Product Requirements Document (PRD): SmartSPD

Product Name: SmartSPD
Company: BeneSense AI
Version: 1.0
Date: July 1, 2025

1. Introduction

1.1. Purpose of this Document

This Product Requirements Document (PRD) outlines the vision, goals, features, and functional requirements for SmartSPD, a flagship product by BeneSense AI. It serves as a foundational guide for the development, design, and testing teams, ensuring a shared understanding of the product to be built and its value proposition.

1.2. Executive Summary

SmartSPD is an AI-powered customer service agent designed to revolutionize how users (customer service agents, health plan members, HR professionals, brokers, and TPAs) interact with complex health plan documentation, specifically Summary Plan Descriptions (SPDs) and Benefit Plan Summaries (BPS). Leveraging advanced Retrieval-Augmented Generation (RAG) technology, SmartSPD provides instant, accurate, and context-aware answers to natural language questions, significantly reducing research time, improving service quality, and enhancing user satisfaction. The product aims to transform dense, often inaccessible health plan information into an easily queryable and understandable format, making benefits information transparent and accessible to all stakeholders.

2. Vision & Goals

2.1. BeneSense AI Company Vision

To empower individuals and organizations with intelligent, intuitive AI solutions that simplify complex information, enhance decision-making, and foster greater understanding and efficiency in critical sectors like healthcare.

2.2. SmartSPD Product Vision

To be the leading AI-driven platform for health plan information retrieval, setting a new standard for accuracy, accessibility, and efficiency in benefits communication and customer service.

2.3. Business Goals

- **Increase Efficiency**: Reduce average time spent by customer service agents researching health plan details by 50% within 12 months of launch.
- **Improve Accuracy**: Achieve a 95% accuracy rate in responses to benefits-related queries within 6 months of launch.
- **Enhance User Satisfaction**: Increase user satisfaction scores (e.g., CSAT) for benefits inquiries by 20% within the first year.
- **Market Penetration**: Secure 5 enterprise health plan clients within the first 18 months.
- **Scalability**: Support seamless integration and processing for over 1,000 unique health plans and millions of documents.
- **Revenue Growth**: Generate $X million in recurring revenue within 2 years.

2.4. User Goals

- **Customer Service Agents**: Quickly find precise answers to member questions, reduce call handling times, and improve first-call resolution rates.
- **Health Plan Members**: Easily understand their benefits, deductibles, and coverage details without navigating complex documents or waiting for agent assistance.
- **HR Professionals/Brokers/TPAs**: Access specific plan details for enrollment, compliance, and client support efficiently.

- **Administrators**: Easily upload, manage, and monitor health plan documents and system performance.

3. Target Audience

3.1. Primary Users

- **Customer Service Agents**: Front-line support staff who answer member inquiries daily.
- **Health Plan Members**: Individuals covered by health plans seeking information about their benefits.
- **HR Professionals/Brokers/TPAs**: Professionals who manage employee benefits, advise clients, or administer plans.

3.2. Secondary Users

- **BeneSense AI Administrators**: Internal team members responsible for system maintenance, monitoring, and content management.
- **Developers**: Team members responsible for extending and maintaining the SmartSPD platform.

3.3. User Personas (Examples)

Persona 1: Agent Amy

- **Role**: Health Plan Customer Service Agent
- **Needs**: Fast, accurate answers to diverse member questions; ability to cite sources; reduce call times.
- **Pain Points**: Sifting through lengthy PDFs; inconsistent information; long hold times for members.
- **Goal with SmartSPD**: Provide exceptional service efficiently.

Persona 2: Member Mark

- **Role**: Health Plan Member
- **Needs**: Understand what his deductible is, if a specific procedure is covered, or how to file a claim.
- **Pain Points**: Confusing jargon; difficulty finding information in large documents; frustration with call center wait times.
- **Goal with SmartSPD**: Get clear, immediate answers about his benefits.

Persona 3: Admin Alex

- **Role**: BeneSense AI System Administrator
- **Needs**: Upload new health plan documents; monitor system performance; manage user access.
- **Pain Points**: Manual document processing; lack of visibility into system health; complex user management.
- **Goal with SmartSPD**: Efficiently manage the SmartSPD platform and its content.

4. Scope & Features

4.1. In-Scope Features

4.1.1. Core AI-Powered Q&A (Smart Agent)

- **Natural Language Understanding**: Users can ask questions in conversational English.
- **Retrieval-Augmented Generation (RAG)**: System retrieves relevant information from indexed SPDs (PDF) and BPS (Excel) documents.
- **Contextual Answers**: AI generates concise, accurate answers based on retrieved content.
- **Source Attribution**: Every answer includes direct citations/references to the specific document, page number, and section from which the information was retrieved.

- **Confidence Scoring**: A visual indicator (e.g., percentage, color-coded) of the AI's confidence in its answer.
- **Conversation Memory**: The Smart Agent maintains context within a conversation, allowing for follow-up questions.
- **Hybrid Search**: Combines keyword search with semantic search for optimal retrieval accuracy.

4.1.2. Document Ingestion & Processing

- **PDF Parsing**: Ability to ingest SPD documents in PDF format.
  - **Intelligent Chunking**: Automatically identifies and separates text blocks, tables, and mixed content.
  - **Table Context Preservation**: Tables are recognized and processed to maintain their structural and semantic meaning.
  - **Metadata Extraction**: Extracts key metadata (e.g., document title, plan name, version) during ingestion.
- **Excel Parsing**: Ability to ingest BPS documents in XLSX/XLS format.
  - **Structured Data Extraction**: Accurately extracts benefit details, co-pays, deductibles, and other structured data.
  - **Cross-referencing**: Links Excel data to relevant PDF sections where applicable.
- **Embedding Generation**: Uses a specialized embedding model to create vector representations of document chunks.
- **Vector Database Storage**: Stores embeddings and associated metadata (chunk type, source document, page) in a robust vector database (e.g., ChromaDB).
- **Knowledge Graph Population**: Extracts entities and relationships from documents to build a knowledge graph, enhancing retrieval and reasoning.
- **Enhanced Ingestion Process**: Includes robust error handling, progress tracking, and detailed logging during document upload and processing.

4.1.3. User Interface (Frontend)

- **Intuitive Chat Interface**: iPhone-style messaging UI for natural interaction with the Smart Agent.
- **Responsive Design**: Optimized for seamless experience across desktop, tablet, and mobile devices.
- **Login/Authentication**: Secure user authentication system.
- **Role-Based Access Control (RBAC)**: Different user roles (e.g., Member, Agent, HR, Admin) have appropriate access levels and views.
- **Landing Page**: Professional, branded sales page for BeneSense AI and SmartSPD, leading to login.
- **Suggested Questions**: Provides dynamic, context-aware suggested questions to guide users.
- **Feedback Mechanism**: Users can provide feedback on answer quality.

4.1.4. Admin Dashboard

- **Document Upload**: Secure interface for administrators to upload new SPD (PDF) and BPS (Excel) files.
  - Requires specifying `health_plan_name`, `health_plan_code`, and `document_type`.
- **Document Management**: View a list of all uploaded documents, their status, and associated metadata.
  - Ability to delete documents and their associated chunks/embeddings.
- **Health Plan Management**: Create, view, and manage health plan configurations (e.g., plan name, code, type, description).
- **System Overview**: Dashboard displaying key system statistics (e.g., total documents, total health plans, total chunks, query volume, average response time, storage usage).
- **User Management (Basic)**: View and manage user accounts (roles, status).

4.1.5. Backend & Infrastructure

- **Flask API**: Robust and scalable backend API built with Flask.
- **Database**: Relational database (e.g., SQLite for local, PostgreSQL for production) for user data, document metadata, and health plan configurations.
- **Vector Database**: ChromaDB (or similar) for storing embeddings.
- **Caching Service**: Implement multi-tier caching (e.g., Redis) for frequently accessed data and query results to improve performance.
- **Logging & Monitoring**: Comprehensive logging for system events, errors, and performance metrics.
- **Security**: JWT-based authentication, secure API endpoints, input validation, and adherence to HIPAA compliance principles (data segregation, access controls).

4.2. Out-of-Scope Features (for v1.0)

- Real-time document updates (requires complex change detection).
- Multi-language support (English only for v1.0).
- Advanced analytics beyond basic system stats.
- Direct integration with external CRM/ticketing systems (API available for future integration).
- Automated document versioning and comparison.
- Complex workflow automation (e.g., approval processes for document ingestion).

5. User Flows

5.1. User Authentication Flow

1. User navigates to SmartSPD landing page.
2. User clicks

5. User Flows

5.1. User Authentication Flow

1. User navigates to SmartSPD landing page.
2. User clicks "Get Started" or "Login".
3. User enters credentials (email/password) or uses a demo account.
4. System authenticates user and issues a JWT token.
5. Based on user role, redirects to either the Chat Interface (Member, Agent, HR, Broker, TPA) or Admin Dashboard (Admin).

5.2. General User (Member/Agent) Q&A Flow

1. User logs in and lands on the Chat Interface.
2. User types a natural language question (e.g., "What is my deductible for out-of-network services?").
3. (Optional) User selects a suggested question.
4. Frontend sends the query to the backend `/api/chat/query` endpoint.
5. Backend RAG service:
   a. Generates embeddings for the query.
   b. Performs hybrid search (semantic + keyword) against the vector database and knowledge graph.
   c. Retrieves relevant document chunks (text, tables) from indexed SPDs/BPS documents.
   d. Sends retrieved context and query to the LLM (Gemini 2.5 Pro).
   e. LLM generates a concise answer, citing sources (document ID, page, section).
   f. Backend calculates a confidence score.
6. Backend sends the answer, sources, and confidence score back to the frontend.
7. Frontend displays the answer in the chat interface.

8. User can ask follow-up questions, provide feedback, or start a new conversation.

5.3. Admin User Document Upload Flow

1. Admin logs in and lands on the Admin Dashboard.
2. Admin navigates to the "Upload Documents" tab.
3. Admin fills in required metadata: Health Plan Name, Health Plan Code, Document Type (SPD/BPS).
4. Admin selects a PDF or XLSX/XLS file.
5. Admin clicks "Upload and Process".
6. Frontend sends the file and metadata to the backend `/api/admin/upload` endpoint.
7. Backend:
   a. Validates file type and size.
   b. Saves the file temporarily.
   c. Initiates document processing (parsing, chunking, embedding, knowledge graph population).
   d. Updates document status in the database.
8. Backend returns a success/failure message with processing details.
9. Frontend displays upload progress and result.
10. Document appears in the "Manage Documents" tab upon successful processing.

6. Technical Requirements

6.1. Backend

- **Language/Framework**: Python 3.10+, Flask
- **Web Server**: Gunicorn (for production deployment)
- **Database**: SQLAlchemy ORM with SQLite (development), PostgreSQL (production)

- **Vector Database**: ChromaDB (or similar, e.g., Pinecone, Weaviate)
- **LLM Integration**: Google Gemini 2.5 Pro API (or other suitable LLMs like OpenAI GPT-4o, Claude 3 Opus)
- **Embedding Model**: `all-MiniLM-L6-v2` (or a more specialized embedding model)
- **Caching**: Redis (for session management, query caching)
- **Task Queue**: Celery (for asynchronous document processing)
- **Authentication**: PyJWT, Bcrypt
- **CORS**: Flask-CORS
- **Logging**: Standard Python logging, integrated with a monitoring solution.

6.2. Frontend

- **Framework**: React.js
- **Language**: JavaScript (ES6+), JSX
- **Build Tool**: Vite
- **Styling**: Tailwind CSS, Shadcn/ui components
- **State Management**: React Context API or Zustand/Jotai
- **Routing**: React Router DOM
- **API Client**: Fetch API or Axios

6.3. Infrastructure & Deployment

- **Development Environment**: GitHub Codespaces, Replit
- **Production Environment**: Cloud platform (e.g., AWS, GCP, Azure) with Docker/Kubernetes for containerization.
- **CI/CD**: GitHub Actions or GitLab CI/CD for automated testing and deployment.

- **Monitoring**: Prometheus/Grafana for system metrics, Sentry/ELK Stack for error logging.

6.4. Performance Requirements

- **Query Response Time**: Average < 2 seconds for chat queries (excluding initial cold start).
- **Document Processing**: SPD (100 pages) processed within 5 minutes; BPS (1000 rows) processed within 1 minute.
- **Scalability**: Support 100 concurrent users for chat interface; ability to scale document processing workers.
- **Uptime**: 99.9% monthly uptime for core services.

6.5. Security Requirements

- **Authentication**: Secure JWT-based authentication with hashed passwords.
- **Authorization**: Role-Based Access Control (RBAC) for all API endpoints and UI elements.
- **Data Encryption**: Data in transit (HTTPS/SSL/TLS) and at rest (database encryption).
- **Input Validation**: All user inputs and file uploads must be validated to prevent injection attacks.
- **HIPAA Compliance**: Design and implement with HIPAA principles in mind, especially for PHI (Protected Health Information) handling, access controls, and audit trails.
- **Vulnerability Management**: Regular security audits and penetration testing.

6.6. Data Storage Requirements

- **Document Storage**: Secure, scalable object storage (e.g., AWS S3, Google Cloud Storage) for raw PDF/Excel files.
- **Database**: Store user profiles, health plan metadata, document metadata, chunk metadata, and conversation history.

- **Vector Store**: Optimized for high-dimensional vector search.

7. Open Questions & Dependencies

7.1. Open Questions

- Specific LLM provider and model to be finalized (Gemini 2.5 Pro is current preference).
- Exact schema for knowledge graph entities and relationships.
- Detailed UI/UX wireframes for complex interactions.
- Strategy for handling document updates/revisions (versioning).
- Specific metrics for confidence scoring and how it will be calibrated.

7.2. Dependencies

- Access to LLM APIs (e.g., Google Cloud, OpenAI).
- Cloud infrastructure accounts (AWS, GCP, Azure).
- Design assets (logos, branding guidelines, specific UI components).
- Legal review for HIPAA compliance and data privacy.

8. Future Considerations (Post v1.0)

- **Multi-language Support**: Expand to other languages for global reach.
- **Advanced Analytics**: Detailed dashboards for query performance, user engagement, and document utilization.
- **Integration with CRM/EHR**: Seamless data exchange with existing healthcare systems.
- **Proactive Insights**: AI-driven alerts or summaries based on document changes or user queries.
- **Voice Interface**: Integration with voice assistants for hands-free interaction.

- **Automated Document Ingestion**: Web scraping or direct API integrations for automated document retrieval.
- **Customizable Workflows**: Allow clients to define custom workflows for document processing or query routing.

9. Success Metrics

- **Accuracy**: Percentage of correct answers provided by the Smart Agent.
- **Response Time**: Average time taken to respond to a user query.
- **User Satisfaction**: CSAT/NPS scores from users.
- **Document Coverage**: Percentage of health plan documents successfully processed and indexed.
- **Adoption Rate**: Number of active users and health plans utilizing the system.
- **Cost Efficiency**: Cost per query, cost per document processed.

10. Appendices

10.1. Glossary

- **SPD**: Summary Plan Description
- **BPS**: Benefit Plan Summary
- **RAG**: Retrieval-Augmented Generation
- **LLM**: Large Language Model
- **API**: Application Programming Interface
- **UI**: User Interface
- **UX**: User Experience
- **JWT**: JSON Web Token

- **RBAC**: Role-Based Access Control
- **PHI**: Protected Health Information
- **CI/CD**: Continuous Integration/Continuous Deployment

10.2. References

- `smartspd_flow.html` (Provided by client)
- Existing SmartSPD Replit project code
- BeneSense AI branding guidelines (assumed)

This PRD serves as a living document and will be updated as the project evolves and new requirements emerge. All stakeholders should refer to this document as the primary source of truth for SmartSPD development.