduction in manual compliance work
- ☐ 50% faster enterprise sales cycles
- ☐ 99%+ automated security questionnaire response rate
- ☐ Customer trust center with 95%+ satisfaction rating
- ☐ Zero compliance violations or control failures
- ☐ \$500K+ annual cost savings from automation
- ☐ 75%+ reduction in audit preparation time

Implementation Timeline:

- **Month 1:** Vanta API integration and basic evidence collection
- **Month 2:** Trust center development and customer-facing features
- **Month 3:** Advanced monitoring and risk management setup
- **Month 4:** Questionnaire automation and sales integration
- **Month 5:** Analytics platform and executive dashboards
- **Month 6:** SOC 2 audit completion and certification

ROI Projection:

- **Investment:** *150K development* + 50K annual Vanta subscription
- **Returns:** *2.4M ARR from enterprise customers* + 500K operational savings
- **Payback Period:** 3 months
- **3-Year NPV:** \$6.2M

Epic 7: React Performance Optimization & User Experience Enhancement

Epic Goal: Optimize the Forward platform's React application for superior performance, achieving sub-second load times, 60fps interactions, and efficient resource utilization to support millions of users at scale.

Business Value: Improve user retention by 25% through faster load times, reduce infrastructure costs by 40% through efficient resource usage, and enable seamless experience for users on slower connections and older devices.

Duration: 3 months (Year 1, Post-MVP validation)

Strategic Priority: High - Critical for user satisfaction and platform scalability

Story 7.1: Bundle Optimization & Code Splitting

As a platform user

I want pages to load instantly

So that I can access my family's information without delays

Acceptance Criteria:

1. Route-based Code Splitting:

- Implement lazy loading for all major routes
- Each route bundle <50KB gzipped
- Preload critical routes based on user behavior
- Progressive enhancement for slow connections

2. Component-level Code Splitting:

- Heavy components (charts, document viewers) lazy loaded
- Suspense boundaries with skeleton loaders
- Error boundaries for failed chunk loads
- Retry logic for network failures

3. Bundle Analysis & Optimization:

- Weekly bundle size monitoring
- Automated alerts for bundles >100KB
- Tree shaking verification
- Duplicate dependency elimination

4. Third-party Library Optimization:

- Replace moment.js with date-fns (80% size reduction)
- Dynamic imports for large libraries
- CDN delivery for common dependencies
- Version locking for consistent performance

Technical Implementation:

```
// Route-based splitting
const AssetDashboard = lazy(() =>
  import(/* webpackChunkName: "assets" */ './AssetDashboard')
);

// Component splitting with retry
const DocumentViewer = lazy(() =>
  retry(() => import('./DocumentViewer'), 3, 1000)
);

// Preload critical routes
const preloadAssets = () => {
  import(/* webpackPrefetch: true */ './AssetDashboard');
};
```

Story 7.2: React Compiler & Rendering Optimization

As a user managing multiple assets

I want smooth, responsive interactions

So that updating information feels instantaneous

Acceptance Criteria:

1. React Compiler Integration (when stable):

- Automatic component memoization
- Optimized re-render elimination
- Development vs production behavior parity
- Performance regression testing

2. Manual Optimization (current):

- Strategic memo() usage for expensive components
- useMemo/useCallback for complex calculations
- React DevTools Profiler integration
- Render phase optimization

3. **Virtual Scrolling:**

- Implement for asset lists >50 items
- Smooth scrolling at 60fps
- Dynamic row height support
- Keyboard navigation maintained

4. **State Management Optimization:**

- Atom-based state for granular updates
- Selector optimization to prevent cascading renders
- Local state for UI-only concerns
- Normalized data structures

Story 7.3: Progressive Web App (PWA) Implementation

As a mobile user

I want app-like performance on my phone

So that I can access Forward anywhere

Acceptance Criteria:

1. **Service Worker Implementation:**

- Offline access to viewed assets
- Background sync for updates
- Push notification support
- Cache versioning strategy

2. **App Shell Architecture:**

- Core UI cached for instant loads
- Navigation available offline
- Graceful degradation for features
- Update prompts for new versions

3. **Performance Metrics:**

- First Paint <1s on 3G
- Time to Interactive <3s
- Lighthouse PWA score >95
- 0 cumulative layout shift

4. **Install Experience:**

- Add to Home Screen prompt
- Custom splash screen
- Native app-like transitions
- Biometric authentication support

Story 7.4: Image & Document Optimization

As a user uploading documents

I want fast previews and minimal data usage

So that I can efficiently manage documents

Acceptance Criteria:

1. Image Optimization Pipeline:

- Automatic WebP conversion with fallbacks
- Responsive image sizing (srcset)
- Lazy loading with intersection observer
- Progressive JPEG for large images
- Blurhash placeholders for UX

2. Document Preview Optimization:

- Generate thumbnails server-side
- PDF.js lazy loading
- Page-by-page rendering
- Low-quality preview while loading

3. Upload Optimization:

- Client-side compression before upload
- Resumable uploads for large files
- Progress indicators with time estimates
- Batch upload optimization

4. CDN Strategy:

- CloudFront distribution for all media
- Automatic cache invalidation
- Geographic edge caching
- Bandwidth optimization per region

Story 7.5: Runtime Performance Monitoring

As a development team

I want real-time performance insights

So that we can proactively fix issues

Acceptance Criteria:

1. Real User Monitoring (RUM):

- Core Web Vitals tracking

- User session performance recording
- Performance budgets enforcement
- Automated performance regression alerts

2. Custom Performance Metrics:

- Time to First Asset Display
- FFC Load Complete Time
- Search Response Time
- Document Upload Duration

3. Performance Dashboard:

- P50/P75/P95/P99 percentiles
- Geographic performance breakdown
- Device category analysis
- Network speed impact correlation

4. A/B Testing Framework:

- Performance-focused experiments
- Statistical significance for metrics
- Rollback capabilities
- User segment targeting

Story 7.6: Server-Side Rendering (SSR) Exploration

As a user discovering Forward via search

I want instant page loads from search results

So that my first impression is positive

Acceptance Criteria:

1. Hybrid Rendering Strategy:

- SSR for public/marketing pages
- SSG for documentation/help
- CSR for authenticated app
- Edge rendering for personalization

2. SEO Optimization:

- Meta tags for all public pages
- Structured data markup
- XML sitemap generation
- Social media previews

3. Performance Targets:

- TTFB <200ms globally

- Full SSR page load <1s
- Seamless hydration
- No layout shift on interaction

4. **Caching Strategy:**

- Edge caching for static content
- User-specific cache keys
- Incremental Static Regeneration
- Cache warming for popular pages

Story 7.7: Accessibility Performance

As a user with assistive technology

I want fast, accessible interactions

So that Forward is usable for everyone

Acceptance Criteria:

1. **Keyboard Navigation Optimization:**

- Focus management in lazy-loaded content
- Skip links for navigation
- Keyboard shortcuts for common actions
- Focus trap in modals

2. **Screen Reader Performance:**

- ARIA live regions for updates
- Semantic HTML throughout
- Loading state announcements
- Error message associations

3. **Reduced Motion Support:**

- Respect prefers-reduced-motion
- Alternative transitions
- Instant state changes option
- Progress indicators without animation

4. **High Contrast & Dark Mode:**

- System preference detection
- Instant theme switching
- Persistent user choice
- No flash of unstyled content

Epic 7 Success Criteria:

- ☐ Initial page load <1.5s on 4G networks
- ☐ Time to Interactive <3s for all major flows
- ☐ 60fps scrolling and interactions
- ☐ Bundle size reduction of 50%
- ☐ Lighthouse performance score >95
- ☐ Core Web Vitals in "Good" range for 90% of users
- ☐ 25% improvement in user engagement metrics
- ☐ 40% reduction in infrastructure costs
- ☐ PWA install rate >15% of active users
- ☐ Zero performance regressions in CI/CD

Implementation Timeline:

- **Month 1:** Bundle optimization and code splitting
- **Month 2:** PWA implementation and image optimization
- **Month 3:** SSR exploration and monitoring setup

Technical Stack Additions:

- **Performance:** Web Workers, Workbox, Partytown
- **Monitoring:** Sentry Performance, DataDog RUM
- **Optimization:** SWC compiler, Million.js
- **Testing:** Lighthouse CI, WebPageTest API

Development Phases

Phase 1A: Foundation and Core Features (Days 1-60)

Primary Objective: Launch with marketing foundation, secure family onboarding, and comprehensive asset management.

Month 1 (Days 1-30): Infrastructure and Marketing Foundation

Development Priorities:

- **Technical Foundation**

- Development, staging, and production environments
- Core database schema implementation
- CI/CD pipelines with automated testing
- AWS infrastructure setup (S3, CloudFront, RDS)
- Authentication and authorization systems
- Monitoring and logging infrastructure
- **Builder.io Marketing Platform**
 - [Builder.io](#) integration and configuration
 - Landing page templates and components
 - A/B testing framework setup
 - Analytics and tracking implementation
 - SEO optimization and performance tuning
 - Marketing team training and handoff
- **Core Backend Services**
 - User management and authentication APIs
 - FFC creation and management endpoints
 - SMS/Email service integrations (Twilio, SendGrid)
 - Permission system implementation
 - Audit logging system
 - Database optimization and indexing

Month 2 (Days 31-60): Complete Onboarding and Asset Management

Development Priorities:

- **Secure Onboarding System**
 - User registration and email verification
 - FFC creation wizard
 - Dual-channel invitation system
 - Mandatory phone verification
 - Owner approval workflow
 - Role assignment and management
- **Comprehensive Asset Management**
 - All 13 asset categories implementation
 - Asset-persona ownership system
 - Individual asset permissions
 - Category-specific forms and validation
 - File upload and document management
 - Asset dashboard and overview

- **HEI Integration**

- API integration for loan data
- Read-only display in asset system
- Data synchronization and updates
- Integration with asset permission system

Phase 1A Success Criteria:

- Marketing team publishing content independently
- 90%+ user registration completion rate
- Dual-channel verification >90% success rate
- All 13 asset categories functional
- Asset permission system working correctly
- Zero security breaches
- Landing page conversion rate >2%

Phase 1B: Enhancement and Optimization (Days 61-120)

Primary Objective: Optimize user experience, add advanced features, and prepare for scale.

Development Priorities:

- Advanced search and filtering
- Bulk asset operations
- Enhanced mobile experience
- Performance optimization
- Security enhancements (2FA)
- Advanced analytics and reporting
- User feedback integration
- International expansion features

Conclusion

This Product Requirements Document V1.2 provides a comprehensive foundation for building the Forward Inheritance SaaS platform with a focus on marketing independence, secure family onboarding, and comprehensive asset management. The refined Phase 1A approach balances immediate market needs with long-term strategic vision, ensuring rapid time-to-market while maintaining technical excellence and user security.

The innovative family-first collaboration features combined with individual asset control and enhanced security positioning Forward to capture significant market share in the estate planning and wealth transfer space. Success depends on flawless execution of the marketing foundation and secure onboarding flow while building toward comprehensive asset management capabilities.

Document Version: 1.2

Last Updated: December 30, 2024

Next Review: January 15, 2025

Changes from V1.1: Added [Builder.io](#) marketing foundation, enhanced security with dual-channel verification, clarified asset permission model, removed chat system from MVP