Product Requirements & Specification Document

Project Name

HealthBridge - Integrated Patient Data Exchange

Description

HealthBridge is a secure, open-source web platform enabling healthcare providers to exchange, analyze, and collaborate on anonymized patient data across organizations. The platform supports interoperability standards, real-time analytics, AI-driven insights, and collaborative case management. Built with Django, Python, PostgreSQL, machine learning, and Docker.

1. Goals & Objectives

Goal	Description
Secure Data Exchange	Enable compliant, encrypted sharing of anonymized patient records
Interoperability	Support healthcare data standards (e.g., HL7 FHIR)
Real-Time Analytics & Visualization	Provide dashboards for treatment outcomes and trends
AI-Driven Insights	Deliver actionable recommendations for care optimization
Collaborative Case Management	Allow multi-organization teams to manage and discuss patient cases
Open-Source & Extensible	Ensure codebase is modular, documented, and community-friendly

2. User Roles & Permissions

Role	Permissions
Admin	Full access: user/org management, system config, audit logs
Provider	View/share data, analytics, case management, receive insights
Data Analyst	Access anonymized data, analytics, export reports
Guest/Student	Limited access to demo data and analytics

3. Functional Requirements

3.1 User Management

- Registration, authentication (OAuth2), and role-based access control
- Organization onboarding and management

3.2 Data Exchange & Integration

- Secure upload/download of patient records (CSV, HL7 FHIR JSON)
- Data anonymization pipeline (removal of PHI)
- API endpoints for data exchange (RESTful, documented via OpenAPI)
- Data validation and error handling

3.3 Interoperability

- Support for HL7 FHIR data model
- Mapping and transformation tools for common EHR formats

3.4 Analytics & Visualization

- Real-time dashboards: treatment outcomes, trends, cohort analysis
- Customizable data visualizations (charts, graphs)
- Export analytics reports (PDF, CSV)

3.5 AI-Driven Insights

- Machine learning models for outcome prediction and care recommendations
- Display insights in provider dashboards
- Feedback loop for model improvement

3.6 Collaborative Case Management

- Shared case workspaces with discussion threads
- Assign/refer cases across organizations
- Audit trail for case activity

4. Non-Functional Requirements

Requirement	Specification
Security	End-to-end encryption (TLS), data anonymization, audit logging
Compliance	HIPAA/GDPR-ready, configurable data retention
Performance	<2s response time for analytics queries (up to 10k records)
Scalability	Dockerized, supports horizontal scaling
Availability	99.5% uptime target
Documentation	API docs (OpenAPI), user/admin guides
Open Source	MIT License, contribution guidelines, code of conduct

5. Technical Specifications

Component	Technology / Approach
Backend	Django (Python 3.10+), Django REST Framework
Database	PostgreSQL 14+
Data Integration	Custom ETL pipelines, FHIR mapping utilities

Machine Learning	scikit-learn, TensorFlow/PyTorch (as needed)
Frontend	Django templates, Chart.js/D3.js for visualizations
Containerization	Docker, docker-compose
API Documentation	OpenAPI/Swagger
CI/CD	GitHub Actions, Docker Hub

6. Key APIs (Pseudocode)

```
# Example: Patient Data Upload (FHIR)
POST /api/v1/patients/upload/
{
  "organization_id": "...",
  "data": [FHIR_JSON]
}

# Example: Get Analytics Dashboard
GET /api/v1/analytics/dashboard/?org_id=...

# Example: AI Insights
GET /api/v1/insights/?case_id=...
```

7. Milestones & Deliverables

Milestone	Deliverable	Timeline (Weeks)
Project Setup	Repo, Docker, CI/CD, base Django app	1
User & Org Management	Auth, roles, onboarding	2
Data Exchange API	Secure upload/download, FHIR support	3
Analytics & Visualization	Dashboards, export features	2
AI Insights	ML models, integration, feedback loop	2
Case Management	Collaborative tools, audit trail	2
Documentation & Release	API docs, user/admin guides, open-source launch	1

8. Risks & Mitigations

Risk	Mitigation
Data privacy breaches	Strict anonymization, encryption, audit logs
Interoperability issues	Adhere to FHIR, provide mapping tools
Model bias/inaccuracy	Continuous feedback, transparent model updates

Scalability bottlenecks	Dockerization, load testing, modular design

---

9. Success Metrics

- 100% data exchange via FHIR-compliant APIs
- <2s analytics dashboard load time (10k records)
- 95%+ user satisfaction (post-pilot survey)
- 100% open-source code coverage with documentation

---

10. Appendix

- FHIR Standard
- MIT License Template
- OpenAPI Specification

End of Document