Bowman Insurance Agency

Digital Insurance Platform

Technical Specifications Document

Document Version:	1.0
Date:	January 26, 2026
Project Status:	Planning Phase
Classification:	Internal - Confidential
Prepared By:	Bowman Insurance Agency - Technical Team
Target Audience:	Development Team, Project Managers, Stakeholders

Table of Contents

1. Executive Summary

Bowman Insurance Agency is embarking on a digital transformation initiative to develop a comprehensive insurance marketplace and management platform. This platform will enable customers to browse, compare, purchase, and manage insurance policies from multiple insurance companies through a single unified digital interface.

1.1 Project Objectives

Primary Objective: Build a scalable, secure, and user-friendly digital platform that serves as a one-stop solution for all insurance needs in Kenya.

- Key deliverables include:
    - Public-facing insurance marketplace (e-commerce style)
    - Customer self-service dashboard
    - Comprehensive admin panel for agency staff
    - Automated policy lifecycle management (workflow engine)
    - Multi-channel payment processing (M-Pesa, cards)
    - End-to-end claims management system
    - Real-time analytics and reporting
    - Mobile-responsive web application

1.2 Success Metrics (Year 1)

Metric	Target
Registered Customers	10,000+
Active Policies	5,000+
Premiums Processed	KES 100M+
Customer Satisfaction	80%+
Digital Policy Issuance	70%+ (vs manual)
Avg Claim Processing Time	<48 hours

1.3 Strategic Benefits

- **Operational Efficiency:** Reduce manual policy processing by 70%, automate repetitive tasks
- **Customer Experience:** Instant policy issuance, 24/7 self-service, transparent pricing
- **Market Competitiveness:** First mover advantage in digital insurance in Kenya
- **Scalability:** Handle 10x customer growth without proportional staff increase
- **Data-Driven Insights:** Real-time analytics for better business decisions
- **Revenue Growth:** New distribution channel, reduced acquisition costs

2. Project Overview

2.1 Problem Statement

The current insurance landscape in Kenya presents several challenges:

- Customers must visit multiple providers to compare policies
- Manual, paper-based processes lead to delays (2-5 days for policy issuance)
- Limited transparency in pricing and coverage options
- Difficult claims process with minimal status visibility
- No centralized platform to manage multiple policies
- High operational costs due to manual workflows

2.2 Proposed Solution

A comprehensive digital platform that aggregates insurance products from multiple companies, provides instant quotes and policy issuance, enables self-service policy management, and streamlines the entire insurance lifecycle from purchase to claims settlement.

2.3 Target Users

User Type	Description	Key Needs
Customers (End Users)	Individuals & businesses purchasing insurance	Easy comparison, instant quotes, self-service, mobile access
Agency Staff (Internal)	Underwriters, policy managers, support staff, admin	Efficient workflows, automation, analytics, user management
Insurance Companies (Partners)	Britam, Old Mutual, Jubilee, CIC, APA, etc.	Digital distribution, performance insights, reduced costs

2.4 Scope

2.4.1 In Scope

- Development of web-based platform (frontend + backend)
- Integration with M-Pesa and Paystack payment gateways
- Integration with SMS/Email/WhatsApp notification services
- Policy management for 10 insurance categories (Motor, Medical, Life, Home, Travel, etc.)
- Customer dashboard with policy management, payments, claims, documents
- Admin panel with user management, policy management, analytics
- Custom workflow engine for policy lifecycle management
- Claims filing and tracking system

- Document generation (certificates, receipts) and storage
- Role-based access control (RBAC)
- Responsive web design (mobile, tablet, desktop)
- Initial deployment on cloud infrastructure
- User training and documentation

2.4.2 Out of Scope (Future Phases)

- Native mobile apps (iOS/Android) - Phase 2
- Agent/broker portal - Phase 2
- AI-powered chatbot - Phase 3
- Blockchain-based claims verification - Future
- IoT device integration (telematics for motor) - Future
- Integration with government databases (NTSA, KRA) - Phase 2
- White-label solution for other agencies - Future

3. System Architecture

3.1 High-Level Architecture

The platform follows a modern three-tier architecture with clear separation of concerns:

Layer	Components
Presentation Layer	Next.js (React) - Server-side rendering, Client-side routing, Progressive Web App
Application Layer	Django REST Framework - RESTful APIs, Business logic, Authentication/Authorization
Data Layer	PostgreSQL (primary database), Redis (cache/sessions), AWS S3 (file storage)

3.2 Architecture Diagram

Architecture flow:

User → CloudFlare CDN → Load Balancer → Next.js Instances → API Gateway → Django API Instances → Database/Cache/Storage

3.3 Component Breakdown

- Frontend Application
    - Built with Next.js 14 (React 18)
    - TypeScript for type safety
    - Tailwind CSS for styling
    - Server-side rendering for SEO and performance
    - Client-side routing for smooth navigation
    - Progressive Web App capabilities
    - Responsive design (mobile-first approach)
- Backend API
    - Django 5.0 framework
    - Django REST Framework for API
    - JWT authentication (access + refresh tokens)
    - Role-based permissions
    - API versioning (v1)
    - Request/response serialization and validation
    - Rate limiting and throttling
    - Swagger/OpenAPI documentation
- Background Jobs

    - Celery task queue
    - Redis as message broker
    - Email sending (SendGrid/AWS SES)
    - SMS notifications (Africa's Talking)
    - Certificate generation (ReportLab)
    - Report generation
    - Payment reminders
    - Data cleanup tasks

- Database Layer

    - PostgreSQL 15+ (primary database)
    - ACID compliance for transactions
    - Foreign key constraints for data integrity
    - Full-text search capabilities
    - JSON fields for flexible policy data
    - Read replicas for scaling
    - Automated backups (daily)

- Caching Layer

    - Redis 7+ in-memory cache
    - API response caching (5-10 min TTL)
    - Session storage
    - Rate limiting counters
    - Job queue backend
    - Real-time pub/sub (optional)

- File Storage

    - AWS S3 for documents
    - Policy documents (PDFs)
    - Insurance certificates
    - User-uploaded documents (ID, KRA PIN)
    - Claim photos/videos
    - Profile pictures
    - Versioning enabled
    - Server-side encryption

3.4 Scalability Strategy

- **Horizontal Scaling:** Add more application instances behind load balancer. Stateless design with JWT authentication enables unlimited horizontal scaling.
- **Database Optimization:** Read replicas for read-heavy operations, connection pooling (PgBouncer), strategic indexing, query optimization, table partitioning for large tables.

- **Caching Strategy:** Multi-layer caching: Browser cache → CDN cache → Redis cache → Database. Target 80%+ cache hit rate to reduce database load.
- **Asynchronous Processing:** Move heavy operations to background jobs. Never block API requests. Email, SMS, PDF generation, reports all run async.
- **CDN Distribution:** CloudFlare CDN serves static assets from 300+ edge locations globally. Reduces latency and bandwidth costs.
- **Rate Limiting:** Per-user and per-IP rate limits prevent abuse and ensure fair usage. Protects system from overload.

4. Technology Stack

4.1 Stack Selection Rationale

The Django + React stack was selected based on three critical criteria:

- **Security:** Django provides built-in protection against OWASP Top 10 vulnerabilities. Used by banks and fintech companies globally. Essential for handling sensitive financial data.
- **Developer Availability:** React and Python/Django developers are abundant in Kenya. Reduces hiring risk and costs. Mid-level developers available at KES 120-200K/month.
- **Proven Scalability:** Instagram (2B users), Pinterest (450M users), and Spotify use Django. React is used by Facebook, Netflix, Airbnb. Both proven at massive scale.

4.2 Technology Matrix

Category	Technology	Version	Purpose
Frontend Framework	React	18.x	UI library
	Next.js	14.x	SSR framework
	TypeScript	5.x	Type safety
Frontend Styling	Tailwind CSS	3.x	Utility-first CSS
	shadcn/ui	Latest	Component library
Frontend State	TanStack Query	5.x	Data fetching & caching
	Zustand	Latest	Global state management
Backend	Python	3.11+	Programming

Language			language
Backend Framework	Django	5.0	Web framework
	Django REST Framework	3.14+	API toolkit
Task Queue	Celery	5.x	Background jobs
Database	PostgreSQL	15+	Primary database
Cache	Redis	7+	Cache & queue
File Storage	AWS S3	Latest	Document storage
CDN	CloudFlare	Latest	Content delivery
Payments	M-Pesa Daraja API	Latest	Mobile payments
	Paystack	Latest	Card payments
Notifications	SendGrid/AWS SES	Latest	Email
	Africa's Talking	Latest	SMS & WhatsApp

4.3 Development Tools

- **Version Control:** Git + GitHub - Source control, code reviews, issue tracking
- **CI/CD:** GitHub Actions - Automated testing, deployment
- **Containerization:** Docker + Docker Compose - Consistent environments
- **Testing:** Jest (frontend), Pytest (backend), Playwright (E2E)
- **API Documentation:** Swagger/OpenAPI - Auto-generated API docs
- **Monitoring:** Sentry - Error tracking, performance monitoring
- **Analytics:** Google Analytics + Mixpanel - User behavior tracking

5. Functional Requirements

5.1 Insurance Marketplace

The marketplace is the public-facing entry point for customers to browse and purchase insurance.

- Homepage
    - Hero section with search bar
    - Insurance category cards (Motor, Medical, Life, Home, Travel, etc.)
    - Popular policy bundles
    - Trust indicators (customer count, IRA license, ratings)
    - Featured policies
    - Customer testimonials
- Category Browsing
    - List all policies in selected category
    - Filter by: insurance company, price range, coverage type, rating
    - Sort by: price (low to high/high to low), popularity, rating
    - Policy cards showing: company logo, policy name, key features, starting price
    - Pagination (20 policies per page)
    - Add to cart and compare buttons
- Policy Detail Page
    - Complete policy information
    - Company details and rating
    - Coverage breakdown (what's covered / not covered)
    - Customer reviews and ratings
    - Instant quote form (2-3 inputs)
    - Related/similar policies
    - Add to cart button
    - Download policy wording PDF
- Policy Comparison
    - Compare up to 3 policies side-by-side
    - Feature-by-feature comparison table
    - Price comparison
    - Coverage limits comparison
    - Export to PDF
    - Share comparison link
- Shopping Cart
    - View all selected policies

    - Edit policy options
    - Apply bundle discounts automatically
    - Select payment plans (full, installments, monthly)
    - Show total premium breakdown
    - Proceed to checkout

- Checkout Process

    - Multi-step wizard (4 steps)
    - Step 1: Verify personal information
    - Step 2: Select payment method
    - Step 3: Review order
    - Step 4: Payment confirmation
    - Guest checkout support (require registration only at payment)
    - Save progress (resume later)

5.2 Customer Dashboard

Self-service portal for policy management after login.

- Dashboard Overview

    - Quick stats (active policies, pending payments, claims)
    - Action items requiring attention
    - Active policies grid/list
    - Policy recommendations
    - Recent activity log
    - Quick actions (pay, file claim, download certificate)

- My Policies

    - List all policies (active, pending, expired)
    - Filter and sort options
    - Detailed policy view (coverage, documents, payment history)
    - Download certificate (PDF)
    - File claim button
    - Renew policy
    - Update policy details
    - Cancel policy (with confirmation)

- Payment Management

    - Pending payments section with due dates
    - Make payment (M-Pesa or card)
    - Payment history table
    - Download receipts
    - Saved payment methods
    - Set up auto-pay

    - Installment tracker

- Claims Management
    - File new claim (step-by-step wizard)
    - Upload supporting documents/photos
    - Track claim status in real-time
    - View claim history
    - Chat with claims assessor
    - Download settlement letter

- Document Hub
    - All insurance certificates
    - Medical cards (digital wallet compatible)
    - Uploaded documents (ID, KRA PIN, etc.)
    - Payment receipts
    - Policy documents
    - Download all as ZIP
    - Email documents to self

- Profile Management
    - Update personal information
    - Change password
    - Add/edit beneficiaries
    - Manage dependents (for medical insurance)
    - Notification preferences
    - Communication preferences (email, SMS, WhatsApp)
    - Two-factor authentication setup

5.3 Admin Panel

Internal tool for agency staff to manage all operations.

- Dashboard
    - Real-time statistics (policies, revenue, users, claims)
    - Recent activity feed
    - Alerts and notifications
    - Performance metrics
    - Quick action buttons

- Policy Management
    - Add/edit/delete policies
    - Configure pricing and coverage options
    - Set up policy workflows (custom stages)

    - Manage insurance companies
    - Set bundle discounts
    - Policy approval workflow
    - Bulk policy upload (CSV)

- User Management
    - View all customers
    - Add/edit/suspend users
    - View user details and policy portfolio
    - Upload documents on behalf of user
    - Send payment reminders
    - Prompt M-Pesa payment
    - View user activity log
    - Staff user management with role assignment

- Transaction Management
    - View all transactions
    - Filter by status (completed, pending, failed)
    - Payment reconciliation
    - Process refunds
    - Export transaction reports
    - Failed payment retry

- Claims Processing
    - View all claims
    - Assign claims to assessors
    - Review claim documents
    - Approve/reject claims
    - Add notes and comments
    - Process settlement payments
    - Claims analytics

- Reports & Analytics
    - Sales reports (daily, weekly, monthly)
    - Revenue reports by product/company
    - Claims ratio analysis
    - User growth metrics
    - Conversion funnel analysis
    - Custom report builder
    - Export to PDF/Excel
    - Schedule automated reports

- Settings

- General settings (company info, contact)
- Payment gateway configuration
- Email/SMS settings
- Notification templates
- Role and permission management
- System configuration
- Audit logs

6. Database Design

6.1 Core Tables

- **users:** User accounts (customers and staff)
  - Columns: id, email, password_hash, first_name, last_name, phone, id_number, kra_pin, role, status, created_at, updated_at
- **policies:** Insurance policies
  - Columns: id, policy_number, user_id, policy_type_id, insurance_company_id, status, start_date, end_date, premium_amount, payment_frequency, created_at
- **policy_types:** Policy templates
  - Columns: id, category, name, description, insurance_company_id, base_premium, coverage_details (JSON), created_at
- **transactions:** Payment records
  - Columns: id, user_id, policy_id, amount, payment_method, status, mpesa_receipt, reference_number, created_at
- **claims:** Insurance claims
  - Columns: id, policy_id, user_id, claim_number, type, amount_claimed, status, filed_date, settlement_date, assessor_id
- **documents:** All documents
  - Columns: id, user_id, policy_id, type, filename, s3_key, uploaded_at
- **notifications:** User notifications
  - Columns: id, user_id, type, title, message, read, created_at
- **workflow_stages:** Policy workflow tracking
  - Columns: id, policy_id, stage_name, status, assigned_to, completed_at, notes

7. API Specifications

RESTful API endpoints following industry best practices.

7.1 Authentication Endpoints

- **POST /api/auth/register/** - Register new user
- **POST /api/auth/login/** - Login (returns access + refresh tokens)
- **POST /api/auth/logout/** - Logout
- **POST /api/auth/refresh/** - Refresh access token

- **POST /api/auth/password-reset/** - Request password reset
- **POST /api/auth/password-reset-confirm/** - Confirm password reset

7.2 Policy Endpoints

- **GET /api/policies/** - List all policies (public)
- **GET /api/policies/{id}/** - Get policy details
- **POST /api/policies/{id}/quote/** - Generate quote
- **GET /api/policies/my-policies/** - List user's policies (authenticated)
- **POST /api/policies/purchase/** - Purchase policy
- **GET /api/categories/** - List categories
- **GET /api/companies/** - List insurance companies

8. Security Requirements

Security is paramount given the sensitive nature of insurance data.

- Authentication
  - JWT-based authentication
  - Access tokens (15 min expiry)
  - Refresh tokens (7 days expiry)
  - Token blacklisting on logout
  - Password hashing (PBKDF2)
  - Minimum 8 characters, must include numbers and special characters
- Authorization
  - Role-based access control (RBAC)
  - Granular permissions
  - Row-level security for sensitive data
  - API endpoint protection
- Data Protection
  - HTTPS/SSL for all communications
  - Database encryption at rest
  - S3 encryption for documents
  - PII data encryption
  - Secure session management
  - CSRF protection
  - XSS prevention
  - SQL injection prevention (ORM)
- Compliance
  - Kenya Data Protection Act compliance
  - IRA (Insurance Regulatory Authority) regulations

- PCI-DSS for payment processing
- Regular security audits
- Data retention policies

9. Performance Requirements

Metric	Target
Page Load Time	<3 seconds (3G network)
API Response Time	<200ms (95th percentile)
Concurrent Users	10,000+
Database Query Time	<50ms (95th percentile)
Uptime	99.9% (8.76 hours downtime/year max)
Cache Hit Rate	>80%
Mobile Performance Score	>90 (Lighthouse)
Time to Interactive	<5 seconds

10. Deployment Strategy

10.1 Hosting Infrastructure

Recommended: AWS or DigitalOcean for cost-effectiveness with CloudFlare CDN for global performance.

- **Frontend Hosting:** Vercel (recommended for Next.js) or Netlify - Free tier initially, automatic deployments from Git
- **Backend Hosting:** AWS EC2, DigitalOcean Droplets, or Railway - Start with 2 instances behind load balancer
- **Database:** Managed PostgreSQL (AWS RDS, DigitalOcean Managed DB, or Supabase) - Automated backups, high availability
- **Cache:** Redis Cloud (free tier up to 30MB) or managed Redis on hosting provider
- **File Storage:** AWS S3 - Pay per use, highly reliable (99.999999999% durability)
- **CDN:** CloudFlare - Free tier includes DDoS protection, SSL, caching

10.2 Deployment Pipeline

Automated CI/CD pipeline using GitHub Actions:

- → Developer pushes code to GitHub

- → GitHub Actions triggered automatically
- → Run linting (ESLint, Black)
- → Run tests (Jest, Pytest)
- → Build frontend (Next.js)
- → Build Docker images
- → Push images to registry
- → Deploy to staging environment
- → Run smoke tests
- → If tests pass, deploy to production
- → Send deployment notifications (Slack/Email)

10.3 Environments

Environment	Purpose	Access
Development	Local development on developer machines	Developers only
Staging	Pre-production testing, mirrors production	Internal team
Production	Live environment serving real customers	Public

11. Development Timeline & Milestones

11.1 Phase 1: Foundation (Months 1-2)

- Week 1-2: Project setup, infrastructure provisioning, team onboarding
- Week 3-4: Database design and implementation
- Week 5-6: Authentication system, user management
- Week 7-8: Basic policy browsing and display
- Deliverable: Working prototype with user auth and basic policy listing

11.2 Phase 2: Core Features (Months 3-4)

- Week 9-10: Shopping cart and checkout flow
- Week 11-12: Payment integration (M-Pesa, Paystack)
- Week 13-14: Customer dashboard
- Week 15-16: Policy workflow engine
- Deliverable: End-to-end policy purchase and issuance

11.3 Phase 3: Advanced Features (Months 5-6)

- Week 17-18: Claims management system
- Week 19-20: Admin panel
- Week 21-22: Notifications and communications

- Week 23-24: Testing, bug fixes, performance optimization
- Deliverable: Production-ready platform

11.4 Phase 4: Launch (Month 7)

- Week 25: UAT (User Acceptance Testing)
- Week 26: Staff training and documentation
- Week 27: Soft launch (limited users)
- Week 28: Full public launch
- Deliverable: Live platform with active users

12. Recommended Team Structure

Role	Count	Responsibilities	Est. Cost (KES/month)
Project Manager	1	Overall coordination, stakeholder management	250,000
Senior Full-Stack Dev	1	Architecture, code review, mentoring	350,000
Frontend Developer	2	React/Next.js development	120,000 each
Backend Developer	2	Django/Python development	130,000 each
UI/UX Designer	1	User interface and experience design	100,000
QA Engineer	1	Testing, quality assurance	80,000
DevOps Engineer	1 (PT)	Deployment, infrastructure management	150,000
Total			~1,300,000

Note: Part-time (PT) roles can be engaged on contract basis. Costs are estimates based on Kenyan market rates as of January 2026.

13. Appendices

13.1 Glossary

- **API:** Application Programming Interface - software intermediary
- **CDN:** Content Delivery Network - distributed server network
- **CRUD:** Create, Read, Update, Delete - basic data operations
- **JWT:** JSON Web Token - authentication standard
- **KYC:** Know Your Customer - identity verification
- **ORM:** Object-Relational Mapping - database abstraction
- **RBAC:** Role-Based Access Control - permission system
- **REST:** Representational State Transfer - API architecture
- **SLA:** Service Level Agreement - performance commitment
- **SSR:** Server-Side Rendering - pre-render pages on server

13.2 References

- Django Documentation: https://docs.djangoproject.com/
- React Documentation: https://react.dev/
- Next.js Documentation: https://nextjs.org/docs
- PostgreSQL Documentation: https://www.postgresql.org/docs/
- M-Pesa Daraja API: https://developer.safaricom.co.ke/
- Paystack API: https://paystack.com/docs/api/
- Kenya Data Protection Act: https://www.odpc.go.ke/
- IRA Kenya: https://www.ira.go.ke/

13.3 Document Control

Version	Date	Author	Changes
1.0	2026-01-26	Technical Team	Initial document creation

END OF DOCUMENT

Bowman Insurance Agency - Digital Insurance Platform

Technical Specifications Document v1.0 - January 2026