IMPACT Technical Specifications

Version: 1.6.2

Last Updated: January 2026

Document Status: Production

Table of Contents

1. System Overview
2. Architecture
3. Technology Stack
4. Database Schema
5. API Specifications
6. Data Structures
7. Security Implementation
8. Performance Specifications
9. Integration Points
10. Testing Framework

System Overview

Purpose

IMPACT (Integrated Monitoring Platform for Audit Care & Treatment) is an enterprise-grade surgical outcomes tracking system designed for NHS healthcare providers. The system provides comprehensive data management for colorectal cancer care with full National Bowel Cancer Audit (NBOCA) compliance.

Design Philosophy

- **Episode-Based Care Model**: Hierarchical patient→episode→treatment structure
- **NBOCA-First Design**: All 59 mandatory COSD fields natively supported
- **Security by Default**: AES-256 encryption for sensitive patient data
- **Audit Trail**: Comprehensive CRUD operation logging with user context
- **Data Quality**: Real-time validation and completeness tracking

System Metrics

Metric	Specification
Lines of Code	~15,900 (backend + frontend)
API Endpoints	50+ REST endpoints
Database Collections	9 collections
Supported ICD-10 Codes	63 colorectal cancer codes
Supported OPCS-4 Codes	126 surgical procedure codes
NBOCA Field Coverage	59/59 mandatory fields (100%)
Concurrent Users	Up to 50 simultaneous users
Data Encryption	AES-256 for PII fields
API Response Time	<100ms average (indexed queries)

Architecture

High-Level Architecture

3-Tier Architecture:

[1] CLIENT TIER

- React 18 SPA (TypeScript + Tailwind CSS)
- User Interface Components

- Client-side Routing (React Router)
- State Management (React Hooks)
- Form Validation (Real-time)

↓ Communication: HTTPS (TLS 1.2+) | REST API (JSON)

[2] APPLICATION TIER

- FastAPI Backend (Python 3.10+)

Authentication & Authorization

- JWT Token Management
- Role-Based Access Control (RBAC)
- Password Hashing (bcrypt)

Business Logic Layer

- Patient Management
- Episode Management
- Treatment Recording
- Data Validation (Pydantic)

Encryption Layer

- AES-256 Field Encryption
- PBKDF2 Key Derivation
- Searchable Hash Generation

Export & Reporting

- COSD XML Generation
- Excel Report Export
- Data Quality Dashboard

↓ Communication: Motor (Async MongoDB Driver)

[3] DATA TIER

- MongoDB 6.0+ (Document Store)

impact (Clinical Data Database)

- patients - demographics, medical history
- episodes - cancer/IBD/benign episodes
- treatments - surgery/chemo/radio
- tumours - TNM staging, pathology
- investigations - imaging, endoscopy

- audit_logs - CRUD operation tracking

impact_system (System Data Database)

- users - authentication, roles
- clinicians - surgeon directory
- nhs_providers - NHS org codes

Deployment Architecture

Production Deployment Stack:

NETWORK LAYER

- Nginx Reverse Proxy (SSL/TLS Termination)
- HTTPS (443) → Frontend (3000)
- HTTPS (443) → Backend API (8000)
- Rate Limiting
- Request Logging

↓

APPLICATION SERVICES LAYER

- Frontend Service (systemd)
- Port: 3000
- User: root
- Service: impact-frontend

- Backend Service (systemd)
- Port: 8000
- User: root
- Service: impact-backend

↓

DATABASE SERVICE LAYER

- MongoDB 6.0
- Port: 27017 (localhost only)
- Auth: Enabled (SCRAM-SHA-256)
- Storage: WiredTiger with encryption

Technology Stack

Frontend Technologies

Component	Technology	Version	Purpose
Framework	React	18.3.1	UI component library
Language	TypeScript	5.7.2	Type-safe JavaScript
Build Tool	Vite	6.0.5	Fast build tool and dev server
Styling	Tailwind CSS	3.4.17	Utility-first CSS framework
Routing	React Router	6.30.2	Client-side routing
HTTP Client	Axios	1.7.9	Promise-based HTTP client
UI Components	Headless UI	2.2.0	Accessible UI components
Icons	Heroicons	2.2.0	SVG icon library
Charts	Recharts	2.15.0	Composable charting library
Date Handling	date-fns	4.1.0	Date utility library
Keyboard Shortcuts	react-hotkeys-hook	5.2.1	Keyboard shortcut hooks

Backend Technologies

Component	Technology	Version	Purpose
Framework	FastAPI	0.115.6	High-performance async web framework
Language	Python	3.10+	Backend programming language
Server	Uvicorn	0.34.0	ASGI server with uvloop
Database Driver	Motor	3.6.0	Async MongoDB driver
Validation	Pydantic	2.10.5	Data validation using Python type hints
Authentication	Python-JOSE	3.5.0	JWT token handling
Password Hashing	Passlib (bcrypt)	1.7.4	Secure password hashing
Encryption	Cryptography	Latest	AES-256 field-level encryption
Rate Limiting	SlowAPI	0.1.9	Request rate limiting
Excel Export	OpenPyXL	3.1.5	Excel file generation
Testing	Pytest	8.3.4	Testing framework

Database Technology

Component	Technology	Version	Purpose
Database	MongoDB	6.0+	Document-oriented NoSQL database
Storage Engine	WiredTiger	Included	Compression and encryption support
Authentication	SCRAM-SHA-256	Included	Secure password-based authentication
Connection Pooling	Motor	3.6.0	Async connection pool management

DevOps & Infrastructure

Component	Technology	Purpose
Service Manager	systemd	Process supervision and logging
Reverse Proxy	Nginx	SSL termination, load balancing
SSL Certificates	Let's Encrypt / Certbot	Free SSL/TLS certificates
Log Management	systemd journal + logrotate	Centralized logging and rotation
Backup	mongodump + encryption	Automated database backups
Version Control	Git	Source code management
OS	Ubuntu 22.04/24.04 LTS	Production operating system

Database Schema

Database Structure

IMPACT uses two MongoDB databases:

1. **impact** - Clinical audit data (patients, episodes, treatments, tumours)
2. **impact_system** - System data (users, clinicians, NHS providers)

Collections Overview

)

Collection	Purpose	Document Count (Typical)
patients	Patient demographics and medical history	8,000+
episodes	Clinical episodes (cancer/IBD/benign)	8,000+
treatments	Individual treatments (surgery/chemo/radio)	8,200+
tumours	Tumour sites with TNM staging	8,000+
investigations	Imaging and laboratory studies	17,500+
clinicians	Active surgical staff directory	20-50
users	System user accounts	10-50
audit_logs	CRUD operation audit trail	Growing (500K+)
nhs_providers	NHS organization lookup (ODS)	24,000+

Data Relationships

```
Patient (1)
    ↓
    ■■■ Episode (N) - Cancer/IBD/Benign
    ■       ↓
    ■       ■■■ Treatment (N) - Surgery/Chemo/Radio
    ■       ■       ↓
    ■       ■       ■■■ Surgery Primary (parent)
    ■       ■       ■■■ Surgery RTT (child → parent)
    ■       ■       ■■■ Surgery Reversal (child → parent)
    ■       ■
    ■       ■■■ Tumour (N) - TNM staging, pathology
    ■       ■
    ■       ■■■ Investigation (N) - Imaging, endoscopy
    ■
    ■■■ [Other Episodes...]
```

Indexes

Performance-Critical Indexes:

```
// patients collection
db.patients.createIndex({ "patient_id": 1 }, { unique: true })
db.patients.createIndex({ "nhs_number_hash": 1 })
db.patients.createIndex({ "mrn_hash": 1 })
db.patients.createIndex({ "mrn": 1 })

// episodes collection
db.episodes.createIndex({ "episode_id": 1 }, { unique: true })
db.episodes.createIndex({ "patient_id": 1 })
db.episodes.createIndex({ "condition_type": 1 })
db.episodes.createIndex({ "referral_date": 1 })
db.episodes.createIndex({ "cancer_type": 1 })

// treatments collection
db.treatments.createIndex({ "treatment_id": 1 }, { unique: true })
db.treatments.createIndex({ "episode_id": 1 })
db.treatments.createIndex({ "patient_id": 1 })
db.treatments.createIndex({ "treatment_type": 1 })
db.treatments.createIndex({ "parent_surgery_id": 1 })

// tumours collection
db.tumours.createIndex({ "tumour_id": 1 }, { unique: true })
db.tumours.createIndex({ "episode_id": 1 })
db.tumours.createIndex({ "patient_id": 1 })
db.tumours.createIndex({ "diagnosis_date": 1 })

// investigations collection
db.investigations.createIndex({ "investigation_id": 1 }, { unique: true })
db.investigations.createIndex({ "episode_id": 1 })
db.investigations.createIndex({ "patient_id": 1 })

// users collection (system db)
db.users.createIndex({ "email": 1 }, { unique: true })

// audit_logs collection
db.audit_logs.createIndex({ "timestamp": -1 })
db.audit_logs.createIndex({ "user_id": 1 })
db.audit_logs.createIndex({ "action": 1 })
```

For complete schema documentation, see DATABASE_SCHEMA.md.

API Specifications

REST API Design

Base URL: `http://impact.vps:8000/api`

Authentication: Bearer token (JWT) in Authorization header

Response Format: JSON

HTTP Status Codes:

- `200 OK` - Successful request
- `201 Created` - Resource created successfully
- `400 Bad Request` - Invalid request data
- `401 Unauthorized` - Missing or invalid authentication
- `403 Forbidden` - Insufficient permissions
- `404 Not Found` - Resource not found
- `422 Unprocessable Entity` - Validation error
- `429 Too Many Requests` - Rate limit exceeded
- `500 Internal Server Error` - Server error

API Endpoints

Authentication Endpoints

```
POST /api/auth/login
  Request: { email, password }
  Response: { access_token, token_type, user }

GET /api/auth/me
  Response: { user_id, email, full_name, role, ... }
```

Patient Endpoints

```
GET /api/patients/
  Query: limit, skip, search
  Response: [{ patient_id, demographics, medical_history, ... }]

POST /api/patients/
  Request: { demographics, medical_history }
  Response: { patient_id, ... }

GET /api/patients/{patient_id}
  Response: { patient_id, demographics, ... }

PUT /api/patients/{patient_id}
  Request: { demographics, medical_history }
  Response: { patient_id, ... }

DELETE /api/patients/{patient_id}
  Response: { message: "Patient deleted successfully" }
```

Episode Endpoints

```
GET /api/episodes/
  Query: limit, skip, condition_type, cancer_type
  Response: [{ episode_id, patient_id, condition_type, ... }]

POST /api/episodes/
  Request: { patient_id, condition_type, cancer_data, ... }
  Response: { episode_id, ... }

GET /api/episodes/{episode_id}
  Response: { episode_id, treatments[], tumours[], ... }

PUT /api/episodes/{episode_id}
  Request: { cancer_data, lead_clinician, ... }
  Response: { episode_id, ... }

DELETE /api/episodes/{episode_id}
  Response: { message: "Episode deleted successfully" }
```

Treatment Endpoints

```
POST /api/episodes/treatments/
  Request: { episode_id, treatment_type, procedure, ... }
  Response: { treatment_id, ... }

GET /api/episodes/treatments/{treatment_id}
  Response: { treatment_id, episode_id, ... }

PUT /api/episodes/treatments/{treatment_id}
  Request: { procedure, outcomes, ... }
  Response: { treatment_id, ... }

DELETE /api/episodes/treatments/{treatment_id}
  Response: { message: "Treatment deleted successfully" }
```

Export Endpoints (Admin Only)

```
GET /api/admin/exports/nboca-xml
  Query: start_date, end_date
  Response: XML file (COSD v9/v10 format)

GET /api/admin/exports/data-completeness
  Response: { total_episodes, patient_demographics, diagnosis, surgery }
```

```
GET /api/admin/exports/nboca-validator
  Response: { summary, episodes: [{ episode_id, errors, warnings }] }
```

Report Endpoints

```
GET /api/reports/summary
  Response: { total_surgeries, complication_rate, mortality_30d_rate, ... }

GET /api/reports/surgeon-performance
  Query: surgeon_name
  Response: { surgeries_by_surgeon: [{ surgeon, total_surgeries, ... }] }

GET /api/reports/export-excel
  Response: Excel file with formatted reports
```

Rate Limiting

- **Default**: 100 requests per minute per IP
- **Authentication endpoints**: 10 requests per minute per IP
- **Export endpoints**: 5 requests per minute per user

Data Structures

Patient Document

```
interface Patient {
  _id: ObjectId;
  patient_id: string;              // 6-character hash (e.g., "ABC123")
  mrn: string | null;              // Medical Record Number (encrypted)
  nhs_number: string | null;       // NHS number (encrypted)
  nhs_number_hash: string | null;  // SHA-256 hash for searching
  mrn_hash: string | null;         // SHA-256 hash for searching

  demographics: {
    first_name: string;            // Encrypted
    last_name: string;             // Encrypted
    date_of_birth: string;         // Encrypted (YYYY-MM-DD)
    age: number | null;
    gender: "male" | "female" | "other";
    ethnicity: string | null;
    postcode: string | null;       // Encrypted
  };

  medical_history: {
    conditions: string[];
    previous_surgeries: object[];
    medications: string[];
    allergies: string[];
    smoking_status: "never" | "former" | "current" | null;
    alcohol_use: string | null;
  };

  created_at: Date;
  updated_at: Date;
}
```

Episode Document

```
interface Episode {
  _id: ObjectId;
  episode_id: string;                    // E-ABC123-01
  patient_id: string;
  condition_type: "cancer" | "ibd" | "benign";

  // NBOCA COSD Fields
  referral_date: string | Date;
  first_seen_date: string | Date | null;
  mdt_discussion_date: string | Date | null;
  referral_source: string | null;        // CR1600
  provider_first_seen: string | null;    // CR1410
  cns_involved: string | null;           // CR2050
  mdt_meeting_type: string | null;       // CR3190
  performance_status: string | null;     // CR0510
  no_treatment_reason: string | null;    // CR0490

  // Clinical team
  lead_clinician: string;
  mdt_team: string[];

  // Episode status
  episode_status: "active" | "completed" | "cancelled";
  treatment_intent: "curative" | "palliative" | null;
  treatment_plan: string | null;

  // Cancer-specific
  cancer_type: "bowel" | "kidney" | "breast_primary" | "oesophageal" | "ovarian" | "prostate" | null;
  cancer_data: object;

  // Related data (denormalized for performance)
  treatments: Treatment[];
  tumours: Tumour[];

  // Audit
  created_at: Date;
  created_by: string;
  last_modified_at: Date;
  last_modified_by: string;
}
```

Treatment Document (Surgery)

```
interface Treatment {
  _id: ObjectId;
  treatment_id: string;                  // T-ABC123-01
  episode_id: string;
  patient_id: string;
  treatment_type: "surgery_primary" | "surgery_rtt" | "surgery_reversal" |
                  "chemotherapy" | "radiotherapy" | "immunotherapy" |
                  "hormone_therapy" | "targeted_therapy" | "palliative" | "surveillance";
  treatment_date: string | Date;
  treating_clinician: string;
  treatment_intent: string;
  notes: string | null;

  // Surgery relationship fields
  parent_surgery_id: string | null;      // For RTT and reversal
  parent_episode_id: string | null;      // Auto-populated
  rtt_reason: string | null;             // Required for RTT
  reversal_notes: string | null;
  related_surgery_ids: Array<{           // For parent surgeries
    treatment_id: string;
    treatment_type: string;
    date_created: Date;
  }>;

  // Provider
  provider_organisation: string | null;  // CR1450

  // Patient vitals (recorded per treatment)
  height_cm: number | null;
  weight_kg: number | null;
  bmi: number | null;
```

```
// Classification
classification: {
  urgency: "elective" | "urgent" | "emergency";
  complexity: string | null;
  primary_diagnosis: string;
  indication: string | null;
};

// Procedure
procedure: {
  primary_procedure: string;
  additional_procedures: string[];
  cpt_codes: string[];
  icd10_codes: string[];
  opcs_codes: string[];              // CR0720 - MANDATORY
  approach: "open" | "laparoscopic" | "robotic" | "converted";
  robotic_surgery: boolean;
  conversion_to_open: boolean;
  conversion_reason: string | null;
  description: string | null;
};

// Timeline
perioperative_timeline: {
  admission_date: string | Date;
  surgery_date: string | Date;
  induction_time: string | Date | null;
  knife_to_skin_time: string | Date | null;
  surgery_end_time: string | Date | null;
  anesthesia_duration_minutes: number | null;
  operation_duration_minutes: number | null;
  discharge_date: string | Date | null;
  length_of_stay_days: number | null;
};

// Team
team: {
  primary_surgeon: string;
  primary_surgeon_text: string;
  assistant_surgeons: string[];
  assistant_grade: string | null;
  second_assistant: string | null;
  anesthesiologist: string | null;
  scrub_nurse: string | null;
  circulating_nurse: string | null;
};

// Intraoperative
intraoperative: {
  anesthesia_type: string | null;
```

```
    asa_score: number | null;          // CR6010 - MANDATORY for surgical patients
    blood_loss_ml: number | null;
    transfusion_required: boolean;
    units_transfused: number | null;
    findings: string | null;
    specimens_sent: string[];
    drains_placed: boolean;
    drain_types: string[];

    // Stoma
    stoma_created: boolean;
    stoma_type: string | null;
    stoma_closure_date: string | Date | null;
    reversal_treatment_id: string | null;

    // Anastomosis
    anastomosis_performed: boolean;
    anastomosis_type: string | null;
    anastomosis_configuration: string | null;
    anastomosis_height_cm: number | null;
    anastomosis_location: string | null;
    anterior_resection_type: string | null;
    defunctioning_stoma: "yes" | "no" | null;
  };

  // Pathology
  pathology: {
    histology: string | null;
    grade: string | null;
    lymph_nodes_examined: number | null;
    lymph_nodes_positive: number | null;
    margins: string | null;
    margin_distance_mm: number | null;
    tumor_size_mm: number | null;
    lymphovascular_invasion: string | null;
    perineural_invasion: string | null;
  };

  // Postoperative events
  postoperative_events: {
    return_to_theatre: {
      occurred: boolean;
      date: Date | null;
      reason: string | null;
      procedure_performed: string | null;
      rtt_treatment_id: string | null;
    };
    escalation_of_care: {
      occurred: boolean;
      destination: string | null;
      date: Date | null;
      reason: string | null;
      duration_days: number | null;
    };
    complications: Array<{
      type: string;
      clavien_dindo_grade: "I" | "II" | "IIIa" | "IIIb" | "IVa" | "IVb" | "V" | null;
      description: string;
      date_identified: Date;
      treatment: string | null;
      resolved: boolean;
    }>;
    anastomotic_leak: {
      occurred: boolean;
      severity: "A" | "B" | "C" | null;
      date_identified: Date | null;
      days_post_surgery: number | null;
      presentation: string | null;
    };
  };

  // Outcomes
  outcomes: {
    readmission_30day: boolean;
    readmission_date: Date | null;
    readmission_reason: string | null;
    mortality_30day: boolean;
    mortality_90day: boolean;
    date_of_death: Date | null;
    cause_of_death: string | null;
  };
}
```

Tumour Document

```
interface Tumour {
  _id: ObjectId;
  tumour_id: string;                     // TUM-ABC123-01
  episode_id: string;
  patient_id: string;
  tumour_type: "primary" | "metastasis" | "recurrence";
  site: string;                          // Anatomical location

  // Diagnosis
  diagnosis_date: string | null;         // CR2030
  icd10_code: string | null;             // CR0370
  snomed_morphology: string | null;      // CR6400
  // TNM Staging - Clinical
  tnm_version: "7" | "8";                // CR2070
  clinical_t: string | null;             // CR0520
  clinical_n: string | null;             // CR0540
  clinical_m: string | null;             // CR0560
  clinical_stage_date: string | null;

  // TNM Staging - Pathological
  pathological_t: string | null;         // pCR6820
  pathological_n: string | null;         // pCR0910
  pathological_m: string | null;         // pCR0920
  pathological_stage_date: string | null;

  // Tumour characteristics
  grade: string | null;                  // pCR0930
  histology_type: string | null;
  size_mm: number | null;

  // Rectal cancer specific
  distance_from_anal_verge_cm: number | null;  // CO5160
  mesorectal_involvement: boolean | null;

  // Pathology
  lymph_nodes_examined: number | null;   // pCR0890
  lymph_nodes_positive: number | null;   // pCR0900
  lymphovascular_invasion: string | null;
  perineural_invasion: string | null;

  // Resection margins
  crm_status: string | null;             // pCR1150 - MANDATORY for rectal
  crm_distance_mm: number | null;
  proximal_margin_mm: number | null;
  distal_margin_mm: number | null;

  // Molecular markers
  mismatch_repair_status: string | null;
  kras_status: string | null;
  braf_status: string | null;

  // Associations
  treated_by_treatment_ids: string[];

  notes: string | null;
  created_at: Date;
  last_modified_at: Date;
}
```

Security Implementation

Authentication

Method: JWT (JSON Web Tokens)

Token Structure:

```
{
  "sub": "user@example.com",
  "role": "surgeon",
  "exp": 1735776000
}
```

Token Lifetime: 24 hours (configurable)

Password Security:

- **Algorithm:** bcrypt with automatic salt generation
- **Work Factor:** 12 rounds (2^12 = 4096 iterations)
- **Storage:** Hashed password only (plaintext never stored)

Password Requirements:

- Minimum 8 characters
- Enforced at application level

Authorization

Role-Based Access Control (RBAC):

Role	Permissions
Admin	Full system access, user management, exports, backups
Surgeon	Read/write patient data, view reports, cannot manage users
Data Entry	Create/edit patients/episodes/treatments, limited reports
Viewer	Read-only access to all patient data and reports

Endpoint Protection:

```
# Example: Admin-only endpoint
@router.get("/admin/exports/nboca-xml")
async def export_nboca_xml(
    current_user: dict = Depends(require_admin),
    db: AsyncIOMotorDatabase = Depends(get_database)
):
    # Only accessible to users with admin role
    ...
```

Encryption

Field-Level Encryption:

- **Algorithm:** AES-256 (Fernet symmetric encryption)
- **Key Derivation:** PBKDF2-HMAC-SHA256 with 100,000 iterations
- **Key Storage:** `/root/.field-encryption-key` (600 permissions)
- **Salt Storage:** `/root/.field-encryption-salt` (600 permissions)

Encrypted Fields:

- NHS Number
- Medical Record Number (MRN)
- First Name
- Last Name
- Date of Birth
- Postcode
- Date of Death

Searchable Hashing:

For NHS numbers and MRNs, searchable SHA-256 hashes enable O(log n) indexed lookups:

```
nhs_number = "1234567890"
encrypted_nhs = encrypt_field('nhs_number', nhs_number)
# Result: "ENC:gAAAAABh..."

nhs_hash = generate_search_hash('nhs_number', nhs_number)
# Result: "c775e7b757ede630cd0aa1113bd102661ab38829..."

# Fast indexed lookup via hash
query = { "nhs_number_hash": nhs_hash }
patient = db.patients.find_one(query)
```

Encryption at Rest:

MongoDB WiredTiger storage engine supports encryption at rest (optional):

```
# mongod.conf
security:
  enableEncryption: true
  encryptionKeyFile: /path/to/keyfile
```

Encryption in Transit:

- MongoDB connections via TLS/SSL (optional, recommended for production)
- HTTPS for all client-server communication via Nginx reverse proxy

Audit Trail

Comprehensive CRUD Logging:

Every create, read, update, delete operation is logged to `audit_logs` collection:

```
interface AuditLog {
  _id: ObjectId;
  timestamp: Date;
  user_id: string;
  username: string;
  action: "create" | "read" | "update" | "delete";
  resource_type: "patient" | "episode" | "treatment" | "tumour" | "investigation";
  resource_id: string;
  changes: object;                 // Before/after values for updates
  ip_address: string;
  user_agent: string;
}
```

PII Redaction:

Sensitive fields are redacted in audit logs:

```
safe_doc = pseudonymize_for_logging(document)
# NHS number, DOB, postcode → [REDACTED]
```

Data Access Controls

Database-Level:

- MongoDB authentication enabled (SCRAM-SHA-256)
- Separate users for application and admin access
- Bind to localhost only (no external access)

Application-Level:

- JWT token required for all API endpoints
- Role-based access control on all routes
- Field-level encryption for PII
- Rate limiting to prevent abuse

Network-Level:

- Firewall rules (UFW) restricting access to internal network
- Nginx reverse proxy with SSL/TLS termination
- Rate limiting at proxy level

Performance Specifications

Response Times

Operation	Target	Actual (Average)
Patient search (indexed)	<100ms	45ms
Episode detail load	<200ms	120ms
Treatment creation	<300ms	180ms
Report generation	<2s	1.2s
NBOCA XML export (1000 episodes)	<10s	6.5s
Excel export	<5s	3.2s

Database Performance

Metric	Specification
Document reads (indexed)	10,000+ per second
Document writes	1,000+ per second
Concurrent connections	100+

Metric	Specification
Index lookup time	<1ms
Full collection scan (8,000 docs)	<200ms

Scalability

Metric	Specification
Concurrent users	Up to 50
Database size	Supports millions of documents
API requests per minute	1,000+ (with rate limiting)
Storage growth	~1GB per 10,000 patients

Caching Strategy

- Frontend: React component memoization
- Backend: No caching (data freshness prioritized)
- Database: WiredTiger cache (50% of RAM by default)

Integration Points

External APIs

NHS ODS API (Organization Data Service)

- **Purpose:** NHS provider organization lookup
- **Endpoint:** `https://directory.spineservices.nhs.uk/ORD/2-0-0/organisations`
- **Usage:** Search for NHS Trust codes and names
- **Rate Limit:** Public API, no authentication required

ICD-10 API (via terminology servers)

- **Purpose:** Validate ICD-10 diagnosis codes
- **Endpoint:** Configured terminology server

- **Usage:** Real-time code validation and lookup

OPCS-4 API (via terminology servers)

- **Purpose:** Validate OPCS-4 procedure codes
- **Endpoint:** Configured terminology server
- **Usage:** Real-time code validation and lookup

COSD XML Export

Format: COSD v9/v10 XML Schema

Submission Target: National Cancer Registration and Analysis Service (NCRAS)

Validation: Pre-submission validation with detailed error reporting

Fields Mapped: All 59 mandatory NBOCA COSD fields

See COSD Export Documentation for detailed specifications.

Testing Framework

Unit Testing

Framework: Pytest

Coverage: Backend models, utilities, encryption

```
cd /root/impact/backend

pytest tests/
```

Integration Testing

Scope: API endpoint testing with test database

```
pytest tests/integration/
```

Manual Testing

Test Plan:

1. User authentication (login, logout, token expiry)
2. Patient CRUD operations
3. Episode creation with tumours
4. Treatment recording (surgery, chemo, radio)
5. Surgery relationships (RTT, reversal)
6. NBOCA XML export and validation
7. Excel report generation
8. Role-based access control
9. Audit log verification
10. Encryption/decryption verification

Load Testing

Tools: Apache JMeter, Locust

Scenarios:

- 50 concurrent users browsing patient records
- 10 concurrent users creating episodes
- 5 concurrent users generating reports

End of Technical Specifications

For additional documentation, see:

- USER_GUIDE.md
- DEPLOYMENT_GUIDE.md
- SECURITY_AND_COMPLIANCE.md
- COSD_EXPORT.md
- DATABASE_SCHEMA.md