Product Requirements Document (PRD)

AI Medical Document Chatbot

Version: 1.0

Date: October 14, 2025

Target Market: India

Status: Planning Phase

---

1. Executive Summary

1.1 Product Vision

An AI-powered chatbot that helps patients, families, and caregivers understand medical prescriptions, diagnoses, and general health information. The system will extract information from doctor's handwritten prescriptions and typed medical reports, then provide clear, accessible explanations in natural language.

1.2 Problem Statement

- Patients struggle to understand doctor's handwriting on prescriptions
- Medical terminology and abbreviations are confusing for non-medical users
- Lack of accessible information about medications, side effects, and dietary restrictions
- Language barriers and limited health literacy in India
- No easy way to get quick clarifications without visiting the doctor again

1.3 Solution Overview

A multi-agent AI system built on Microsoft Azure that:

1. Extracts text from medical documents using OCR
2. Answers medical queries using specialized AI models
3. Provides drug information, dietary guidance, and health education
4. Stores user data securely for historical reference

1.4 Success Metrics

- **Accuracy**: >90% OCR accuracy on prescriptions

- **User Satisfaction**: >4.0/5.0 rating
- **Response Time**: <5 seconds for queries
- **Adoption**: 10,000 active users in first 6 months
- **Safety**: Zero incidents of harmful medical advice

---

2. Target Users

2.1 Primary Users

1. **Patients** (Age 25-65)
    - Need to understand their own prescriptions
    - Want to know about side effects and interactions
2. **Family Members/Caregivers**
    - Managing elderly parents' medications
    - Need dietary and care instructions
3. **Health-Conscious Individuals**
    - Seeking general medical information
    - Want to understand health reports

2.2 User Personas

Persona 1: Ramesh (58, Diabetic Patient)

- Receives multiple prescriptions monthly
- Struggles with English medical terms
- Wants to know if medications conflict
- Needs diet recommendations

Persona 2: Priya (32, Caregiver)

- Manages mother's medications
- Wants reminders and explanations
- Needs to understand dosage instructions
- Prefers Hindi interface

Persona 3: Arjun (28, Health Enthusiast)

- Asks general health questions
- Wants to understand lab reports
- Seeks preventive health advice

---

3. Core Features & Requirements

3.1 Must-Have Features (MVP - Phase 1)

F1: Document Upload & OCR

Description: Users can upload prescription images (photo/scan)

Requirements:

- Support formats: JPG, PNG, PDF
- Max file size: 10MB
- Process within 10 seconds
- Extract: Medicine names, dosage, frequency, doctor notes
- Handle handwritten and typed prescriptions
- Minimum 85% accuracy on extraction

Acceptance Criteria:

- □ User can upload document via mobile/web
- □ System extracts structured data (JSON format)
- □ Low confidence extractions are flagged for review
- □ User can correct misread text

F2: Medical Q&A Chatbot

Description: Answer general medical queries

Requirements:

- Support queries about:
  - Diseases and symptoms
  - Medications and side effects
  - Diet and nutrition
  - General health advice
- Response time: <5 seconds
- Include sources/references where applicable
- Support English and Hindi

Acceptance Criteria:

- □ Provides accurate, helpful responses
- □ Includes medical disclaimers
- □ Refuses to diagnose new conditions
- □ Refuses to alter prescribed medications
- □ Natural conversational flow

F3: Drug Information Database

Description: Detailed information about medications

Requirements:

- Database of common Indian medications
- Information includes:
  - What the drug treats
  - How to take it
  - Side effects
  - Food/drug interactions
  - Precautions
- Support generic and brand names
- India-specific formulations

Acceptance Criteria:

- □ Covers top 500 prescribed drugs in India
- □ Information is sourced from CDSCO/reliable databases
- □ Updates quarterly

F4: User Data Storage

Description: Securely store user documents and chat history

Requirements:

- Store uploaded documents encrypted
- Save chat history for context
- User can view/delete their data
- Data retention: configurable per user preference
- Comply with Indian data protection laws

Acceptance Criteria:

- □ All data encrypted at rest and in transit
- □ User authentication required
- □ User can export/delete all data
- □ GDPR-like privacy controls

F5: Safety & Compliance

Description: Ensure safe, responsible medical information

Requirements:

- Display disclaimer on every medical response
- Content filtering for harmful queries
- Cannot diagnose conditions
- Cannot prescribe medications
- Cannot alter existing prescriptions

- Escalation prompts for emergencies

Acceptance Criteria:

- □ Disclaimer visible in UI
- □ Harmful queries refused with explanation
- □ Emergency situations redirected to helpline
- □ Regular safety audits conducted

3.2 Nice-to-Have Features (Phase 2)

F6: Medication Reminders

- Set reminders based on prescription
- Push notifications for doses
- Track adherence

F7: Multi-Language Support

- Add regional languages: Tamil, Telugu, Bengali, Marathi
- Voice input/output

F8: Medical Image Analysis

- Analyze X-rays, lab reports
- Extract values from blood test reports
- Trend analysis for repeat tests

F9: Health Profile

- Store medical history
- Track chronic conditions
- Share reports with doctors

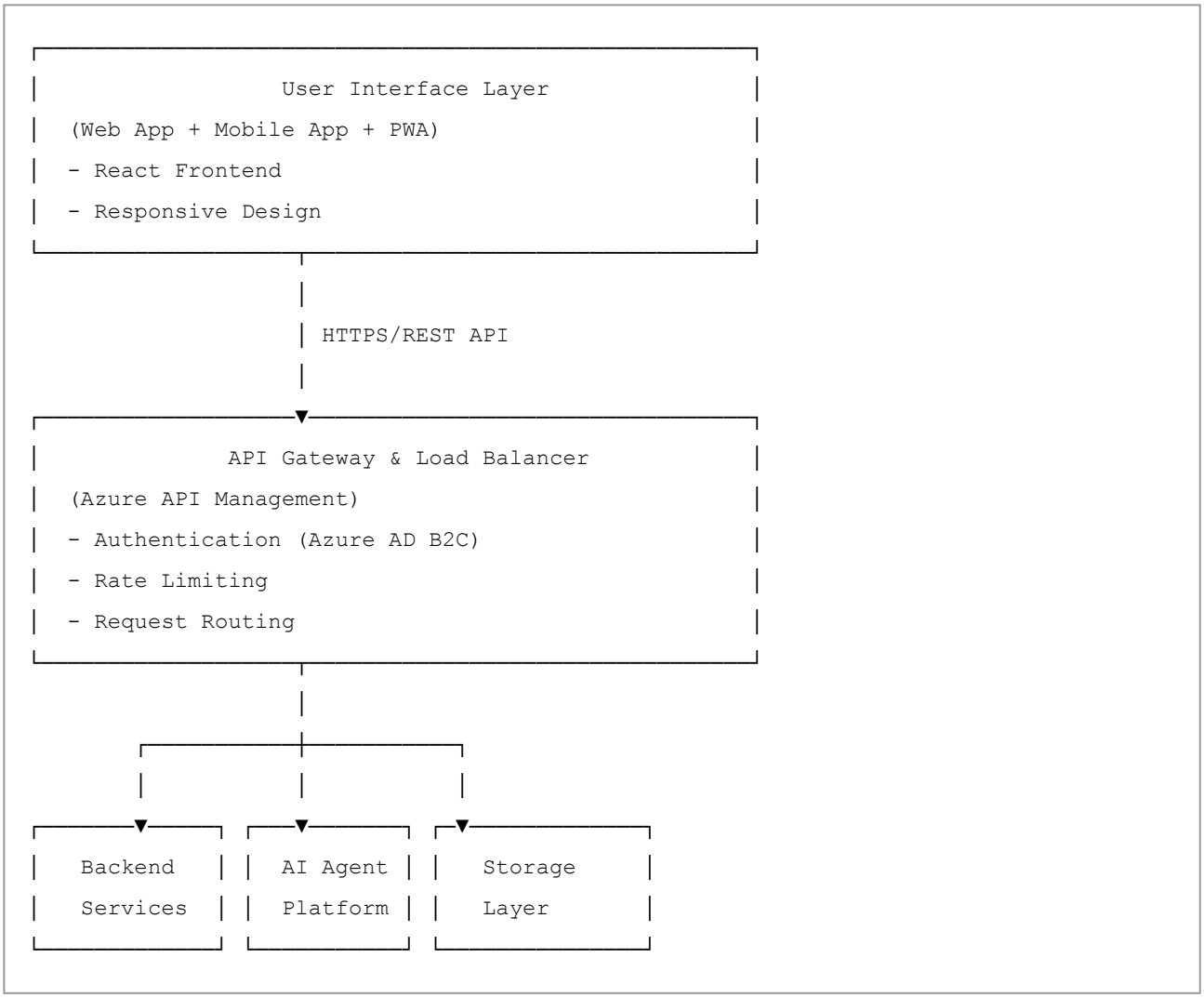
F10: Telemedicine Integration

- Connect with doctors for consultation
- Share extracted data with physician

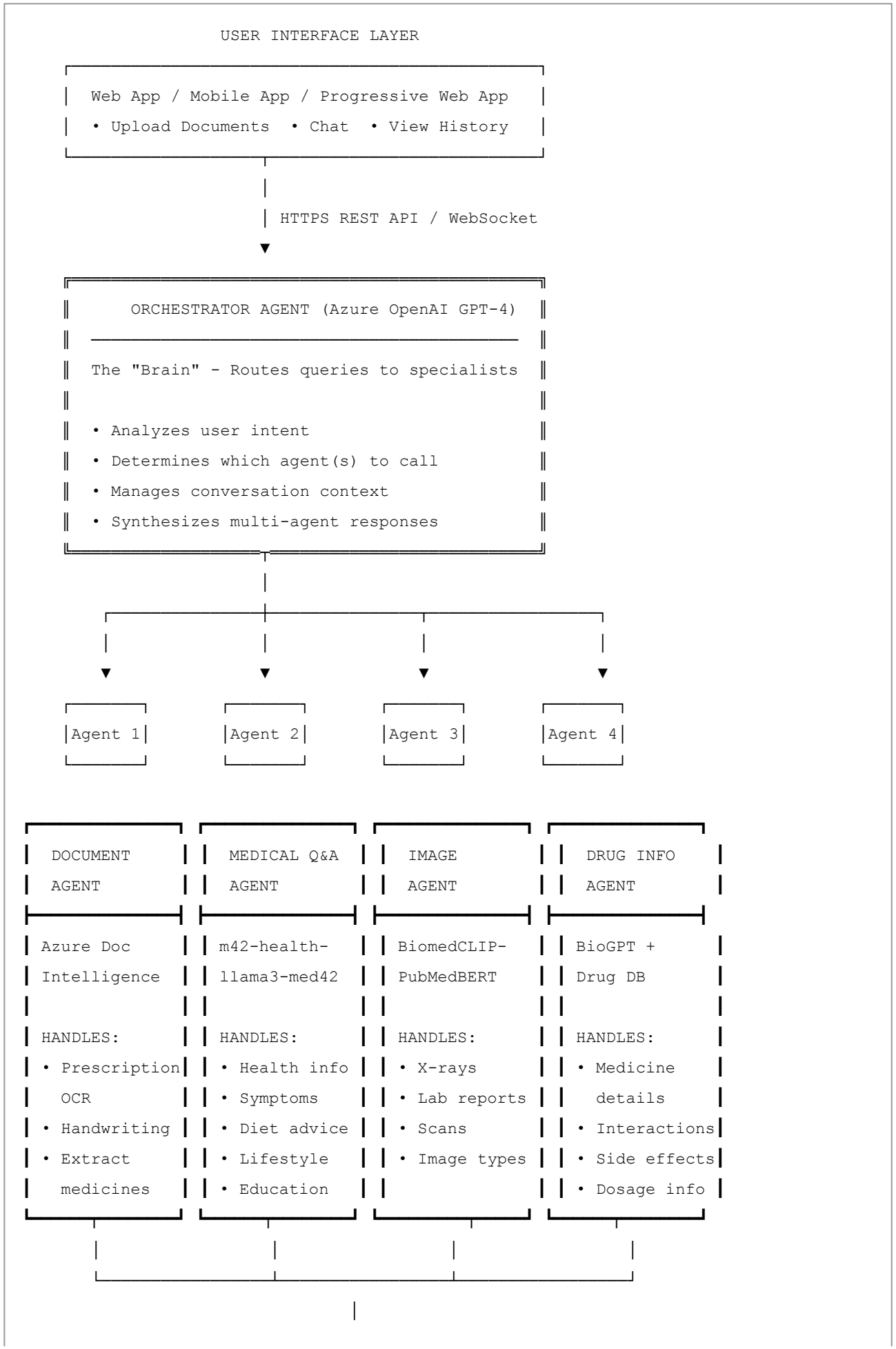
---

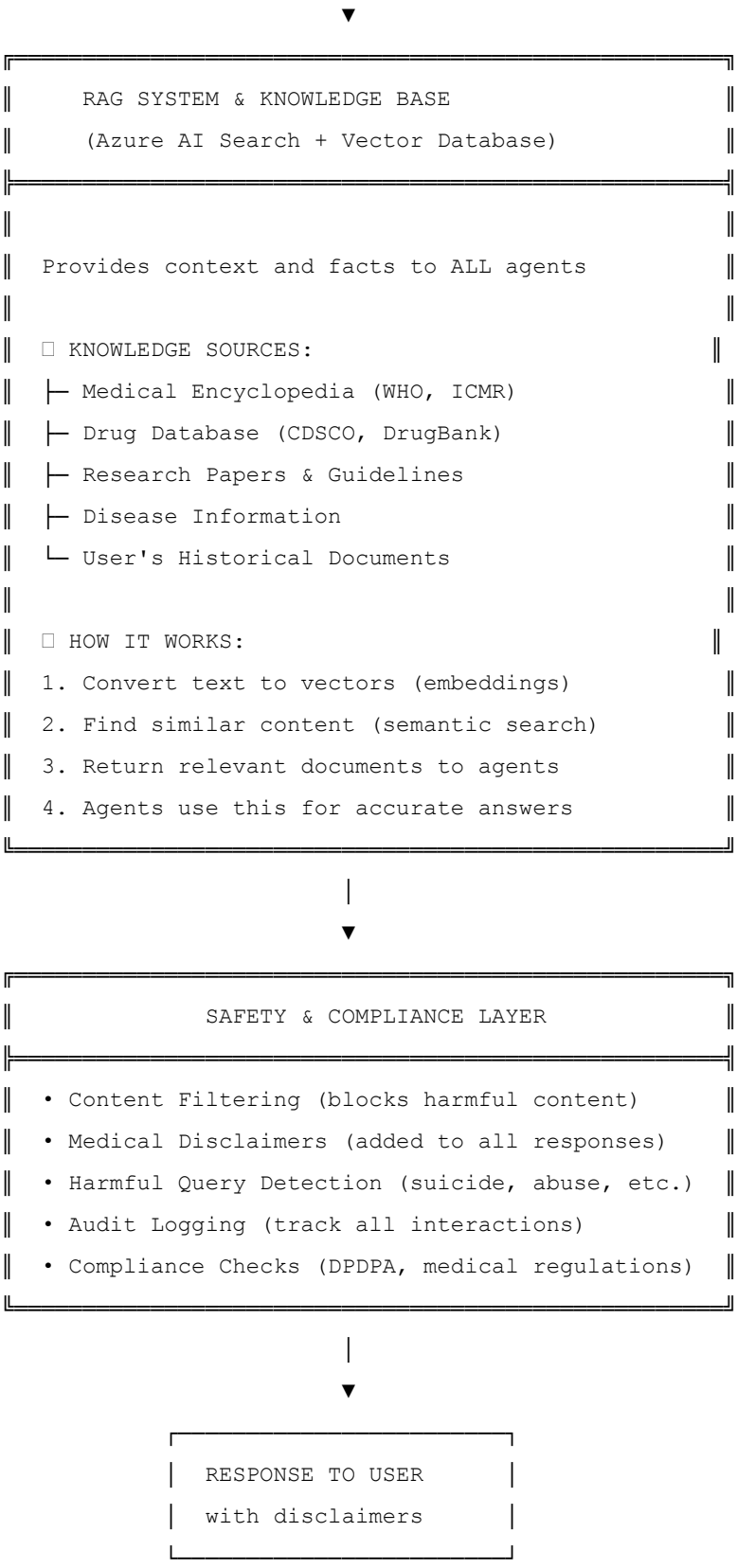
4. System Architecture

4.1 High-Level Architecture



4.2 Multi-Agent AI Architecture


USER INTERFACE LAYER
Web App / Mobile App / Progressive Web App
• Upload Documents • Chat • View History
HTTPS REST API / WebSocket
ORCHESTRATOR AGENT (Azure OpenAI GPT-4)
The "Brain" - Routes queries to specialists
• Analyzes user intent
• Determines which agent(s) to call
• Manages conversation context
• Synthesizes multi-agent responses
Agent 1
Agent 2
Agent 3
Agent 4
DOCUMENT AGENT
Azure Doc Intelligence
HANDLES:
• Prescription OCR
• Handwriting
• Extract medicines
MEDICAL Q&A AGENT
m42-health-llama3-med42
HANDLES:
• Health info
• Symptoms
• Diet advice
• Lifestyle
• Education
IMAGE AGENT
BiomedCLIP-PubMedBERT
HANDLES:
• X-rays
• Lab reports
• Scans
• Image types
DRUG INFO AGENT
BioGPT + Drug DB
HANDLES:
• Medicine details
• Interactions
• Side effects
• Dosage info



How It Works - Example Flow:

Scenario: User asks "What is Metformin? Can I eat bananas?"

```
1. USER → Orchestrator
   Query: "What is Metformin? Can I eat bananas?"

2. ORCHESTRATOR analyzes:
   - "Metformin" = Drug question → Route to Drug Info Agent
   - "bananas" + context = Diet question → Route to Medical Q&A Agent
   - Decision: Call BOTH agents

3. DRUG INFO AGENT:
   ├─ Queries RAG System: "Metformin information"
   ├─ Gets: Drug data from database + vector search results
   ├─ BioGPT generates: "Metformin is a diabetes medication..."
   └─ Returns response to Orchestrator

4. MEDICAL Q&A AGENT:
   ├─ Queries RAG System: "Diabetes diet bananas"
   ├─ Gets: WHO guidelines + ICMR dietary advice
   ├─ Med42-Llama3 generates: "Yes, diabetics can eat bananas in moderation..."
   └─ Returns response to Orchestrator

5. ORCHESTRATOR:
   ├─ Combines both agent responses
   ├─ Adds context from user's prescription (if available)
   └─ Creates unified answer

6. SAFETY LAYER:
   ├─ Checks for harmful content
   ├─ Adds medical disclaimer
   └─ Logs interaction for audit

7. USER receives:
   "Metformin is a medication used to treat Type 2 diabetes...

    Regarding bananas, yes you can eat them in moderation...

    ⚠ This is educational information only. Consult your doctor."
```

4.3 Understanding RAG System & Knowledge Base

What is RAG (Retrieval-Augmented Generation)?

Simple Explanation: Think of RAG like an "open book exam" for AI. Instead of the AI relying only on what it memorized during training (which can be outdated or incomplete), it can look up current, accurate information from a knowledge base before answering.

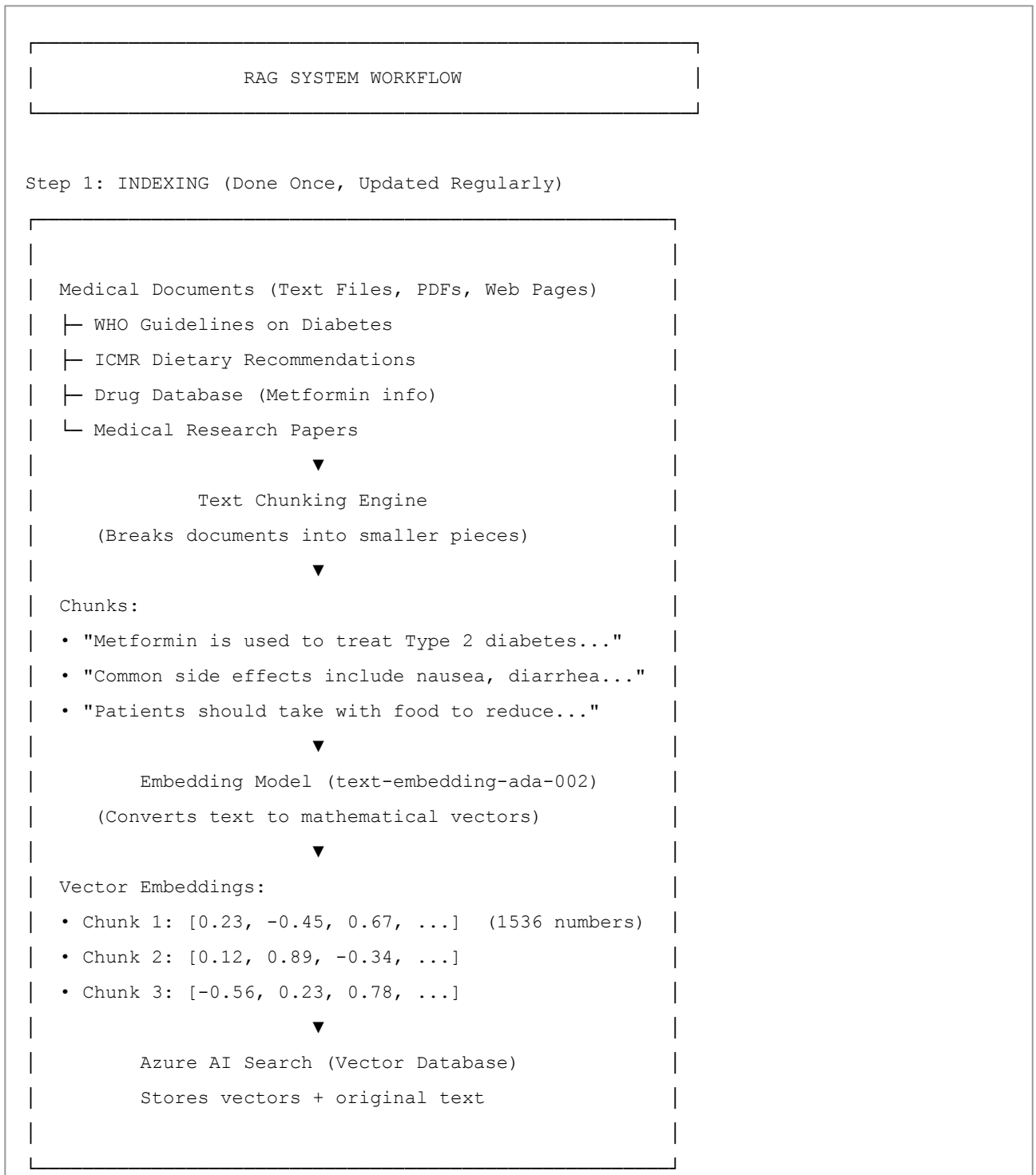
The Problem RAG Solves:

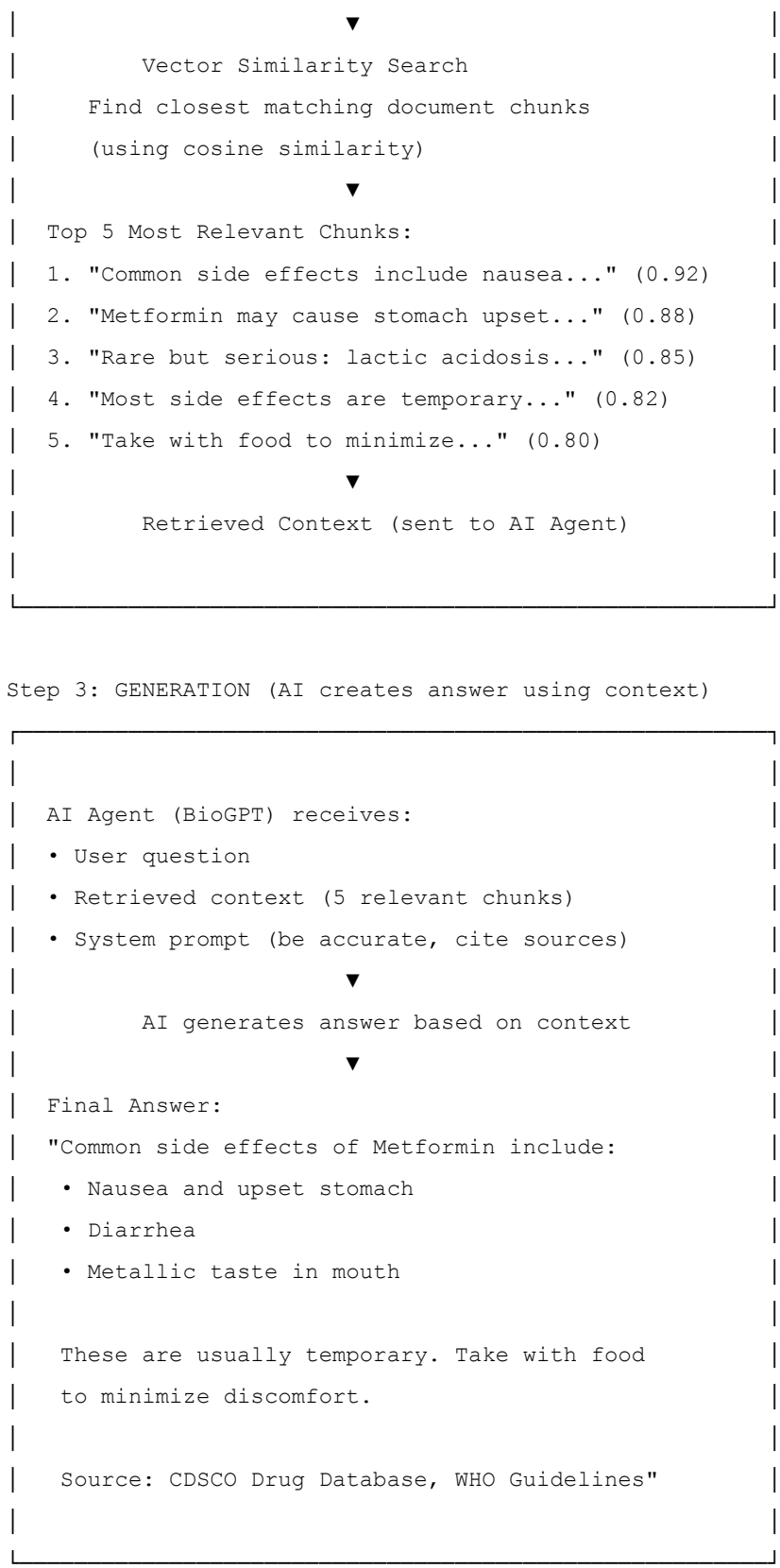
- AI models have a knowledge cutoff date
- Medical information changes frequently
- AI can "hallucinate" (make up facts)
- No citations or sources for answers

How RAG Fixes This:

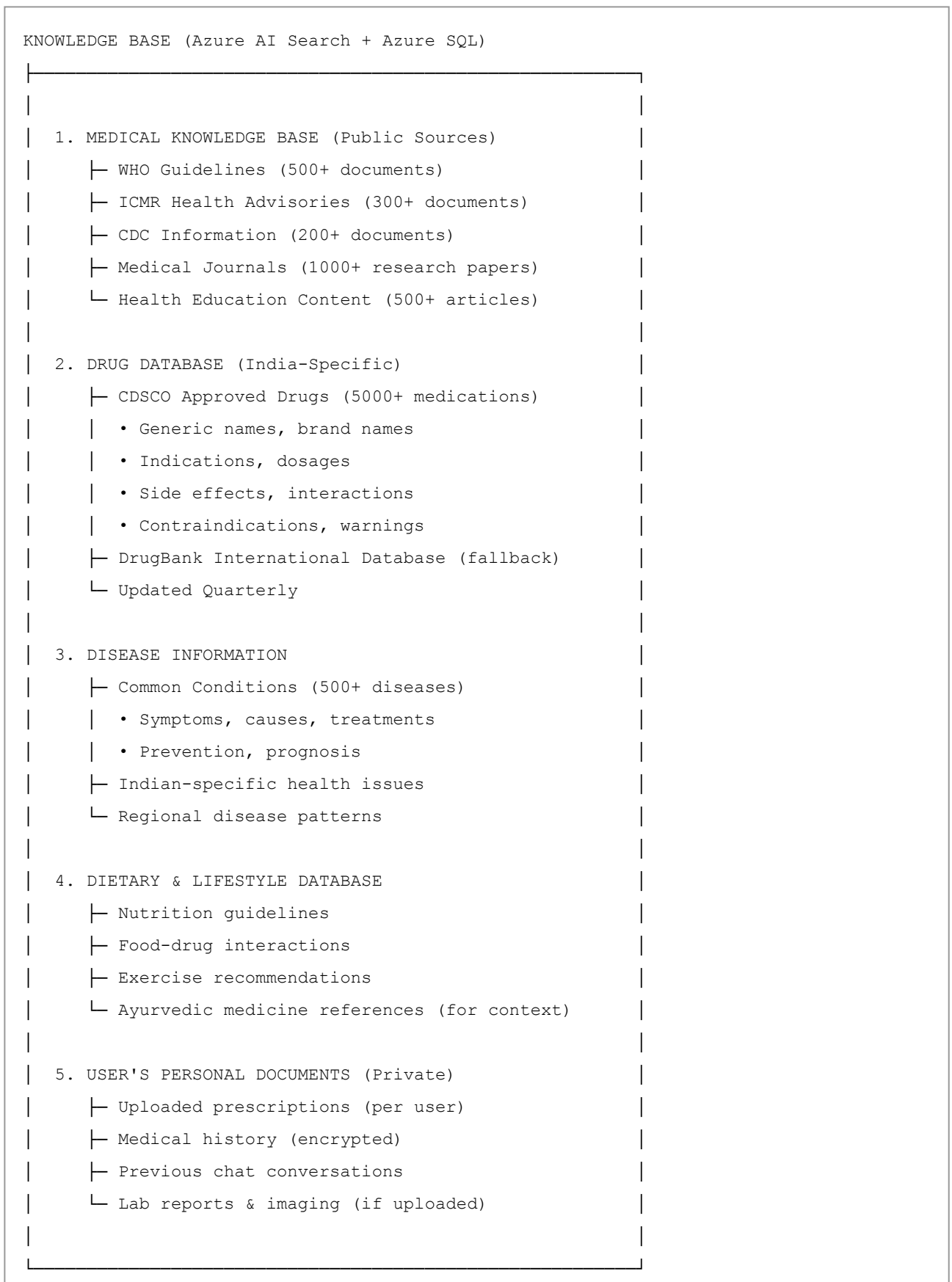
- AI searches a database of reliable documents first
- Finds relevant information for the query
- Uses that information to generate accurate answers
- Can cite sources

RAG System Architecture in Detail

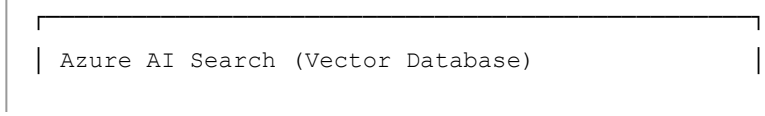

RAG SYSTEM WORKFLOW
Step 1: INDEXING (Done Once, Updated Regularly)
Medical Documents (Text Files, PDFs, Web Pages)
├─ WHO Guidelines on Diabetes
├─ ICMR Dietary Recommendations
├─ Drug Database (Metformin info)
└─ Medical Research Papers
▼
Text Chunking Engine
(Breaks documents into smaller pieces)
▼
Chunks:
• "Metformin is used to treat Type 2 diabetes..."
• "Common side effects include nausea, diarrhea..."
• "Patients should take with food to reduce..."
▼
Embedding Model (text-embedding-ada-002)
(Converts text to mathematical vectors)
▼
Vector Embeddings:
• Chunk 1: [0.23, -0.45, 0.67, ...] (1536 numbers)
• Chunk 2: [0.12, 0.89, -0.34, ...]
• Chunk 3: [-0.56, 0.23, 0.78, ...]
▼
Azure AI Search (Vector Database)
Stores vectors + original text
Step 2: RETRIEVAL (Happens with Every Query)
User Question: "What are side effects of Metformin?"
▼
Embedding Model (same as above)
Converts question to vector
▼
Question Vector: [0.25, -0.43, 0.71, ...]



Knowledge Base Structure



STORAGE BREAKDOWN:



```
| • Stores document vectors for similarity search |
| • ~2GB for 10,000 documents                     |
| • Fast retrieval (<100ms)                       |
└─────────────────────────────────────────────────┘


┌─────────────────────────────────────────────────┐
| Azure SQL Database (Structured Data)            |
| • Drug information (relational)                 |
| • User profiles & prescriptions                 |
| • Optimized for exact lookups                   |
└─────────────────────────────────────────────────┘


┌─────────────────────────────────────────────────┐
| Azure Blob Storage (Original Documents)         |
| • Full PDFs, images                             |
| • User uploaded files                           |
| • Backup of all content                         |
└─────────────────────────────────────────────────┘
```

How Agents Use RAG - Detailed Examples

Example 1: Drug Info Agent Query

```
User asks: "What is the dosage of Metformin for adults?"

1. Drug Info Agent queries RAG:
   - Search: "Metformin dosage adults"
   - RAG retrieves:
     * CDSCO entry: "Initial dose 500mg twice daily..."
     * WHO guideline: "Maximum daily dose 2550mg..."
     * Clinical protocol: "Adjust based on kidney function..."

2. Drug Info Agent + BioGPT:
   - Receives context from RAG
   - Generates structured answer
   - Cites CDSCO database

3. Response: "For adults with Type 2 diabetes:
   - Starting dose: 500mg twice daily with meals
   - Maximum dose: 2550mg per day in divided doses
   - Adjust based on blood sugar control and kidney function

   Source: CDSCO Drug Database"
```

Example 2: Medical Q&A Agent Query

```
User asks: "Can diabetics eat rice?"

1. Medical Q&A Agent queries RAG:
   - Search: "diabetes diet rice carbohydrates India"
   - RAG retrieves:
     * ICMR guideline: "Complex carbs in moderation..."
     * Diabetes India study: "Brown rice preferred..."
     * Dietary advice: "Portion control is key..."

2. Med42-Llama3 + context:
   - Combines retrieved information
   - Adds practical advice
   - Culturally relevant for India

3. Response: "Yes, diabetics can eat rice in moderation:
   - Prefer brown rice over white rice
   - Limit portion to 1/2 cup per meal
   - Pair with vegetables and protein
   - Monitor blood sugar after meals

   Sources: ICMR Diabetes Guidelines, Indian Diabetic Diet Study"
```

Example 3: Multi-Agent Collaboration

```
User uploads prescription + asks: "Is this medicine safe with my condition?"

1. Document Agent:
   - Extracts: "Metformin 500mg, Aspirin 75mg"

2. Orchestrator routes to:
   - Drug Info Agent
   - Medical Q&A Agent

3. Drug Info Agent queries RAG:
   - "Metformin Aspirin interaction"
   - Retrieves: "Generally safe together, monitor for side effects..."

4. Medical Q&A Agent queries RAG:
   - Checks user's condition from previous chats
   - "Diabetes + heart condition + these medications"
   - Retrieves: "Common combination for diabetics with cardiovascular risk..."

5. Orchestrator combines:
   - Both agents' findings
   - User's medical history
   - Generates comprehensive answer

6. Response: "Based on your uploaded prescription:
   - Metformin and Aspirin can be safely taken together
   - This is a common combination for diabetic patients
   - Take Metformin with meals, Aspirin as directed
   - Monitor for stomach upset

   ⚠ Always follow your doctor's instructions. This is educational info only.

   Sources: Drug Interaction Database, Cardiovascular-Diabetes Guidelines"
```

RAG System Benefits for Medical Chatbot

□ **Accuracy**: Always uses latest medical information □ **Transparency**: Can cite specific sources □ **Freshness**: Update knowledge base without retraining AI □ **Privacy**: User data stays separate in encrypted storage □ **Compliance**: Traceable sources for regulatory requirements □ **Personalization**: Can reference user's own documents □ **Cost-Effective**: Don't need to fine-tune large models

---

Update & Maintenance Strategy

```
KNOWLEDGE BASE UPDATES:

Monthly (Automated):
├─ New research papers indexed
├─ Government health advisories added
└─ Drug database updates from CDSCO

Quarterly (Manual Review):
├─ Deprecated information removed
├─ Guidelines updated (WHO, ICMR)
├─ New disease information added
└─ Quality check on vector accuracy

Annual (Complete Refresh):
├─ Re-embed all documents with latest embedding model
├─ Reorganize knowledge structure
├─ Audit for medical accuracy
└─ Expand to new medical domains
```

4.3 Data Flow

Flow 1: Document Processing

```
1. User uploads prescription image
   ↓
2. Image stored in Azure Blob Storage (encrypted)
   ↓
3. Orchestrator routes to Document Agent
   ↓
4. Document Intelligence extracts text
   ↓
5. Structured data (JSON) returned:
   {
     "medicines": [
       {
         "name": "Metformin",
         "dosage": "500mg",
         "frequency": "twice daily",
         "confidence": 0.95
       }
     ],
     "diagnosis": "Type 2 Diabetes",
     "doctor_notes": "Take after meals"
   }
   ↓
6. Data stored in user profile (Azure SQL)
   ↓
7. Orchestrator generates summary for user
   ↓
8. Response displayed with option to ask questions
```

Flow 2: Medical Query

```
1. User asks: "What is Metformin used for?"
   ↓
2. Orchestrator analyzes intent → Drug query
   ↓
3. Routes to Drug Info Agent
   ↓
4. Agent searches drug database + uses BioGPT
   ↓
5. Safety Layer checks response
   ↓
6. Response formatted with disclaimer
   ↓
7. Displayed to user with sources
```

Flow 3: General Health Question

```
1. User asks: "Can diabetics eat mangoes?"
   ↓
2. Orchestrator → Medical Q&A Agent
   ↓
3. RAG retrieves relevant documents
   ↓
4. Med42-Llama3 generates response with context
   ↓
5. Safety Layer validation
   ↓
6. Response with disclaimer shown
```

4.4 Technology Stack

Frontend

- **Framework:** React.js / Next.js
- **Mobile:** React Native / PWA
- **UI Library:** Material-UI / Chakra UI
- **State Management:** Redux / Zustand
- **Authentication:** Azure AD B2C integration

Backend

- **API Layer:** Azure Functions / Azure App Service (Node.js/Python)
- **API Gateway:** Azure API Management
- **Authentication:** Azure AD B2C

- **Database:**
    - Azure SQL Database (structured data)
    - Azure Cosmos DB (chat history, NoSQL)
- **Storage:** Azure Blob Storage (documents)
- **Cache:** Azure Redis Cache

AI/ML Services

- **Orchestrator:** Azure OpenAI (GPT-4)
- **Medical Q&A:** m42-health-llama3-med42 (Azure AI Foundry)
- **Drug Info:** microsoft-biogpt-large (Azure AI Foundry)
- **OCR:** Azure Document Intelligence
- **Vector DB:** Azure AI Search (for RAG)
- **Embeddings:** text-embedding-ada-002 or MedImageInsight-onnx

DevOps & Monitoring

- **CI/CD:** Azure DevOps / GitHub Actions
- **Monitoring:** Azure Application Insights
- **Logging:** Azure Log Analytics
- **Alerts:** Azure Monitor

Security

- **Encryption:** Azure Key Vault
- **WAF:** Azure Web Application Firewall
- **DDoS:** Azure DDoS Protection
- **Compliance:** Azure Policy for HIPAA-equivalent controls

4.5 Azure Resources Breakdown

```
Resource Group: medical-chatbot-prod-rg (West India Region)
├── Compute
│   ├── App Service Plan (Standard S1)
│   ├── Azure Functions (Consumption Plan)
│   └── Container Instances (for agent services)
├── AI Services
│   ├── Azure OpenAI Service
│   ├── Azure AI Foundry (Model deployments)
│   ├── Azure Document Intelligence
│   └── Azure AI Search (Vector DB)
├── Storage
│   ├── Azure Blob Storage (documents)
│   ├── Azure SQL Database (user data)
│   └── Azure Cosmos DB (chat history)
├── Networking
│   ├── Azure API Management
│   ├── Application Gateway
│   └── VNet for private connectivity
├── Security
│   ├── Azure AD B2C (authentication)
│   ├── Key Vault (secrets)
│   └── Azure Firewall
└── Monitoring
    ├── Application Insights
    ├── Log Analytics Workspace
    └── Azure Monitor
```

---

5. User Experience

5.1 User Journey

Journey 1: First-Time User with Prescription

1. **Landing:** User arrives at website/app
2. **Onboarding:** Quick tutorial (30 seconds)
3. **Sign Up:** Email/phone authentication
4. **Upload:** Takes photo of prescription or uploads from gallery
5. **Processing:** "Analyzing prescription..." (5-10 seconds)
6. **Results:** Displays extracted medicines with confidence scores
7. **Correction:** User can tap to correct any misread text
8. **Explanation:** Chatbot proactively explains each medicine

9. **Questions:** User can ask follow-up questions
10. **Save:** Document saved to profile for future reference

Journey 2: Returning User with Medical Query

1. **Login:** Quick authentication
2. **Chat Interface:** Opens to previous conversation
3. **Question:** Types "Can I drink alcohol with these medicines?"
4. **Response:** Bot checks user's saved prescriptions + provides answer
5. **Follow-up:** User asks more questions naturally
6. **Save:** Conversation saved for reference

5.2 Key User Flows

5.3 UI/UX Requirements

Design Principles

1. **Simplicity:** Minimal, uncluttered interface
2. **Accessibility:** Large fonts, high contrast, voice support
3. **Trust:** Professional medical aesthetic, disclaimers visible
4. **Speed:** Fast loading, instant responses
5. **Transparency:** Show confidence scores, sources

Key Screens

Screen 1: Home Dashboard

- Large "Upload Prescription" button
- "Ask a Question" search bar
- Recent prescriptions carousel
- Quick actions: "My Medications", "Reminders", "Profile"

Screen 2: Document Upload

- Camera viewfinder with guidelines
- "Take Photo" / "Choose from Gallery" buttons
- Preview with retake option
- Upload progress indicator

Screen 3: Extraction Results

- Document preview thumbnail
- Extracted medicines in cards
- Each card shows: Name, Dosage, Frequency, Confidence %
- Tap to edit/correct
- "Looks good" confirmation button

Screen 4: Chat Interface

- Clean WhatsApp-style chat bubbles
- Bot avatar (medical cross icon)
- Proactive bot messages explaining medicines
- User input box at bottom
- Disclaimer footer: "This is educational info, not medical advice"

Screen 5: Medicine Detail

- Medicine name (brand + generic)
- "What it's for" section
- "How to take" section
- "Side effects" (collapsible)
- "Interactions" (collapsible)
- "Dietary tips" (collapsible)

---

6. Technical Requirements

6.1 Performance Requirements

- **Response Time:** <5 seconds for chat responses
- **OCR Processing:** <10 seconds for document extraction
- **Uptime:** 99.5% availability
- **Concurrent Users:** Support 1,000+ simultaneous users
- **Scalability:** Auto-scale to handle 10x traffic spikes

6.2 Security Requirements

- **Authentication:** Multi-factor authentication support
- **Data Encryption:**
  - At rest: AES-256
  - In transit: TLS 1.3
- **Access Control:** Role-based access control (RBAC)
- **Audit Logging:** All medical queries logged (anonymized)
- **Data Residency:** All data stored in Indian Azure regions
- **Compliance:**
  - Indian Digital Personal Data Protection Act (DPDPA)
  - ISO 27001 certified infrastructure
  - SOC 2 Type II compliance

6.3 Reliability Requirements

- **Backup:** Daily automated backups, 30-day retention
- **Disaster Recovery:** RPO <4 hours, RTO <8 hours
- **Error Handling:** Graceful degradation (e.g., if OCR fails, allow manual entry)
- **Monitoring:** 24/7 automated monitoring with alerts

6.4 Data Requirements

Data Storage

```
User Profile (SQL):
├── user_id (PK)
├── email
├── phone
├── created_at
├── language_preference
└── privacy_settings

Documents (Blob Storage):
├── document_id (PK)
├── user_id (FK)
├── file_url (blob reference)
├── upload_date
├── document_type (prescription/report)
└── extracted_data (JSON)

Chat History (Cosmos DB):
├── conversation_id (PK)
├── user_id
├── messages: [
│   {
│     timestamp,
│     role: "user"/"assistant",
│     content,
│     sources: []
│   }
├── created_at
└── last_updated

Prescriptions (SQL):
├── prescription_id (PK)
├── user_id (FK)
├── document_id (FK)
├── medicines: [JSON array]
├── diagnosis
├── doctor_name
└── date_prescribed
```

Data Retention

- **User Data:** Retained until user requests deletion
- **Documents:** 5 years (configurable by user)
- **Chat Logs:** 2 years (anonymized after 6 months)

- **Audit Logs:** 7 years (compliance requirement)

6.5 Integration Requirements

- **Payment Gateway:** (Phase 2) Razorpay/Stripe for premium features
- **SMS/Email:** Twilio SendGrid for notifications
- **Analytics:** Google Analytics / Mixpanel
- **Customer Support:** Intercom / Zendesk integration

---

7. AI Model Configuration

7.1 Orchestrator Agent

Model: Azure OpenAI GPT-4

Configuration:

- Temperature: 0.3 (consistent, reliable routing)
- Max Tokens: 500
- Top-p: 0.9
- Frequency Penalty: 0.0

System Prompt:

```
You are an intelligent medical assistant orchestrator. Your job is to:
1. Understand user intent from queries
2. Route queries to appropriate specialized agents
3. Combine responses from multiple agents coherently
4. Always include medical disclaimers
5. Refuse harmful requests politely

Available agents: DocumentAgent, MedicalQAAgent, DrugInfoAgent

Rules:
- Never diagnose new conditions
- Never alter prescribed medications
- Always prioritize safety
- Cite sources when using search results
```

7.2 Medical Q&A Agent

Model: m42-health-llama3-med42

Configuration:

- Temperature: 0.5 (balanced creativity and accuracy)
- Max Tokens: 800
- Top-p: 0.95
- Repetition Penalty: 1.1

RAG Configuration:

- Vector DB: Azure AI Search
- Embedding Model: text-embedding-ada-002
- Top-K documents: 5
- Similarity threshold: 0.7

Knowledge Sources:

- WHO guidelines
- ICMR health advisories
- Trusted medical journals
- Government health portals

7.3 Drug Info Agent

Model: microsoft-biogpt-large

Configuration:

- Temperature: 0.2 (highly factual)
- Max Tokens: 600
- Top-p: 0.85

Database Integration:

- Primary: Custom Indian drug database (CDSCO-sourced)
- Fallback: DrugBank API
- Updates: Quarterly refresh

7.4 Document Agent

Service: Azure Document Intelligence (Form Recognizer)

Configuration:

- Model: prebuilt-read (for general OCR)
- Custom Model: Trained on Indian prescriptions
- Confidence Threshold: 0.75
- Features Enabled:
    - Handwriting recognition
    - Table extraction
    - Key-value pair extraction

Post-Processing:

- Medicine name validation against drug database
- Dosage format standardization (mg, ml, tablets)
- Frequency parsing (OD, BD, TDS → once, twice, thrice daily)

---

8. Safety & Compliance

8.1 Medical Disclaimers

Display on every medical response:

```
⚠ Educational Information Only
This information is for educational purposes and does not
constitute medical advice. Always consult your doctor before
making any changes to your treatment.
```

8.2 Content Filtering Rules

Refuse to:

1. Diagnose conditions ("What disease do I have?")
2. Prescribe medications ("What should I take for headache?")
3. Alter existing prescriptions ("Can I stop this medicine?")
4. Provide emergency medical advice (redirect to emergency services)
5. Give advice on controlled substances

Escalation Triggers:

- Keywords: suicide, overdose, severe pain, chest pain, stroke
- Action: Display emergency helpline numbers
    - India Emergency: 112
    - Ambulance: 102
    - Mental Health: 9152987821 (NIMHANS)

8.3 Quality Assurance

Accuracy Validation:

- Medical advisory board reviews sample responses monthly
- User feedback loop for incorrect information
- A/B testing different model outputs

Testing Protocol:

- 1,000+ test queries covering common scenarios
- Red team testing for harmful outputs
- Adversarial testing for jailbreak attempts

8.4 Privacy & Data Protection

User Rights:

- Right to access: Download all personal data
- Right to deletion: Delete account and all data
- Right to correction: Edit extracted data
- Right to portability: Export data in JSON format

Data Minimization:

- Collect only essential information
- Anonymize data after 6 months for analytics
- No sale/sharing of user data

Consent Management:

- Clear opt-in for data collection
- Granular privacy settings
- Regular consent renewal (annual)

---

9. Implementation Plan

9.1 Development Phases

Phase 0: Setup (Weeks 1-2)

- Azure account setup and resource provisioning
- Development environment configuration
- Team onboarding
- Architecture finalization

Deliverables:

- Azure resources deployed
- Dev/staging/prod environments ready
- Git repository structure
- CI/CD pipelines configured

Phase 1: MVP Development (Weeks 3-10)

Sprint 1 (Weeks 3-4): Document Processing

- Azure Document Intelligence integration
- Image upload functionality
- OCR extraction and structuring
- Basic UI for document upload

Sprint 2 (Weeks 5-6): Chatbot Foundation

- Orchestrator agent implementation
- Medical Q&A agent deployment
- Chat UI development
- Basic conversational flow

Sprint 3 (Weeks 7-8): Drug Database

- Drug information database setup
- Drug Info Agent implementation
- Integration with chat flow
- Medicine detail pages

Sprint 4 (Weeks 9-10): Integration & Testing

- Multi-agent orchestration
- User authentication (Azure AD B2C)
- Data storage implementation
- End-to-end testing

MVP Features: □ Document upload and OCR

□ Basic medical Q&A

□ Drug information lookup

□ User authentication

□ Chat history storage

□ Safety disclaimers

Phase 2: Enhancement (Weeks 11-16)

Sprint 5 (Weeks 11-12): RAG System

- Vector database setup (Azure AI Search)
- Medical knowledge base ingestion
- RAG pipeline implementation
- Citation system

Sprint 6 (Weeks 13-14): UI/UX Polish

- Responsive design improvements
- Mobile app (PWA) optimization
- Accessibility features
- User profile management

Sprint 7 (Weeks 15-16): Safety & Compliance

- Content filtering implementation
- Audit logging
- Privacy controls
- Compliance documentation

Phase 3: Beta Launch (Weeks 17-20)

- Beta user recruitment (100 users)
- Monitoring and bug fixes
- User feedback collection
- Performance optimization
- Documentation

Phase 4: Public Launch (Week 21+)

- Marketing and user acquisition
- Scaled infrastructure
- Customer support setup
- Continuous improvement

9.2 Resource Requirements

Team Structure

Core Team (MVP):

- 1 Product Manager (you)
- 2 Backend Developers (Python/Node.js + Azure)
- 1 Frontend Developer (React)
- 1 ML Engineer (Azure AI/LLMs)
- 1 QA Engineer
- 1 DevOps Engineer (part-time)
- 1 Medical Advisor (consultant)

Phase 2 Additions:

- 1 UI/UX Designer
- 1 Mobile Developer
- 1 Data Scientist

Technology Skills Needed

- Azure cloud services (AI Foundry, Document Intelligence, OpenAI)
- React.js / Next.js
- Node.js / Python (FastAPI)

- LLM orchestration (LangChain/Semantic Kernel)
- Vector databases
- SQL and NoSQL databases
- DevOps (Docker, Kubernetes, CI/CD)

9.3 Budget Estimate (First 6 Months)

Development Costs

- Team salaries (7 people × 6 months): ₹35-50 lakhs
- Freelancers/consultants: ₹5-8 lakhs

Azure Infrastructure (Monthly)

- Azure OpenAI (GPT-4): ₹40,000 - ₹80,000
- Azure AI Foundry (Med42, BioGPT): ₹30,000 - ₹60,000
- Document Intelligence: ₹10,000 - ₹20,000
- Compute (App Services, Functions): ₹15,000 - ₹30,000
- Storage (Blob, SQL, Cosmos): ₹8,000 - ₹15,000
- Networking (API Management, CDN): ₹5,000 - ₹10,000
- **Total Monthly Azure:** ₹1.1 - 2.2 lakhs
- **6 Months Azure:** ₹6.6 - 13.2 lakhs

Other Costs

- Domain, SSL, email: ₹20,000
- Third-party APIs (if any): ₹50,000
- Legal/compliance: ₹2-3 lakhs
- Marketing (beta): ₹5-8 lakhs
- Contingency (20%): ₹10 lakhs

Total 6-Month Budget: ₹64-95 lakhs (~$80K-$115K USD)

9.4 Risks & Mitigation

Risk	Impact	Probability	Mitigation
Low OCR accuracy on handwritten prescriptions	High	Medium	Use custom trained model; allow manual correction; continuous retraining
AI hallucination giving incorrect medical info	Critical	Medium	Implement RAG; Use medical-specific models; Content filtering; Human review
User data breach	Critical	Low	Azure security best practices; Regular security audits; Encryption; Access controls
High Azure costs exceeding budget	High	Medium	Implement cost monitoring; Use reserved instances; Optimize token usage; Cache responses

Risk	Impact	Probability	Mitigation
Regulatory compliance issues	High	Low	Legal consultation; Regular compliance audits; Follow DPDPA guidelines
Low user adoption	High	Medium	User research; Iterative UX improvements; Marketing; Free tier
Model latency affecting UX	Medium	Medium	Caching; Load balancing; Async processing; Model optimization
Medicine database outdated	Medium	Medium	Automated quarterly updates; Partnership with pharma data provider

---

10. Success Metrics & KPIs

10.1 Product Metrics

Engagement Metrics:

- Daily Active Users (DAU)
- Monthly Active Users (MAU)
- Documents uploaded per user
- Questions asked per session
- Session duration
- Return rate (7-day, 30-day)

Quality Metrics:

- OCR accuracy rate (target: >90%)
- Response relevance score (user rating)
- Task completion rate
- Time to first response
- Query resolution rate

Business Metrics:

- User acquisition cost (CAC)
- User retention rate
- Conversion rate (free → paid, if applicable)
- Net Promoter Score (NPS)
- Customer satisfaction (CSAT)

10.2 Success Criteria

MVP Success (3 months post-launch):

- □ 5,000+ registered users
- □ 10,000+ documents processed

- □ 50,000+ chat messages
- □ >85% OCR accuracy
- □ >4.0/5 user satisfaction
- □ <5 second average response time
- □ Zero critical safety incidents

Year 1 Success:

- □ 50,000+ active users
- □ 100,000+ documents processed
- □ NPS >40
- □ Monthly growth rate >15%
- □ Revenue positive (if monetized)
- □ Partnerships with 3+ healthcare providers

10.3 Monitoring Dashboard

Real-Time Metrics:

- Active users (current)
- API response times
- Error rates
- OCR success rate
- AI model latency

Daily Reports:

- New user signups
- Documents processed
- Chat conversations
- User satisfaction scores
- Top queries/features used

Weekly/Monthly Analytics:

- User retention cohorts
- Feature adoption rates
- Cost per user
- Revenue (if applicable)
- Support ticket volume

---

11. Go-to-Market Strategy

11.1 Target Market Segments

Primary Segment (Year 1):

- Urban India (Tier 1 & 2 cities)
- Age: 25-65
- Smartphone users with internet access
- English/Hindi speakers
- Middle-class households

Secondary Segment (Year 2):

- Rural India (Tier 3 cities)
- Elderly population (65+)
- Regional language speakers
- Lower-income groups

11.2 Marketing Channels

Pre-Launch (Beta Phase):

1. **Product Hunt Launch** - Tech-savvy early adopters
2. **Social Media** - LinkedIn, Twitter, Instagram health communities
3. **Healthcare Forums** - Reddit (r/india, r/healthcare), Quora
4. **Partnerships** - Collaborate with 2-3 clinics for pilot testing
5. **Press** - Reach out to health tech journalists (YourStory, Inc42)

Post-Launch:

1. **Content Marketing**
    - Blog: "Understanding Your Prescription", "Common Medicine FAQs"
    - YouTube: Tutorial videos, patient testimonials
    - SEO optimization for health queries
2. **Digital Advertising**
    - Google Ads (search: "understand prescription", "medicine information")
    - Facebook/Instagram Ads (targeting health-conscious users)
    - Budget: ₹50,000-1 lakh/month
3. **Partnerships**
    - Pharmacies: Partner for QR code on medicine bills
    - Diagnostic labs: Integrate with report delivery
    - Health insurance companies: Add-on service
    - Telemedicine platforms: Integration partnerships
4. **Community Building**

- WhatsApp communities for chronic patients
- Free webinars on health literacy
- Ambassador program (patient advocates)

5. **Referral Program**

    - Invite friends, get premium features
    - Healthcare worker referral bonuses

11.3 Pricing Strategy

Freemium Model:

Free Tier:

- 5 document uploads per month
- Basic chat queries (20 per day)
- Standard response time
- Single user profile

Premium Tier (₹299/month or ₹2,999/year):

- Unlimited document uploads
- Unlimited chat queries
- Priority response time
- Medication reminders
- Multi-user profiles (family plan)
- Advanced features (image analysis, trend tracking)
- Ad-free experience
- Export reports (PDF)

Enterprise Tier (Custom Pricing):

- For clinics, hospitals, pharmacies
- White-label solution
- API access
- Custom integrations
- Dedicated support

Monetization Timeline:

- Months 1-6: 100% free (growth focus)
- Months 7-12: Introduce premium tier
- Year 2: Enterprise partnerships

---

12. Future Roadmap (Beyond MVP)

12.1 Phase 3 Features (Months 7-12)

F11: Voice Interface

- Voice input for queries (Hindi + English)
- Text-to-speech responses
- Accessibility for elderly and illiterate users

F12: Medication Reminders & Tracking

- Smart reminders based on prescription
- Adherence tracking
- Refill reminders
- Family caregiver notifications

F13: Health Profile & History

- Longitudinal health record
- Track chronic conditions
- Medication history
- Lab report storage and trends

F14: Regional Language Support

- Tamil, Telugu, Bengali, Marathi, Gujarati
- Language detection and auto-translation
- Localized medical terms

F15: Medical Report Analysis

- Blood test report interpretation
- Trend analysis (glucose, cholesterol over time)
- Highlight abnormal values
- Explain medical terminology

12.2 Phase 4 Features (Year 2)

F16: Telemedicine Integration

- Connect with doctors for consultation
- Share extracted prescription data
- Video call integration
- Doctor marketplace

F17: Pharmacy Integration

- Order medicines directly
- Price comparison

- Delivery tracking
- Authenticity verification

F18: Insurance Integration

- Claim documentation
- Expense tracking
- Insurance plan recommendations
- Cashless treatment guidance

F19: Community Features

- Patient support groups (diabetes, hypertension, etc.)
- Q&A forums (moderated)
- Success stories
- Expert AMA sessions

F20: Personalized Health Insights

- AI-driven health recommendations
- Diet plans based on medications
- Exercise suggestions
- Preventive care reminders

F21: Wearable Integration

- Sync with fitness trackers
- Blood sugar, BP monitoring
- Correlate with medication adherence
- Alert on anomalies

12.3 Long-Term Vision (Year 3+)

Vision: Become India's most trusted AI health companion

Strategic Goals:

1. **Scale:** 5 million+ users across India
2. **Accuracy:** 95%+ satisfaction with AI responses
3. **Impact:** Measurably improve medication adherence
4. **Partnerships:** Integrated with top 50 hospitals
5. **Revenue:** ₹10+ crore annual recurring revenue

Expansion Opportunities:

- International markets (Southeast Asia, Africa)
- B2B SaaS for healthcare providers
- API licensing for third-party apps

- Health insurance premium discounts (evidence-based)
- Chronic disease management programs

---

13. Technical Specifications

13.1 API Design

REST API Endpoints

```
Authentication:
POST /api/v1/auth/signup
POST /api/v1/auth/login
POST /api/v1/auth/logout
GET  /api/v1/auth/verify-token

Documents:
POST /api/v1/documents/upload
GET  /api/v1/documents/{document_id}
GET  /api/v1/documents/list
DELETE /api/v1/documents/{document_id}
POST /api/v1/documents/{document_id}/correct

Chat:
POST /api/v1/chat/message
GET  /api/v1/chat/history/{conversation_id}
GET  /api/v1/chat/conversations/list
DELETE /api/v1/chat/{conversation_id}

Medicines:
GET  /api/v1/medicines/{medicine_name}
GET  /api/v1/medicines/search?q={query}
GET  /api/v1/medicines/interactions?medicines={list}

User Profile:
GET  /api/v1/profile
PUT  /api/v1/profile
POST /api/v1/profile/prescriptions
GET  /api/v1/profile/prescriptions/list
```

WebSocket for Real-Time Chat

```
ws://api.example.com/chat
- Connection with JWT token
- Real-time message streaming
- Typing indicators
- Read receipts
```

13.2 Data Models

```
// User Model
interface User {
  id: string;
  email: string;
  phone?: string;
  name: string;
  dateOfBirth?: Date;
  gender?: 'male' | 'female' | 'other';
  languagePreference: string;
  createdAt: Date;
  lastLogin: Date;
  privacySettings: PrivacySettings;
}

// Document Model
interface Document {
  id: string;
  userId: string;
  fileUrl: string;
  fileName: string;
  fileSize: number;
  mimeType: string;
  uploadedAt: Date;
  documentType: 'prescription' | 'lab_report' | 'medical_bill' | 'other';
  extractedData: ExtractedData;
  ocrConfidence: number;
  status: 'processing' | 'completed' | 'failed';
}

// Extracted Data Model
interface ExtractedData {
  medicines: Medicine[];
  diagnosis?: string;
  doctorName?: string;
  doctorSignature?: string;
  hospitalName?: string;
  prescriptionDate?: Date;
  patientName?: string;
  additionalNotes?: string;
}

// Medicine Model
interface Medicine {
```

```
  name: string;
  genericName?: string;
  dosage: string;
  frequency: string; // "BD", "TDS", "QID", etc.
  duration?: string;
  instructions?: string; // "after meals", "before bed"
  confidence: number;
}

// Chat Message Model
interface ChatMessage {
  id: string;
  conversationId: string;
  userId: string;
  role: 'user' | 'assistant';
  content: string;
  timestamp: Date;
  metadata?: {
    sources?: Source[];
    agentUsed?: string;
    tokens?: number;
    latency?: number;
  };
}

// Drug Information Model
interface DrugInfo {
  name: string;
  genericName: string;
  brandNames: string[];
  category: string;
  uses: string[];
  dosage: {
    adult: string;
    pediatric?: string;
  };
  sideEffects: {
    common: string[];
    serious: string[];
  };
  interactions: {
    drugs: string[];
    food: string[];
    conditions: string[];
```

```
  };
  precautions: string[];
  contraindications: string[];
  pregnancyCategory?: string;
  lastUpdated: Date;
}
```

13.3 Security Implementation

Authentication Flow

```
1. User signs up with email/phone
   ↓
2. Azure AD B2C creates account
   ↓
3. OTP/email verification sent
   ↓
4. User verifies and completes profile
   ↓
5. JWT token issued (expires in 7 days)
   ↓
6. Refresh token stored (expires in 30 days)
   ↓
7. All API calls include Authorization: Bearer {token}
```

Data Encryption

- **At Rest:** Azure Storage encryption with customer-managed keys
- **In Transit:** TLS 1.3 for all API communication
- **Sensitive Fields:** Additional field-level encryption for PII
- **Key Management:** Azure Key Vault with rotation policy

Access Control

```
Roles:
- User: Can access own data only
- Admin: Can view anonymized analytics
- Support: Can view user data with audit logging
- Medical Advisor: Can review flagged content

Permissions enforced at API Gateway level
```

13.4 Monitoring & Alerting

Application Insights Metrics

```
Custom Events:
- document_uploaded
- ocr_completed
- chat_message_sent
- drug_info_queried
- error_occurred

Custom Metrics:
- ocr_accuracy_score
- response_latency_ms
- ai_token_usage
- active_users_count
- api_error_rate
```

Alert Rules

```
Critical Alerts (PagerDuty):
- API error rate >5% for 5 minutes
- Response latency >10 seconds
- Azure service outage
- Data breach detection

Warning Alerts (Email):
- Error rate >2% for 15 minutes
- Budget threshold exceeded (80%)
- OCR accuracy <85%
- Low confidence extractions >20%
```

13.5 DevOps & Deployment

CI/CD Pipeline (Azure DevOps)

Environment Strategy

```
Development:
- Local development with Docker Compose
- Azure Dev environment for integration testing
- Use smaller AI models to reduce costs

Staging:
- Replica of production
- Full-scale Azure resources
- Used for QA and demo
- Synthetic test data

Production:
- High availability setup
- Auto-scaling enabled
- Real user data
- Blue-green deployment for zero downtime
```

---

14. Compliance & Legal

14.1 Regulatory Compliance

India-Specific:

- **Digital Personal Data Protection Act (DPDPA) 2023**
    - Obtain explicit user consent
    - Implement data minimization
    - Provide data portability
    - Right to erasure
- **Information Technology Act, 2000**
    - Secure data storage requirements
    - Incident reporting (72 hours)
- **Telemedicine Practice Guidelines (if applicable in Phase 2)**
    - RMP registration requirements
    - Consent management
    - Prescription format compliance

International Standards (for future expansion):

- GDPR (European Union)

- HIPAA (United States)
- PDPA (Singapore)

14.2 Terms of Service (Key Clauses)

Disclaimer:

```
This application provides educational information only and is not a
substitute for professional medical advice, diagnosis, or treatment.
Always seek the advice of your physician or qualified health provider
with any questions regarding a medical condition.

The OCR technology may contain errors. Users should verify all
extracted information against original prescriptions. We are not
liable for any harm resulting from misinterpreted or incorrect
information.
```

Liability Limitation:

- Service provided "as-is" without warranties
- Not liable for medical decisions made using the app
- Liability capped at subscription fees paid

Data Usage:

- Anonymized data may be used for research and improvement
- No sharing with third parties without consent
- User can opt-out of data usage for research

14.3 Medical Advisory Board

Composition:

- 1 General Physician
- 1 Pharmacist
- 1 AI Ethics Expert
- 1 Patient Advocate

Responsibilities:

- Review AI responses quarterly
- Approve medical content updates
- Assess safety incidents
- Provide guidance on edge cases

---

15. Support & Documentation

15.1 User Support

Support Channels:

1. **In-App Help Center**
    - FAQs (50+ common questions)
    - Video tutorials
    - Troubleshooting guides
2. **Email Support** (support@example.com)
    - Response SLA: 24 hours
    - Available 7 days/week
3. **Chatbot Support**
    - Automated responses for common issues
    - Escalation to human agent if needed
4. **Community Forum**
    - User discussions
    - Moderated by support team

Support Metrics:

- First response time: <24 hours
- Resolution time: <48 hours
- Customer satisfaction: >4.5/5

15.2 Developer Documentation

API Documentation:

- OpenAPI/Swagger specification
- Code examples (Python, JavaScript, curl)
- Postman collection
- Rate limits and best practices

Integration Guides:

- Getting started tutorial
- Authentication guide
- Webhook setup

- Error handling

SDK Libraries:

- Python SDK
- JavaScript/Node.js SDK
- Sample applications

15.3 Internal Documentation

Operations Runbooks:

- Deployment procedures
- Rollback process
- Incident response playbook
- Database backup/restore

Technical Documentation:

- Architecture diagrams (Lucidchart/Draw.io)
- Data flow diagrams
- API specifications
- Database schemas

---

16. Appendices

Appendix A: Glossary

Terms:

- **OCR:** Optical Character Recognition - technology to extract text from images
- **RAG:** Retrieval-Augmented Generation - AI technique combining search with generation
- **LLM:** Large Language Model - AI models like GPT-4, Claude
- **Azure AI Foundry:** Microsoft's platform for deploying AI models
- **Document Intelligence:** Azure's OCR and document processing service
- **Multi-Agent System:** Architecture with multiple specialized AI agents
- **OD/BD/TDS:** Medical abbreviations for once/twice/thrice daily dosing

Appendix B: Reference Links

Azure Documentation:

- Azure AI Foundry: https://ai.azure.com (https://ai.azure.com)
- Document Intelligence: https://learn.microsoft.com/azure/ai-services/document-intelligence (https://learn.microsoft.com/azure/ai-services/document-intelligence)

- Azure OpenAI: https://learn.microsoft.com/azure/ai-services/openai (https://learn.microsoft.com/azure/ai-services/openai)

Medical Resources:

- CDSCO (Drug Regulation): https://cdsco.gov.in (https://cdsco.gov.in)
- ICMR Guidelines: https://www.icmr.gov.in (https://www.icmr.gov.in)
- WHO INN Database: https://www.who.int/teams/health-product-and-policy-standards/inn (https://www.who.int/teams/health-product-and-policy-standards/inn)

AI/ML Resources:

- LangChain Documentation: https://python.langchain.com (https://python.langchain.com)
- Azure Machine Learning: https://learn.microsoft.com/azure/machine-learning (https://learn.microsoft.com/azure/machine-learning)

Appendix C: Sample User Stories

Story 1: Elderly Patient

```
As an elderly patient with multiple medications,
I want to understand what each medicine does,
So that I can take them correctly and safely.

Acceptance Criteria:
- Can upload prescription photo easily
- Receives clear explanation in simple language
- Can ask follow-up questions
- Information saved for future reference
```

Story 2: Caregiver

```
As a caregiver managing my parent's health,
I want to track all medications and schedules,
So that I ensure they take medicines on time.

Acceptance Criteria:
- Can store multiple prescriptions
- Receives reminders for doses
- Can check drug interactions
- Can share information with doctors
```

Story 3: First-time User

```
As a patient receiving my first prescription,
I want to know about side effects and diet,
So that I can take precautions.

Acceptance Criteria:
- Easy signup process
- Clear side effect warnings
- Dietary recommendations provided
- Can ask questions in my language
```

Appendix D: Competitive Analysis

Competitors:

1. **1mg / Tata 1mg**
    - Medicine ordering + info
    - Lacks AI chat and OCR
2. **Practo**
    - Telemedicine + health records
    - No prescription OCR or AI assistant
3. **PharmEasy**
    - Medicine delivery
    - Limited informational features

Our Differentiation:

- □ AI-powered prescription understanding
- □ Advanced OCR for handwritten prescriptions
- □ Conversational interface
- □ Educational focus (not just commerce)
- □ Multi-agent specialized system

---

17. Version History

Version	Date	Author	Changes
0.1	Oct 14, 2025	Initial Draft	Created initial PRD structure
1.0	Oct 14, 2025	Complete Draft	Finalized all sections with architecture

---

18. Approval & Sign-off

Document Owner: [Your Name]

Last Updated: October 14, 2025

Next Review: January 2026

Approvals Required:

- [ ] Product Manager
- [ ] Technical Lead
- [ ] Medical Advisor
- [ ] Legal/Compliance
- [ ] Executive Sponsor

End of Product Requirements Document

For questions or clarifications, contact: [your-email@example.com]