# Product Requirements Document (PRD)

## Conveyor Marketing Group - Client Portal Platform MVP

### Version 1.0 | August 23, 2025

## Table of Contents

1. [Executive Summary](#1-executive-summary)
2. [System Architecture](#2-system-architecture)
3. [User Types & Permissions](#3-user-types--permissions)
4. [Core Features - MVP](#4-core-features---mvp)
5. [Technical Specifications](#5-technical-specifications)
6. [User Journeys](#6-user-journeys)
7. [Success Metrics](#7-success-metrics)
8. [Implementation Timeline](#8-implementation-timeline)
9. [Risk Mitigation](#9-risk-mitigation)
10. [Future Roadmap](#10-future-roadmap-post-mvp)
11. [Development Guidelines](#11-development-guidelines)
12. [Approval & Sign-off](#12-approval--sign-off)

## 1. Executive Summary

### Product Vision

A multi-tenant client portal platform that serves as a centralized hub for professional services delivery, starting with Conveyor Marketing Group's internal use and eventually scaling to a white-label SaaS platform for other agencies.

### MVP Goals

* **Launch Date:** October 31, 2025
* **Pilot Clients:** 3-5 clients
* **Primary Objectives:**
  + Establish secure multi-tenant architecture
  + Centralize client communications and deliverables
  + Prove the service-as-a-subscription model
  + Reduce operational overhead by 30%

### Key Business Drivers

* **Current Challenge:** Fragmented client experience across multiple tools
* **Solution:** Unified portal combining all client touchpoints
* **Opportunity:** White-label platform for professional services industry
* **Investment Stage:** Self-funded MVP to demonstrate model for future funding

## 2. System Architecture

### 2.1 Company Hierarchy

Platform Instance (Conveyor)

├── Owner Company (Conveyor Marketing Group)

│ ├── Partner Companies (Freelancers/Agencies)

│ │ └── Partner Users

│ ├── Client Companies

│ │ ├── Sub-Companies (Divisions/Departments)

│ │ └── Client Users

│ └── Internal Users (CMG Team)

### 2.2 Data Isolation Model

**Core Principle:** Complete data isolation between company types with project-based access control.

#### Data Access Matrix

| **Entity** | **Access Scope** | **Restrictions** |
| --- | --- | --- |
| **Owner Company** | All platform data | None |
| **Partner Company** | Assigned clients/projects only | Cannot see other partners' data |
| **Partner User** | Specific assigned projects | Limited to individual assignments |
| **Client Company** | Own company data only | Cannot see other clients |
| **Client User** | Based on role (Editor/Viewer) | Project-specific permissions |

### 2.3 Project-Based Access Control

Project Access Hierarchy:

1. Owner Company assigns Partner Companies to Client Projects

2. Partner Companies assign specific users to projects

3. Client Companies control their users' access levels

4. All access is audited and logged



## 3. User Types & Permissions

### 3.1 Owner Company Users (Conveyor)

| **Role** | **Description** | **Key Permissions** | **System Access** |
| --- | --- | --- | --- |
| **Owner** | Platform administrator | • Full system configuration  • Billing management  • Platform settings | All modules |
| **Admin** | Operations manager | • User management  • Client/partner setup  • Service configuration | All client data |
| **Client Services** | Account managers | • Client relationship management  • Project assignment  • Reporting access | Assigned accounts |
| **Specialty Skills** | Delivery team | • Project execution  • Deliverable uploads  • Client communication | Assigned projects |

### 3.2 Partner Company Users

| **Role** | **Description** | **Key Permissions** | **System Access** |
| --- | --- | --- | --- |
| **Partner Admin** | Partner company manager | • Manage team users  • View all assigned projects  • Resource allocation | Company projects |
| **Partner Contributor** | Service delivery | • Edit project deliverables  • Upload files  • Update status | Assigned projects |
| **Partner Viewer** | Stakeholder | • View project status  • Read deliverables  • Access reports | Assigned projects |

### 3.3 Client Company Users

| **Role** | **Access Level** | **Capabilities** | **Available Tags** |
| --- | --- | --- | --- |
| **Editor** | Read/Write | • View all company data  • Comment on deliverables  • Approve/reject work  • Request changes | • Champion  • Primary Contact  • Team Member |
| **Viewer** | Read Only | • View reports  • Access deliverables  • See project status | • Executive Sponsor  • Stakeholder  • Observer |

**Note:** Tags are metadata labels for CMG team visibility and communication targeting, not permission levels.

## 4. Core Features - MVP

### 4.1 Phase 1: Foundation (Weeks 1-4)

#### User & Company Management

* **Company Creation**
  + Company types: Owner, Partner, Client, Sub-company
  + Custom branding per company (logo, colors)
  + Hierarchical relationship mapping
  + Settings and preferences
* **User Management**
  + CRUD operations with role assignment
  + Bulk user import via CSV
  + User profile management
  + Password policies and 2FA
* **Authentication System**
  + Email/password authentication
  + Google SSO integration
  + Magic links for first-time registration
  + Session management with configurable timeout
  + Remember me functionality
* **Access Request Workflow**
  + Client-initiated access requests
  + CMG approval queue
  + Automated notifications
  + Audit trail of all requests

#### Security & Compliance

* **Data Isolation**
  + Row-level security by company/project
  + Encrypted data at rest and in transit
  + Secure API endpoints
  + Regular security audits
* **GDPR Compliance**
  + Consent management for data connections
  + Data portability (export functionality)
  + Right to deletion workflows
  + Privacy policy acceptance tracking
* **AI Usage Consent**
  + Explicit opt-in for AI features
  + Granular control per data type
  + Clear data usage explanations
  + Opt-out mechanisms

### 4.2 Phase 2: Core Services (Weeks 5-8)

#### Service Module 1: Digital Audits

**Features:**

* HTML report hosting and display
* Secure, time-limited access for prospects
* Permanent access for converted clients
* Interactive navigation within reports
* Print and PDF export functionality

**Tracking:**

* Page view analytics
* Time spent per section
* Download tracking
* Engagement scoring

**Workflow:**

* Draft → Review → Published → Archived
* Version control for updates
* Comment and annotation system
* Approval workflows

#### Service Module 2: Monthly Reporting

**Dashboard Components:**

* Client-selected KPIs
* Goal tracking and progress
* Year-over-year comparisons
* Custom date ranges
* Executive summary generation

**Data Integration:**

* Google Analytics 4
* Google Ads
* Meta Ads Manager
* LinkedIn Campaign Manager
* HubSpot CRM
* Custom data sources via API

**Automation:**

* Scheduled data synchronization
* Automated alerts for anomalies
* Monthly report generation
* Email digest delivery

#### Service Module 3: Brand Monitoring

**Monitoring Capabilities:**

* Real-time news aggregation
* Social media mentions
* Competitor tracking
* Industry trend analysis
* Sentiment analysis

**Configuration:**

* Custom keyword lists
* Competitor definitions
* Alert thresholds
* Notification preferences
* Report frequency

**Data Sources:**

* Brand24 integration
* Meltwater feed
* Google Alerts
* Social media APIs
* RSS feed aggregation

### 4.3 Phase 3: Integration Layer (Weeks 9-10)

#### Project Management Integration

* **ClickUp Integration**
  + Embedded project views
  + Task status synchronization
  + Milestone tracking
  + Document linking

#### Analytics Integration

* **Marketing Analytics**
  + GA4 real-time data
  + Goal conversion tracking
  + Traffic source analysis
  + User behavior flow

#### CRM Integration

* **HubSpot Connection**
  + Contact synchronization
  + Deal pipeline visibility
  + Activity timeline
  + Email engagement metrics

## 5. Technical Specifications

### 5.1 Technology Stack

| **Component** | **Technology** | **Rationale** |
| --- | --- | --- |
| **Frontend** | Replit with React/Next.js | Developer preference, rapid deployment, built-in hosting |
| **Backend** | Node.js + Express.js | JavaScript ecosystem, extensive libraries, async handling |
| **Database** | PostgreSQL | Relational integrity, row-level security, JSON support |
| **Authentication** | Supabase Auth or Auth0 | Managed service, SSO support, security compliance |
| **File Storage** | AWS S3 or Supabase Storage | Scalability, CDN integration, cost-effective |
| **Queue System** | Bull/Redis | Background job processing, scheduled tasks |
| **Integration** | Zapier + Custom APIs | No-code for simple integrations, custom for complex |
| **Analytics** | Google Analytics 4 | Free tier sufficient, familiar interface |
| **Monitoring** | Sentry + Uptime Robot | Error tracking, performance monitoring, uptime alerts |
| **Email** | SendGrid or Postmark | Transactional emails, high deliverability |

### 5.2 Database Schema

-- Companies and Hierarchy

CREATE TABLE companies (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

type VARCHAR(20) CHECK (type IN ('owner', 'partner', 'client', 'sub')),

parent\_id UUID REFERENCES companies(id),

name VARCHAR(255) NOT NULL,

domain VARCHAR(255),

logo\_url TEXT,

primary\_color VARCHAR(7),

settings JSONB DEFAULT '{}',

created\_at TIMESTAMP DEFAULT NOW(),

updated\_at TIMESTAMP DEFAULT NOW()

);

-- Users

CREATE TABLE users (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

company\_id UUID REFERENCES companies(id) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

password\_hash VARCHAR(255),

role VARCHAR(50) NOT NULL,

tags TEXT[],

google\_id VARCHAR(255),

is\_active BOOLEAN DEFAULT true,

last\_login TIMESTAMP,

created\_at TIMESTAMP DEFAULT NOW(),

updated\_at TIMESTAMP DEFAULT NOW()

);

-- Projects

CREATE TABLE projects (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

client\_company\_id UUID REFERENCES companies(id) NOT NULL,

name VARCHAR(255) NOT NULL,

description TEXT,

status VARCHAR(50) DEFAULT 'active',

created\_by UUID REFERENCES users(id),

settings JSONB DEFAULT '{}',

start\_date DATE,

end\_date DATE,

created\_at TIMESTAMP DEFAULT NOW(),

updated\_at TIMESTAMP DEFAULT NOW()

);

-- Project Access Control

CREATE TABLE project\_access (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

project\_id UUID REFERENCES projects(id) NOT NULL,

company\_id UUID REFERENCES companies(id),

user\_id UUID REFERENCES users(id),

access\_level VARCHAR(20) CHECK (access\_level IN ('edit', 'view')),

granted\_by UUID REFERENCES users(id),

granted\_at TIMESTAMP DEFAULT NOW(),

UNIQUE(project\_id, user\_id)

);

-- Services

CREATE TABLE services (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

name VARCHAR(255) NOT NULL,

type VARCHAR(50) NOT NULL,

description TEXT,

is\_active BOOLEAN DEFAULT true,

settings JSONB DEFAULT '{}',

created\_at TIMESTAMP DEFAULT NOW()

);

-- Client Services (which services are enabled per client)

CREATE TABLE client\_services (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

client\_company\_id UUID REFERENCES companies(id) NOT NULL,

service\_id UUID REFERENCES services(id) NOT NULL,

is\_enabled BOOLEAN DEFAULT true,

configuration JSONB DEFAULT '{}',

enabled\_by UUID REFERENCES users(id),

enabled\_at TIMESTAMP DEFAULT NOW(),

UNIQUE(client\_company\_id, service\_id)

);

-- Audit Log

CREATE TABLE audit\_log (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

user\_id UUID REFERENCES users(id),

action VARCHAR(100) NOT NULL,

resource\_type VARCHAR(50),

resource\_id UUID,

details JSONB,

ip\_address INET,

user\_agent TEXT,

created\_at TIMESTAMP DEFAULT NOW()

);

-- Consent Management

CREATE TABLE consent\_log (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

company\_id UUID REFERENCES companies(id) NOT NULL,

consent\_type VARCHAR(50) NOT NULL,

status BOOLEAN NOT NULL,

granted\_by UUID REFERENCES users(id),

ip\_address INET,

details JSONB,

created\_at TIMESTAMP DEFAULT NOW()

);

-- Digital Audits

CREATE TABLE digital\_audits (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

client\_company\_id UUID REFERENCES companies(id) NOT NULL,

title VARCHAR(255) NOT NULL,

html\_content TEXT,

status VARCHAR(20) DEFAULT 'draft',

access\_type VARCHAR(20) DEFAULT 'permanent',

access\_expires\_at TIMESTAMP,

created\_by UUID REFERENCES users(id),

published\_at TIMESTAMP,

created\_at TIMESTAMP DEFAULT NOW(),

updated\_at TIMESTAMP DEFAULT NOW()

);

-- Indexes for Performance

CREATE INDEX idx\_companies\_parent ON companies(parent\_id);

CREATE INDEX idx\_users\_company ON users(company\_id);

CREATE INDEX idx\_projects\_client ON projects(client\_company\_id);

CREATE INDEX idx\_project\_access\_project ON project\_access(project\_id);

CREATE INDEX idx\_audit\_log\_user ON audit\_log(user\_id);

CREATE INDEX idx\_audit\_log\_created ON audit\_log(created\_at);

### 5.3 API Endpoints

// Authentication

POST /api/auth/login

POST /api/auth/logout

POST /api/auth/refresh

POST /api/auth/google

POST /api/auth/magic-link

// Companies

GET /api/companies

POST /api/companies

GET /api/companies/:id

PUT /api/companies/:id

DELETE /api/companies/:id

// Users

GET /api/users

POST /api/users

GET /api/users/:id

PUT /api/users/:id

DELETE /api/users/:id

POST /api/users/invite

POST /api/users/bulk-import

// Projects

GET /api/projects

POST /api/projects

GET /api/projects/:id

PUT /api/projects/:id

DELETE /api/projects/:id

POST /api/projects/:id/access

DELETE /api/projects/:id/access/:userId

// Services

GET /api/services

POST /api/services

GET /api/services/:id

PUT /api/services/:id

POST /api/clients/:id/services

DELETE /api/clients/:id/services/:serviceId

// Digital Audits

GET /api/audits

POST /api/audits

GET /api/audits/:id

PUT /api/audits/:id

POST /api/audits/:id/publish

GET /api/audits/:id/analytics

// Reports

GET /api/reports/dashboard

GET /api/reports/kpis

POST /api/reports/generate

GET /api/reports/:id

// Integrations

POST /api/integrations/ga4/connect

GET /api/integrations/ga4/data

POST /api/integrations/hubspot/sync

GET /api/integrations/clickup/projects

### 5.4 Security Implementation

// Row-Level Security Implementation

class SecurityService {

async canAccessProject(userId, projectId) {

const user = await User.findById(userId).include('company');

const project = await Project.findById(projectId);

// Owner company users can access all projects

if (user.company.type === 'owner') {

return true;

}

// Check explicit project access

const access = await ProjectAccess.findOne({

where: { userId, projectId }

});

if (access) {

return access.accessLevel;

}

// Partner company users need explicit access

if (user.company.type === 'partner') {

const companyAccess = await ProjectAccess.findOne({

where: {

companyId: user.companyId,

projectId

}

});

return companyAccess ? companyAccess.accessLevel : false;

}

// Client users can access their own company's projects

if (user.company.type === 'client') {

return project.clientCompanyId === user.companyId;

}

return false;

}

async filterAccessibleProjects(userId, projects) {

const accessible = [];

for (const project of projects) {

if (await this.canAccessProject(userId, project.id)) {

accessible.push(project);

}

}

return accessible;

}

}

// Middleware for API Protection

const requireAuth = async (req, res, next) => {

const token = req.headers.authorization?.split(' ')[1];

if (!token) {

return res.status(401).json({ error: 'No token provided' });

}

try {

const decoded = jwt.verify(token, process.env.JWT\_SECRET);

req.user = await User.findById(decoded.userId);

if (!req.user || !req.user.isActive) {

return res.status(401).json({ error: 'Invalid user' });

}

// Log the access

await AuditLog.create({

userId: req.user.id,

action: `${req.method} ${req.path}`,

ipAddress: req.ip,

userAgent: req.headers['user-agent']

});

next();

} catch (error) {

return res.status(401).json({ error: 'Invalid token' });

}

};



## 6. User Journeys

### 6.1 Prospect Journey

1. Discovery

├── Receives email invitation with unique link

├── Link includes time-limited token

└── Redirected to registration page

2. Registration

├── Enter basic information (name, company, role)

├── Create password or use Google SSO

├── Accept terms and privacy policy

└── Verify email address

3. First Experience

├── Guided tour of portal interface

├── View digital audit report

├── Interactive navigation through findings

└── Download PDF version option

4. Engagement

├── Schedule consultation directly in portal

├── Submit questions or comments

├── Share report with team (request access)

└── Receive follow-up communications

5. Conversion

├── Accept proposal in portal

├── Account upgraded to client status

├── Additional services activated

└── Full portal features unlocked

### 6.2 Client Onboarding Journey

1. Account Setup

├── CMG creates client company profile

├── Configures enabled services

├── Sets up KPI dashboard

└── Imports historical data

2. User Provisioning

├── Primary contact added as Editor

├── Welcome email with login credentials

├── Additional stakeholders invited

└── Role assignments configured

3. Initial Login

├── Password setup (if not SSO)

├── Profile completion

├── Notification preferences

└── Dashboard personalization

4. Team Expansion

├── Request additional user access

├── CMG approves and provisions

├── New users receive invitations

└── Permissions inherited from company

5. Active Usage

├── Daily dashboard monitoring

├── Weekly report reviews

├── Monthly strategy sessions

└── Ongoing collaboration

### 6.3 Partner Collaboration Journey

1. Partner Onboarding

├── CMG creates partner company

├── Partner admin designated

├── Access credentials provided

└── NDA/agreement acceptance

2. Project Assignment

├── CMG assigns partner to client project

├── Specific deliverables defined

├── Timeline established

└── Access permissions set

3. Team Configuration

├── Partner admin adds team members

├── Assigns users to specific projects

├── Sets individual permissions

└── Configures notifications

4. Project Execution

├── Access project workspace

├── View requirements and briefs

├── Upload deliverables

└── Update status/progress

5. Client Collaboration

├── Receive client feedback

├── Iterate on deliverables

├── Participate in discussions

└── Final approval workflow



## 7. Success Metrics

### 7.1 MVP Success Criteria

| **Category** | **Metric** | **Target** | **Measurement Method** |
| --- | --- | --- | --- |
| **Adoption** | Client Login Rate | 80% | Unique logins / invited clients |
| **Engagement** | Weekly Active Users | 60% | Weekly unique logins / total users |
| **Engagement** | Session Duration | >5 minutes | GA4 average session time |
| **Usage** | Features Used | 3+ per session | GA4 event tracking |
| **Performance** | Page Load Time | <2 seconds | Performance monitoring |
| **Reliability** | Uptime | 99.9% | Uptime monitoring service |
| **Security** | Security Incidents | Zero | Incident tracking system |
| **Satisfaction** | Client NPS | 8+ | Quarterly survey |
| **Efficiency** | Support Tickets | <5/week | Help desk tracking |
| **Growth** | Service Adoption | 20% increase | New services activated |

### 7.2 Business Impact Metrics

| **Metric** | **Baseline** | **Target** | **Timeline** |
| --- | --- | --- | --- |
| **Email Volume** | 100/week/client | 70/week/client | 3 months |
| **Client Onboarding Time** | 2 weeks | 1 week | 2 months |
| **Report Delivery Time** | 48 hours | 2 hours | Immediate |
| **Client Retention** | 85% | 95% | 6 months |
| **Upsell Rate** | 15% | 25% | 4 months |
| **Team Productivity** | 5 clients/manager | 8 clients/manager | 3 months |
| **Client Satisfaction** | 7.5 NPS | 9.0 NPS | 3 months |

### 7.3 Technical Performance Metrics

| **Metric** | **Target** | **Alert Threshold** |
| --- | --- | --- |
| **API Response Time** | <200ms | >500ms |
| **Database Query Time** | <50ms | >100ms |
| **Error Rate** | <0.1% | >1% |
| **Concurrent Users** | 100+ | System degradation |
| **Data Sync Frequency** | Every 6 hours | Failure after 12 hours |

## 8. Implementation Timeline

### Week 1-2: Foundation Setup

* **Week 1**
  + Development environment setup
  + Database schema implementation
  + Authentication system setup
  + Basic user model and CRUD
* **Week 2**
  + Company hierarchy implementation
  + Role-based access control
  + Security middleware
  + API endpoint structure

### Week 3-4: Core Platform

* **Week 3**
  + User interface framework
  + Navigation and routing
  + Company management UI
  + User management UI
* **Week 4**
  + Project management structure
  + Access control implementation
  + Audit logging system
  + Security testing

### Week 5-6: Digital Audit Module

* **Week 5**
  + Audit report data model
  + HTML content storage
  + Report viewer component
  + Access control for prospects
* **Week 6**
  + Analytics tracking
  + Engagement metrics
  + Export functionality
  + Publishing workflow

### Week 7: Reporting Dashboard

* **Day 1-2:** KPI framework and data model
* **Day 3-4:** Dashboard UI components
* **Day 5:** Data visualization libraries
* **Day 6-7:** Integration with data sources

### Week 8: Brand Monitoring

* **Day 1-2:** Integration with Brand24/Meltwater
* **Day 3-4:** Alert system and notifications
* **Day 5:** News feed UI
* **Day 6-7:** Configuration interface

### Week 9: Integration & Testing

* **Day 1-2:** ClickUp integration
* **Day 3:** GA4 and analytics setup
* **Day 4:** HubSpot CRM connection
* **Day 5-7:** Comprehensive testing

### Week 10: Launch Preparation

* **Day 1-2:** Bug fixes and optimization
* **Day 3:** Documentation completion
* **Day 4:** Client onboarding preparation
* **Day 5:** Training materials
* **Day 6-7:** Soft launch with first client

## 9. Risk Mitigation

### Technical Risks

| **Risk** | **Probability** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| **Security Breach** | Low | Critical | • Penetration testing  • Security audit  • Encryption at rest/transit  • Regular updates |
| **Performance Issues** | Medium | High | • Load testing  • Caching strategy  • CDN implementation  • Database optimization |
| **Integration Failures** | Medium | Medium | • Fallback mechanisms  • Manual processes  • Retry logic  • Error notifications |
| **Data Loss** | Low | Critical | • Automated backups  • Replication  • Point-in-time recovery  • Disaster recovery plan |
| **Scalability Issues** | Low | Medium | • Horizontal scaling plan  • Microservices architecture  • Queue systems  • Load balancing |

### Business Risks

| **Risk** | **Probability** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| **Low Client Adoption** | Medium | High | • Onboarding support  • Training videos  • Incentive program  • Feedback loops |
| **Scope Creep** | High | Medium | • Strict MVP definition  • Change control process  • Feature freeze  • Phase 2 backlog |
| **Resource Constraints** | Medium | Medium | • Partner augmentation  • Prioritization matrix  • Phased delivery  • Automation focus |
| **Competitive Pressure** | Low | Medium | • Rapid MVP delivery  • Unique value props  • Client lock-in  • Continuous innovation |

### Operational Risks

| **Risk** | **Probability** | **Impact** | **Mitigation Strategy** |
| --- | --- | --- | --- |
| **Team Burnout** | Medium | High | • Realistic timelines  • Resource planning  • Automation priority  • Work distribution |
| **Communication Gaps** | Medium | Medium | • Daily standups  • Clear documentation  • Regular demos  • Stakeholder updates |
| **Quality Issues** | Low | High | • Code reviews  • Automated testing  • QA process  • User acceptance testing |

## 10. Future Roadmap (Post-MVP)

### Phase 2: Q1 2026 (Months 1-3 Post-Launch)

* **Enhanced Analytics**
  + Custom report builder
  + Predictive analytics
  + Benchmarking tools
  + ROI calculators
* **Workflow Automation**
  + Automated task creation
  + Smart notifications
  + Workflow templates
  + Process automation
* **Advanced Integrations**
  + Slack integration
  + Microsoft Teams
  + Salesforce CRM
  + Custom API webhooks

### Phase 3: Q2 2026 (Months 4-6 Post-Launch)

* **Mobile Applications**
  + iOS native app
  + Android native app
  + Push notifications
  + Offline capabilities
* **AI-Powered Features**
  + Smart insights
  + Anomaly detection
  + Content generation
  + Chatbot support
* **Collaboration Tools**
  + Real-time commenting
  + Document collaboration
  + Video conferencing
  + Screen sharing

### Phase 4: Q3-Q4 2026 (Scale & White-Label)

* **White-Label Platform**
  + Multi-instance architecture
  + Custom domains
  + Branding customization
  + Independent databases
* **Advanced Business Tools**
  + Proposal generator
  + Contract management
  + Invoice processing
  + Payment integration
* **Enterprise Features**
  + SSO providers
  + Advanced permissions
  + Compliance tools
  + API marketplace

### Long-Term Vision (2027+)

* **Market Expansion**
  + Industry-specific versions
  + International support
  + Multi-language
  + Regional compliance
* **Platform Ecosystem**
  + App marketplace
  + Developer APIs
  + Partner integrations
  + Community features

## 11. Development Guidelines

### 11.1 Coding Standards

// File Structure

/src

/components // React components

/pages // Route pages

/services // Business logic

/utils // Helper functions

/hooks // Custom React hooks

/contexts // React contexts

/styles // CSS/styled-components

/constants // App constants

/types // TypeScript types

// Naming Conventions

- Components: PascalCase (UserDashboard.jsx)

- Functions: camelCase (getUserData())

- Constants: UPPER\_SNAKE\_CASE (MAX\_RETRIES)

- Files: kebab-case (user-service.js)

// Code Quality Rules

- ESLint configuration required

- Prettier for formatting

- 80% test coverage minimum

- No console.logs in production

- Error boundaries for components

- Proper TypeScript typing

### 11.2 Security Guidelines

1. **Authentication**
   * JWT tokens with expiration
   * Refresh token rotation
   * Session invalidation on logout
   * Rate limiting on auth endpoints
2. **Authorization**
   * Check permissions on every request
   * Validate user context
   * Audit sensitive operations
   * Principle of least privilege
3. **Data Protection**
   * Encrypt sensitive data
   * Sanitize all inputs
   * Parameterized queries
   * XSS protection
4. **API Security**
   * HTTPS only
   * CORS configuration
   * API rate limiting
   * Request validation

### 11.3 Testing Requirements

// Unit Tests (Jest)

describe('UserService', () => {

test('should create user with valid data', async () => {

const user = await UserService.create(validUserData);

expect(user).toHaveProperty('id');

expect(user.email).toBe(validUserData.email);

});

});

// Integration Tests

describe('API Endpoints', () => {

test('POST /api/users requires authentication', async () => {

const response = await request(app)

.post('/api/users')

.send(userData);

expect(response.status).toBe(401);

});

});

// E2E Tests (Cypress)

describe('User Login Flow', () => {

it('should login successfully with valid credentials', () => {

cy.visit('/login');

cy.get('[data-testid="email"]').type('user@example.com');

cy.get('[data-testid="password"]').type('password123');

cy.get('[data-testid="submit"]').click();

cy.url().should('include', '/dashboard');

});

});

### 11.4 Documentation Standards

1. **Code Documentation**

/\*\*

\* Creates a new project for a client company

\* @param {Object} projectData - Project configuration

\* @param {string} projectData.name - Project name

\* @param {string} projectData.clientId - Client company ID

\* @param {Array} projectData.partners - Partner companies

\* @returns {Promise<Project>} Created project object

\* @throws {ValidationError} Invalid project data

\* @throws {AuthorizationError} User lacks permission

\*/

async function createProject(projectData) {

// Implementation

}



1. **API Documentation**
   * OpenAPI/Swagger specification
   * Postman collection
   * Example requests/responses
   * Error code definitions
2. **User Documentation**
   * Getting started guide
   * Feature tutorials
   * Video walkthroughs
   * FAQ section

## 12. Approval & Sign-off

### Stakeholder Approval

| **Stakeholder** | **Role** | **Department** | **Approval Date** | **Signature** |
| --- | --- | --- | --- | --- |
| Jason Johnston | CEO | Executive | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_ |
| [Lead Developer] | Technical Lead | Engineering | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_ |
| [Client Services Lead] | Operations Manager | Client Services | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_ |
| [QA Lead] | Quality Assurance | Engineering | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_ |

### Document Revision History

| **Version** | **Date** | **Author** | **Changes** |
| --- | --- | --- | --- |
| 1.0 | Aug 23, 2025 | Initial Draft | Complete PRD creation |
|  |  |  |  |
|  |  |  |  |

### Acceptance Criteria

* All stakeholders have reviewed and approved
* Technical architecture validated
* Security review completed
* Budget approved
* Timeline accepted
* Resources allocated
* Success metrics agreed upon

## Appendices

### Appendix A: Glossary

| **Term** | **Definition** |
| --- | --- |
| **Owner Company** | The agency that owns the platform instance (initially Conveyor) |
| **Partner Company** | External service providers working with the owner company |
| **Client Company** | End clients receiving services through the platform |
| **MVP** | Minimum Viable Product - initial feature set for launch |
| **KPI** | Key Performance Indicator - measurable business metrics |
| **SSO** | Single Sign-On - unified authentication method |
| **RLS** | Row-Level Security - database access control mechanism |
| **GDPR** | General Data Protection Regulation - EU privacy law |

### Appendix B: Reference Links

* [Figma Designs] - *To be added*
* [Technical Architecture Diagram] - *To be added*
* [API Documentation] - *To be added*
* [Security Audit Checklist] - *To be added*
* [GDPR Compliance Matrix] - *To be added*
* [Testing Strategy Document] - *To be added*

### Appendix C: Example User Scenarios

**Scenario 1: Agency Account Manager**

* Logs in Monday morning
* Checks weekend alerts across 5 clients
* Reviews KPI dashboards for anomalies
* Schedules client check-in calls
* Assigns tasks to delivery team
* Updates project status

**Scenario 2: Client Marketing Director**

* Receives mobile alert about traffic spike
* Logs in to investigate cause
* Reviews real-time analytics
* Downloads report for executive meeting
* Requests additional analysis from agency
* Approves new campaign creative

**Scenario 3: Partner Developer**

* Receives project assignment notification
* Accesses project requirements
* Reviews client brand guidelines
* Uploads development milestones
* Responds to client feedback
* Marks project complete

### Appendix D: Data Privacy & Compliance

**Data Retention Policy**

* Active client data: Retained indefinitely
* Inactive client data: 2 years
* Prospect data: 1 year
* Audit logs: 3 years
* Analytics data: 13 months

**Compliance Requirements**

* GDPR (European Union)
* CCPA (California)
* PIPEDA (Canada)
* Industry-specific regulations

**Security Certifications (Future)**

* SOC 2 Type II
* ISO 27001
* HIPAA (if healthcare clients)

## Contact Information

**Project Owner**

* Name: Jason Johnston
* Role: CEO, Conveyor Marketing Group
* Email: [jason@conveyormg.com](mailto:jason@conveyormg.com)

**Technical Lead**

* Name: [To be assigned]
* Role: Lead Developer
* Email: [To be assigned]

**Project Manager**

* Name: [To be assigned]
* Role: Project Coordinator
* Email: [To be assigned]

*This document is confidential and proprietary to Conveyor Marketing Group. Distribution is limited to authorized personnel only.*

**Document Version:** 1.0  
**Last Updated:** August 23, 2025  
**Next Review:** TBD