Product Requirements Document (PRD) – AI Disability Claims Advisor (Israel)

Project Name: AI National Insurance Disability Claims Advisor
Version: 1.9 (Enhanced Admin Features, Communication, and Funnel Tracking Added)
Date: 2025-10-09
Author: Gemini AI
Status: Ready for Development Team Briefing

1. Introduction and Goals

1.1 Project Overview

This project aims to develop a web application for an Israeli law firm specializing in National Insurance (Bituach Leumi – BTL) disability claims. The core innovation is the **Agent Strategy Layer**, which acts as a professional BTL lawyer, guiding the claimant through the entire process using specialized sub-agents. The entire backend logic and agent orchestration will be managed via **n8n workflows**.

1.2 Business Objectives

1. **Increase Efficiency:** Reduce the manual time spent by legal staff analyzing initial medical documentation by 70%.
2. **Improve Claim Accuracy:** Ensure consistent and legally compliant application of BTL rules and precedents.
3. **Optimize Conversion:** Increase user conversion to paying clients by providing data-driven trust (Monetization Screen).

2. Core Features and User Flow

2.1 User Flow (Claimant - Updated Sequence)

The payment step has been moved to occur only after the Core Analysis confirms the claim's initial viability, justified by the Monetization Screen.

Step	Action	Description	System Component
1. Landing Page	User enters the site.	Overview of services and Call to Action (CTA) for eligibility check.	Frontend (Replit)

2. Eligibility Check	User answers a short qualification questionnaire.	Determines basic claim eligibility before formal intake.	Frontend (Replit), Custom Logic
3. Intake & ID Upload	User registers and provides ID.	Secure registration, but **no payment is taken yet.**	DB
4. Document Upload	User uploads medical documents (PDFs, images of reports).	Secure file upload portal.	Frontend (Replit), Storage
5. Core Analysis	Automated process extracts data, analyzes BTL codes, and calculates preliminary findings.	**n8n (Workflow)** + **Gemini API** + Custom BTL Logic	
6. Monetization Screen (Value Proposition) (NEW)	**System displays the claim's viability score and projected percentage range.**	Frontend (Replit), Data Display	
7. Payment & Commitment (MOVED)	User pays the fixed consultation fee via Hyp. **This payment triggers the full agent engagement.**	Hyp Payment Integration, DB	
8. Interactive Q&A	System asks up to 5 questions to clarify ambiguous data.	Gemini API (Conversation Mode), Frontend Form	
9. Agent Strategy Control	The BTL Lawyer Agent takes control and routes the user	**Agent Strategy Layer** (n8n/Gemini)	

	to the next appropriate specialized sub-agent.		
10. Sub-Agent Workflow	User engages with specialized Agents (Submission, Prep for Doctor, Prep for Committee, Appeal, or Vocational Rehab).	Specialized Sub-Agent Modules	

2.2 Feature Requirements (Claimant Facing)

2.2.1 AI Document Analysis and Extraction (Gemini API)

- **OCR & Text Extraction:** Process uploaded PDFs and images and extract raw text.
- **Named Entity Recognition (NER) & Classification:** Identify and extract key medical entities (Diagnosis, Dates, Functional Limitations) in Hebrew.
- **BTL Code Mapping:** Map extracted findings to relevant BTL regulation items.
- **Preliminary Percentage Calculation (Core Logic):** Apply complex BTL combination rules to calculate a **projected total disability percentage and an overall viability score (High/Medium/Low)**. This score is required for the Monetization Screen (2.2.5).

2.2.2 Interactive Q&A Module (Gemini API)

- **Limit: Maximum of 5 dynamic questions** generated by the AI based on data gaps or ambiguity from the analysis.
- **Goal:** Increase the legal viability and monetary value of the claim.

2.2.3 Agent Strategy Layer (BTL Lawyer Persona)

- **Definition:** The main orchestration engine. It maintains a professional, knowledgeable persona.
- **Routing Logic:** The Agent analyzes the case readiness and automatically routes the claimant to one of the specialized sub-agents.

2.2.4 Specialized Sub-Agents

Agent Name	Goal	Required Functionality
1. Initial Claim Submission Agent	Ensure the initial BTL claim form is filed perfectly and compliantly.	**Form Pre-fill:** Generate a pre-filled BTL claim form. **Attachment Checklist:**

		Verify all necessary documents.
2. Expert Doctor Prep Agent	Prepare the user for specialist medical appointments.	**Prep Script Generation:** Generate custom, BTL-focused talking points.
3. Medical Committee Prep Agent	Prepare the user optimally for the BTL Medical Committee hearing.	**Simulation Q&A:** Provide simulated BTL Committee questions.
4. Appeal & Reconsideration Agent	Manage the post-decision process by automatically generating legal grounds to challenge unfavorable BTL decisions.	**Grounds Generation:** Create a structured document outlining the legal/medical grounds for appeal based on precedent.
5. Vocational Rehabilitation Agent	Maximize client benefits and rights from the vocational rehabilitation program.	**Trigger Condition:** The system confirms either **Medical Disability OR Loss of Earning Capacity** from the BTL decision. **Benefit Optimization:** Identify the maximal financial and non-financial rights. **Automated Forms:** Generate draft letters/requests.

2.2.5 Monetization & Payment System (Hyp Integration)

- **Monetization Screen Logic (NEW):** A dedicated screen must display the key outputs from the Core Analysis (Step 5):
  - **Viability Score** (e.g., "High Probability of Success").
  - **Projected Disability Percentage Range** (e.g., "Expected 30% - 45%").
  - A compelling Call to Action (CTA) explaining the fixed consultation fee and what the user gets in the full Agent Strategy phase.
- **Consultation Fee (CRITICAL):** The fixed, upfront payment (using Hyp's API) must be collected on this screen **after** the user accepts the value proposition and **before** proceeding to Step 8 (Interactive Q&A).
- **Payment Gate:** If the Core Analysis (Step 5) returns a **"Very Low"** or **"Non-Viable"** score, the system must not proceed to the Monetization Screen, but instead provide a

rejection/referral message, avoiding payment.

2.3 Admin Portal Features (Human-in-the-Loop - HITL)

This section defines the features required for the legal staff and administrators to manage claims and ensure quality control over the AI's output.

2.3.1 Claims Dashboard and Management (Enhanced Tracking)

- **Claims Queue:** A filterable list of all claims with their current status (e.g., Pending Analysis, Ready for Monetization, Waiting for Payment).
- **Search/Filter:** Ability to search claims by Claimant ID (T.Z.), Claimant Name, and Status.
- **Funnel Progress Tracking (NEW):** Display the claimant's current position in the funnel (e.g., "Intake", "Core Analysis Complete", "**Payment Pending**", "Agent Q&A").
- **Payment Status Monitoring (NEW):** Display the Hyp transaction status for each claim (e.g., "Pending", "Paid", "Failed").
- **Failure Point Analysis (NEW):** Provide aggregate data and filtering options to **identify common bottlenecks or drop-off points** in the user flow (e.g., of users drop off at the "Document Upload" stage).

2.3.2 HITL Review Interface (Core Logic Validation)

- **Access Point:** Admins must be able to access the review interface immediately after the AI completes **Step 5 (Core Analysis)** and **before Step 6 (Monetization Screen)**.
- **Data Display:** The interface must show:
  1. Raw text extracted from all uploaded documents.
  2. The structured JSON output generated by the Gemini API (Section 3.3).
  3. The calculated and **Viability Score** (Section 3.2.3).
- **Manual Data Correction (CRITICAL HITL):** Admins must be able to **manually edit** the BTL code, description, and **medical_percentage** for any impairment in the list.
- **Recalculation:** After any manual edit, the system must **immediately re-run the BTL Combined Percentage Algorithm** (Section 3.2) and update the final and Viability Score.
- **Override:** Ability to manually override the viability status (e.g., change from 'Low' to 'High') before the user sees the Monetization Screen.

2.3.3 Client Communication and Management (NEW SECTION)

- **WhatsApp Quick Contact (CRITICAL):** A prominent, one-click button/link on the claim's detail page to initiate a WhatsApp chat with the claimant's registered phone number.
- **Request Additional Documents Workflow:**
  - An interface allowing the admin to select from a predefined list of missing documents (e.g., "Neurological Assessment," "Recent Blood Tests").
  - This selection must trigger a dedicated **n8n workflow** that sends a custom notification (Email/SMS/WhatsApp, based on user preference) to the claimant, pausing the claim's progress until the documents are uploaded.
- **Meeting Scheduling Integration:** A tool or link integrated into the claim's detail page that allows the admin to **coordinate and schedule a meeting** (in-person or virtual) and

automatically send a calendar invite to the claimant.

2.4 Dynamic Prompt Management Tab

The Admin Portal must include a dedicated tab for authorized users to view and edit the AI's instructional prompts.

- **Editable Prompts:** The interface must allow editing of the following system prompts (minimum):
  1. **Core Analysis System Prompt** (Section 3.4.1 - The "BTL Expert" persona).
  2. **Interactive Q&A Prompt** (The prompt guiding question generation in Step 8).
  3. **Agent Strategy Layer Persona Prompt** (The initial persona definition for the main Agent).
  4. **All Sub-Agent Persona Prompts** (e.g., Appeal Agent, Doctor Prep Agent).
- **Saving to Firestore:** Edits must be saved to the **prompt_configs** collection in Firestore (see updated Section 3.1.3).
- **Dynamic Loading:** All n8n workflows making Gemini API calls **MUST** load the current prompt text from this Firestore collection at the start of the workflow execution, ensuring changes take effect immediately without code deployment.

3. Technical Stack and Implementation

Component	Technology	Description
Frontend/App Hosting	Replit / HTML/React/Angular + Tailwind CSS	Single-page application (SPA). Must be fully responsive. **Includes Admin Portal.**
Backend/Gateway	Replit (Node.js/Python environment)	Handles file uploads, user authentication, and triggers the n8n workflow.
AI/NLP	Gemini API (gemini-2.5-flash-preview-05-20)	Used for text extraction, NER, and dynamic Q&A generation.
Workflow Automation	n8n	Orchestrates the entire backend process (API calls, data processing, DB updates, notifications).
Payment Processing	Hyp API	Used for secure credit card transactions and payment

		method tokenization.
Database	Google Firestore	Persistent storage for user profiles, claim analysis JSON, **and dynamic prompt configuration.**
Authentication	Firebase Auth (using __initial_auth_token)	Handles user authentication.

3.1 n8n Workflow Design (Implementation Instructions)

The entire backend operation relies on a primary n8n workflow to handle data, logic, and agent routing.

3.1.1 Main Workflow Structure (The Agent Strategy Layer)

The main workflow should be triggered by an external HTTP Webhook (triggered by the Frontend after document upload, Step 4/5).

1. **Start Node:** WebHook (Receives userId and document file path from the Frontend).
2. **Prompt Configuration Load:**
   - Firestore Node: Load the active **Core Analysis System Prompt** from the prompt_configs collection.
3. **Document Processing Branch:**
   - Code Node (OCR/File Prep): Retrieve the document from storage, convert it (if necessary), and prepare the data/base64 for the Gemini API call.
   - HTTP Request (Gemini API): Call the Gemini API to perform **OCR, NER, and BTL Code Mapping** (Section 2.2.1), using the dynamically loaded prompt from Step 2. The prompt must include the BTL.pdf context for legal grounding and **MUST** return data according to the schema in **Section 3.3**.
   - Code Node (BTL Logic): Execute the complex mathematical logic for BTL combined percentage calculation (detailed in Section 3.2). This module determines the **Projected Disability Percentage** and **Viability Score**.
4. **Monetization Gate:**
   - IF Node: Check the calculated **Viability Score**.
     - **Path A (Viable/High/Medium):** Proceed to update Firestore with the scores and send a response back to the Frontend to display the **Monetization Screen** (Step 6).
     - **Path B (Very Low/Non-Viable):** Send a rejection/referral message back to the Frontend.
5. **Payment Listener:** A separate, secondary webhook is triggered by the Frontend **after successful Hyp payment** (Step 7). This webhook retrieves the claim data and proceeds to the full agent engagement.

6. **Interactive Q&A:**
   - HTTP Request (Gemini API): Call the Gemini API to generate the **up to 5 dynamic questions** (Section 2.2.2), loading the relevant prompt from Firestore. Store questions in Firestore.
7. **Agent Orchestration:**
   - Code Node (Routing Logic): Based on the current stage of the claim (e.g., initial submission, waiting for committee, received decision), this node determines which **Specialized Sub-Agent** workflow to call next.

3.1.2 Sub-Agent Workflow Implementation (Specialized Services)

Each of the five specialized sub-agents (Section 2.2.4) should be implemented as a **separate, callable n8n workflow**. Each sub-agent workflow **MUST** load its specific system prompt from the prompt_configs collection before executing its Gemini API call.

3.1.3 Data Management (Firestore)

- **Private Data:** All claimant details, documents, and analysis results must be stored privately: /artifacts/{__app_id}/users/{userId}/claims/{claimId}.
- **Public Data (Prompt Configs):** All dynamically editable system prompts must be stored in a **public collection** for easy access by both the Admin Frontend and the n8n workflows: /artifacts/{__app_id}/public/data/prompt_configs/{promptName}.
  - Example document ID: core_analysis_system_prompt.
  - Example document content: { "text": "תוכן הפרומפט המעודכן..." }.

3.2 Custom BTL Logic Specification (Code Node - Critical for Step 5)

This section defines the precise logic required for the Code Node (BTL Logic) within the main n8n workflow (Section 3.1.1, Step 3). This code handles the complex combination rules for multiple disabilities.

3.2.1 Input Data Structure

The code node receives a JSON array representing the identified impairments and their determined medical percentages (extracted and normalized by the previous Gemini API call, Section 3.3).

```
[
  {
    "btl_code": "04.01",
    "description_he": "הגבלה קלה בפרק",
    "medical_percentage": 10
  },
  {
    "btl_code": "06.01",
    "description_he": "ליקוי שמיעה קל",
    "medical_percentage": 25
  },
  {
    "btl_code": "34.02",
    "description_he": "הפרעת קשב וריכוז (ADHD)",
    "medical_percentage": 50
  }
]
```