

Smart Steps Platform: Patent Features vs. Implementation Status

Overview: The Smart Steps platform is envisioned in a provisional patent as a **psychological assessment tool** supporting practitioner-client sessions with scenario-based exercises, guided AI dialogue, and rigorous analytics ¹ ². The following report examines each major feature from the patent and analyzes the current GitHub repository (SmartSteps_PROD) to determine which features are **fully implemented, partially developed, or missing**. We focus on the three-phase process (scenario → query → evaluation), the AI persona module (LLM integration), data privacy (anonymous usage), role-based design (practitioner vs. client), and extensibility (VR/voice integrations). A summary comparison table is provided after the detailed analysis.

Scenario-Based Interaction Process (Phase 1: Scenario)

Patent Description: The system's first phase centers on **scenario-based interactions**. Clients engage with *structured scenarios* that present decision points and dilemmas to prompt discussion and self-reflection ³. Scenarios may be delivered via an intuitive interface and can include immersive elements (even VR) for realistic practice ². This phase is meant to elicit emotional and cognitive responses which will later be evaluated.

Current Implementation: The repository shows **early or fragmented development** of scenario delivery. In an older web prototype, there is a *scenario selection interface* with sample scenarios ("The Meeting," "The Wallet," "The Argument") and a "Launch Session" button ⁴. Choosing a scenario loads a title, description, and an image, and allows the user to select response options ⁵ ⁶. This suggests the team has **conceptual scenario content** prepared (including images and descriptions) and basic UI workflow for scenario presentation. However, in the **new Unity-based application**, we find no fully implemented scenario module. The Unity project's scripts focus on UI infrastructure and the AI chat system, but **lack code for branching scenario logic** or content delivery. There is no evidence of a library of scenario scripts or a completed mechanism to present dilemmas and record user choices in Unity – the scenario phase appears **not yet integrated** into the main app. The presence of planning documents (e.g. *STEP_PLAN.md*) and a prototype UI indicates the scenario concept exists, but it remains **partially implemented** (some static content and design) and not fully functional within the latest architecture. This is a development gap: the patent's rich scenario experience is not yet realized in the production code.

Query Phase (Phase 2: Evaluative Questions)

Patent Description: After each scenario, the second phase involves *tailored evaluative queries* to encourage reflection ⁷. These questions can be posed by the human practitioner or delivered **automatically by the system** (AI) to probe the client's thoughts and feelings about the scenario ⁸. The intent is to deepen engagement and gather qualitative data on the client's reasoning and emotional responses.

Current Implementation: The platform's **conversational system** provides a means to conduct this query phase, though not as a distinct automated module. The **AI Persona module** effectively functions as the system's voice to ask and answer questions in natural language. In the repository, each AI persona is configured to engage users with a high frequency of questions. For example, the *persona profile* for "Dr. Morgan Hayes" (a therapist persona) includes a conversation style with a "question_frequency": "high" and typical prompts like "How does that feel for you?" and "What strategies have helped you in similar situations?"⁹. Similarly, the behavioral analyst persona uses probing questions (e.g., "What preceded this behavior?") as part of its typical phrases¹⁰. This indicates the AI is set up to **prompt the client with reflective questions**, fulfilling the automated query role.

On the **manual side**, the system supports practitioner-driven queries and discussion. The facilitator (practitioner) can always directly ask the client questions through the session, either in person or via the app's interface. The software's design – with a facilitator dashboard and session panel – implies that the human facilitator guides the interaction and can input their own questions or select prompts as needed¹¹¹². However, we did not find a dedicated interface for pre-scripted "post-scenario" question lists or a logic that triggers a set of queries right after a scenario completes. In practice, **any follow-up queries would occur through the chat interface**: either the practitioner enters them or the AI persona organically asks.

Conclusion: The capability for a **query phase is present in a basic form**. The AI personas are explicitly designed to ask insightful questions (covering the "system automation" case) and the facilitator can manually query the client at any time. What's missing is an explicit coupling between scenario completion and a *structured set* of follow-up questions. In other words, the platform can handle Q&A dialogue, but it has **no special logic implementing a "scenario complete – now ask these evaluation questions" sequence** yet. This portion of the patent is only **partially realized**: dynamic dialogue is implemented, but tailored scenario-specific questionnaires are not clearly in place.

Evaluation and Analysis Logic (Phase 3: Evaluation Metrics)

Patent Description: The third phase involves evaluating the client's performance and growth using **psychometric analysis**¹³¹⁴. The patent emphasizes use of models like the **Rasch model**¹⁵¹⁶ to quantify social-emotional attributes (self-awareness, self-management, social awareness, relationship skills, decision-making, self-disclosure) before and after the program. Both client self-reports and practitioner observations contribute to this evaluation, producing baseline, growth, and final measures¹⁷. The system should generate numeric scores and qualitative insights, and even suggest personalized interventions based on the results¹⁸¹⁹.

Current Implementation: The repository contains an **analysis module**, but it currently provides *basic or placeholder functionality* rather than advanced psychometrics. On the positive side, there is a `SessionAnalyzer` class that collates session data and produces a report with a summary, insights, metrics, themes, recommendations, and next steps²⁰²¹. This shows the architecture is ready to generate a comprehensive report after a session. The insights generation uses AI: the `InsightGenerator` calls an LLM provider (OpenAI/Anthropic) to derive insights in categories like emotional, cognitive, behavioral, progress, etc.²²²³. This aligns with the patent's notion of interpreting participant responses and extracting meaning.

However, the **quantitative psychometric aspect is minimal so far**. The code does not implement Rasch modeling or any item-response theory calculations. In fact, searching the code finds **no mention of “Rasch” or similar psychometric algorithms**. Instead, the `SessionAnalyzer` uses simple heuristics for metrics – e.g., an “Engagement” metric calculated as `min(session.messages_count / 10, 10.0)` (a rough score based on number of messages) ²⁴. Other metrics like “Session Progress” are hard-coded placeholders ²⁵. These are *proxies* rather than data-driven psychometrics. The `SessionData` model does define fields like `EffectivenessRating`, `EngagementLevel`, and `ProgressRating` on a 1–10 scale ²⁶, but these appear to be either manually assigned or set by simple logic, not by an advanced analysis engine. There’s no evidence of pre/post assessment instruments or standardized scoring as described in the patent.

The **evaluation outputs** in the current system are therefore largely narrative and basic metrics. The AI-generated session **summaries and insights** provide qualitative evaluation (e.g. noting patterns or challenges) ²⁷ ²⁸. The numeric “metrics” are present but not grounded in rigorous psychometric models. Additionally, while the patent describes two-level evaluation (client self-report vs practitioner observation), the software does not explicitly implement this distinction. It could be that the facilitator’s notes and any client feedback are simply part of the session data, without a formalized two-level scoring system coded.

Conclusion: The **infrastructure for evaluation is in place**, but the *depth of analysis is far from the patent’s aspirations*. The current repo implements a **basic analytics pipeline**: capturing interaction data, summarizing with AI, and outputting a simple progress report. It **does not yet perform standardized psychometric analysis** (no Rasch model or similar statistics). This feature is **partially developed** – the framework exists to add such models, but actual implementation of rigorous evaluation metrics is **missing** at present. This discrepancy is a notable inconsistency between the patent’s claims of precise measurement and the software’s current capabilities.

AI Persona Module (LLM-Assisted Dialogue & Analysis)

Patent Description: A key innovative aspect is the integration of **AI-driven personas powered by large language models (LLMs)** ²⁹. The patent suggests that an LLM will *guide scenario navigation, provide adaptive feedback*, and enrich the dialogue between practitioner and client ³⁰. Over time, anonymized interaction data could even train a specialized LLM tailored to social-emotional learning contexts ³¹. In short, the platform’s AI should act as a virtual facilitator or participant in sessions, enhancing engagement and insight.

Current Implementation: The **AI Persona module is the most developed component** of the repository, closely aligning with the patent vision of LLM-assisted interaction. The system already supports multiple personas, each defined with rich profiles and prompts. For example, the repository includes persona configuration files for several therapist AIs (e.g., **Dr. Morgan Hayes, CBT specialist**; **Dr. Alex Rivera, behavioral analyst**; likely others for motivational interviewing and trauma-informed care) ³² ³³. Each persona JSON defines the AI’s *name, background, style, and rules*. Dr. Morgan Hayes’s profile illustrates this: a system prompt establishes her 15 years of CBT experience, empathetic yet professional tone, and guidelines like *“never diagnose conditions”* but discuss symptoms and coping strategies ³⁴ ³⁵. Trait scores (empathy, analytical, etc.) and typical phrases are included to shape the dialogue style ³⁶ ⁹. This level of detail ensures the AI persona responds in a manner consistent with a human therapist of that profile.

The **dialogue system** around these personas is implemented via an LLM provider abstraction. The code integrates with OpenAI and Anthropic APIs for chat completions ³⁷ ³⁸. The *ProviderManager* can select which model to use (e.g., GPT-4, Claude) or use a Mock provider for testing ³⁹ ⁴⁰. When a user sends a message, the backend constructs a prompt with the persona's system role and the conversation history, then calls the LLM API to generate the persona's next response. The Unity front-end connects to this via the `SmartStepsApiClient` – for example, `CreateSession` and `SendMessage` endpoints are exposed to initiate sessions and exchange messages with the AI ⁴¹ ⁴². Real-time aspects are considered too: there's a WebSocket router for streaming or live updates ⁴³ (likely to stream token-by-token responses from the LLM).

Beyond conversation, the AI module contributes to **analysis**. The `InsightGenerator` uses the LLM (with specialized prompts) to analyze a session transcript and produce insights (categorized e.g. “emotional insight” or “behavioral pattern”) ⁴⁴ ⁴⁵. This means the AI is not only chatting but also **assisting in post-session evaluation** by identifying themes and making observations, much like a co-facilitator might.

Importantly, the current implementation relies on external LLMs (OpenAI/Anthropic) with no evidence of a custom-trained model yet. The patent's notion of training a proprietary LLM on anonymized data remains *aspirational* – for now, the system is using third-party models with carefully crafted prompts. Nonetheless, the design is modular, so a future fine-tuned model could be plugged in via the provider interface.

Conclusion: The **AI persona functionality is largely implemented and is a strong point** of the project. The repository successfully realizes the patent's claim of an LLM-guided dialogue: multiple specialized personas can carry on conversations and adapt to user input, providing therapeutic feedback in real time. The AI also aids analysis by summarizing and highlighting key points from sessions. This module aligns well with the patent (minus the yet-to-be-realized custom model training). It represents an area of strength in the codebase, having substantial development effort and complexity already in place.

Anonymous Login & Data Handling

Patent Description: To encourage honest participation, the patent proposes **anonymous login and data anonymization** ⁴⁶ ³¹. Users (particularly clients) would not need to provide personal identifying information to use the system ⁴⁶. All interaction data would be stored without PII, and analyses would be done on anonymized data sets ⁴⁷ ³¹. This feature is aimed at privacy protection and reducing user hesitation.

Current Implementation: The actual software **does not fully implement true anonymity** in its current form. The repository's data models and user flows resemble a traditional account-based system, especially for practitioners. The Unity client uses an **authentication system with username/password** – by default, it even hardcodes a demo login (“therapist” / “password”) to connect to the local API server ⁴⁸. The API endpoints include an `/auth/token` route for obtaining JWT tokens after login ⁴⁹ ⁵⁰, indicating that named user accounts are expected. There is no mention of an anonymous or guest login mode for clients in the API docs or code. In fact, the *Facilitator Guide* instructs users to “enter your username and password” and possibly complete MFA to access the system ⁵¹, which is contrary to an anonymous access paradigm.

Furthermore, the data structures suggest collection of personal data. For instance, the `ClientData` model includes fields for first name, last name, date of birth, and even diagnoses and notes ⁵² ⁵³. This

implies the system is designed for practitioners to input client profiles – possibly real names and details – rather than keeping clients unidentified. Nothing in the code explicitly strips or hashes personal info for anonymization. If anything, the presence of these fields **underscores a pivot away from strict anonymity**.

It's possible that “anonymous login” was intended in the sense that **clients might not need their own accounts** – a client could be registered in the system by a code or alias rather than an email, and their identity would only be known to the practitioner. The facilitator could log in and then initiate a session for a client without the client providing credentials. In practice, though, the codebase does not demonstrate a special handling of anonymity; it treats client records in a standard way (including PII fields). There is also a mention of compliance in security docs (GDPR, etc.) and “data anonymization options” ⁵⁴, but these appear in a checklist with no concrete implementation details visible.

Conclusion: The **privacy-focused features are largely not implemented** in alignment with the patent's description. The current system uses identifiable accounts and stores personal client info, which conflicts with the promise that “No personally identifiable information will be required” ⁴⁶. At best, practitioners could choose to enter pseudonymous data for a client, but that's a manual workaround, not a system feature. This is an **area of inconsistency**: the patent highlights anonymity as a selling point, yet the software at present operates with conventional user identification and data storage. Implementing true anonymized workflows would require additional development (e.g., guest access tokens, automatic de-identification of transcripts), none of which is evident in the repo.

Practitioner vs. Client Role Support

Patent Description: The platform is meant to serve both **practitioners (therapists, counselors, etc.)** and **clients (participants)** with tailored interfaces ⁵⁵. Practitioners should have an administrative interface to manage scenarios, monitor progress, and view analytics ⁵⁵. Clients (participants) get an interactive interface to engage with scenarios and the AI queries ². The system should thus differentiate user roles and offer appropriate features to each. Essentially, it standardizes practitioner-client interactions while still addressing their different needs ⁵⁶.

Current Implementation: The repository's design does account for **distinct user roles**, but the client-facing side is relatively underdeveloped compared to the practitioner side. On the practitioner side, we see substantial support: The **Facilitator Dashboard** (from documentation) includes a client list, session management panel, and access to reports/insights ¹¹. Data models like `ClientData` and `SessionData` link each session to a facilitator ID and client ID ⁵⁷, indicating the system tracks who the therapist is and who the client is for each session. There are also references to *facilitator guides* and *technical references* that describe how a facilitator uses the system to run sessions and interpret results. All this shows that the **practitioner role is explicitly supported**: they log in, choose or register clients, initiate scenarios or AI sessions, and review the outcomes.

For clients, the picture is a bit sparser. The patent envisioned a separate participant interface, but the current implementation seems to prefer a **facilitator-mediated model**. In practical terms, the therapist might run the app on a tablet or screen and the client participates by talking to the AI persona (with the therapist overseeing). The documentation includes an *End-User Session Participation Guide* for clients ⁵⁸ ¹², which gives tips on interacting with the AI and notes that the human facilitator is “overseeing” the session. This implies that clients might not be directly operating the software on their own (especially if anonymity is desired). Instead, the facilitator could be in control, or the client uses a simplified interface or

link to join a session that the facilitator set up ⁵⁹. The mention of joining via a provided link and testing one's microphone ⁶⁰ ⁵⁹ suggests there might be (or will be) a client-side app or web interface for remote sessions, though the repository we have is primarily the Unity application and AI backend.

Within the Unity project, we did not find separate scenes or UIs exclusively for a “client login” or a strict wall between what a client sees versus a facilitator. Instead, likely the facilitator logs in and drives the session. The *smart-steps-web-app* prototype, however, had the notion of an “Admin-only section” on the scenario discussion screen ⁶, which hints that certain information (like metrics or correct answers) might be visible only to the practitioner. This idea may carry into the Unity design as well (though not yet implemented) – for example, a facilitator might see the running analysis or suggestions from the AI, while the client just sees the conversation. The *SessionData* also contains fields like `SessionNotes` and `EffectivenessRating` which the facilitator would fill, not the client ⁶¹ ²⁶.

Conclusion: The **software architecture does recognize two user roles** – practitioner and client – primarily through data structures and documentation. The practitioner’s capabilities (user management, session control, viewing analytics) are being built out and documented. The client’s experience is described in guides but relies on the facilitator to initiate. At this stage, a **fully independent client interface is not evident** in the code (no separate client login or app with limited permissions). Clients likely engage through a session created by the practitioner, whether in person or via a meeting link. In sum, role separation exists conceptually and partially in implementation (especially for practitioners), but a robust, standalone client UI is **still in development**. This is an area to watch as the project grows, to ensure the client-facing side catches up with the practitioner tools.

Extensibility to VR, Voice, and Third-Party Integration

Patent Description: The platform is intended to be **extensible** to leverage emerging tech. Examples given include expanding scenarios into **virtual reality (VR)** environments for immersive training, adding **voice interaction** (speech input/output), integrating with **third-party systems**, and providing real-time feedback modules ². The underlying software should be flexible enough to incorporate these enhancements as needed, preserving the core process while extending its reach.

Current Implementation: The choice of **Unity** for the client application strongly positions the project for VR integration. Unity’s engine is well-suited for building VR experiences, and the mention of a Unity project (`ss_unity`) suggests the team has VR in mind. However, **no explicit VR content or SDK usage** is visible in the repository. We did not find VR-specific code (e.g., no references to XR plugins or VR headset inputs). At this point, the Unity app appears focused on 2D screens (UI canvases for dashboard and chat). The groundwork (Unity + cross-platform C# code) means adding VR scenarios is feasible later, but as of now, VR support is **not yet implemented**.

For **voice interaction**, there are hints that it’s considered but not done. The end-user guide advises checking your microphone and speakers if using voice features ⁶², implying the system may allow spoken conversation with the AI. The current code doesn’t include a speech recognition component or text-to-speech, so if voice was attempted, it’s not in this repo or not merged yet. Possibly the team plans to integrate an API (like Azure Cognitive Services or Google STT) to handle voice input in the future. The documentation’s inclusion of voice preparation tips suggests it’s on the roadmap, but **no concrete voice integration is present** in the code now.

Regarding **third-party integrations**, this is broad. We see a **“sync” API router** ⁴³ and a `SyncClient` in Unity ⁶³, indicating the application can synchronize data (perhaps between local device and server, or with an external database). There’s also a `monitoring` module and mention of compliance (GDPR, HIPAA) in security planning ⁵⁴, which could relate to integrating with health record systems or institutional databases. However, specific integrations (e.g., sending results to an EHR, or connecting to a learning management system) are not explicitly coded yet. The architecture is **modular and uses modern web APIs** (FastAPI backend), which makes future integration possible via REST endpoints or data export. But at the moment, third-party integration seems **mostly theoretical** or limited to generic data sync.

Conclusion: Extensibility to new interfaces and systems is **largely untested at this stage**. The design decisions (Unity front-end, structured API backend) are wise for future expansion – VR can be added on Unity, voice can be piped into the chat, and external integrations can call the API. Yet, in the current repo we find **no VR scenes, no voice engine, and no specific third-party connectors** implemented. These remain potential enhancements. The patent’s vision for VR/voice is forward-looking, and the repository is catching up slowly: it has laid an extensible foundation but hasn’t built these extensions yet (aside from preparing documentation and configuration hooks). This is understandable for a project in active development, but it means these patent features are **not realized yet** in the working product.

Patent vs. Implementation Comparison Table

Below is a side-by-side summary of the key features claimed in the patent and how they stack up against the current repository status:

Patent Feature / Claim	Implementation Status in Repository
Scenario-Based Interaction – Structured scenarios with decision points for client reflection ³ , potentially in VR ² .	Partially implemented. Conceptual scenario flows exist (e.g. web prototype with scenario selection ⁴), but no full-fledged scenario module in the Unity app yet. VR support is not implemented.
Query Phase (Post-Scenario Questions) – Tailored follow-up questions delivered by practitioner or automatically by system ⁸ .	Partially implemented. The AI persona and chat system enable Q&A dialogue (AI personas ask frequent reflective questions ⁹). Practitioners can manually ask questions. However, no dedicated logic ties specific scenario completions to a scripted set of evaluative questions.
Evaluation & Psychometric Analysis – Capture responses and measure growth via Rasch model and other metrics for SEL attributes ¹⁵ ¹³ . Two-level evaluation (client & practitioner reports) ¹⁷ .	Basic implementation only. Session data and analysis pipeline exist (AI-generated summaries & insights, plus placeholder metrics ⁶⁴). But no Rasch or advanced psychometrics are actually coded – metrics are simplistic and qualitative. Client vs. practitioner feedback is not distinctly handled in software.

Patent Feature / Claim	Implementation Status in Repository
AI-Guided Dialogue (Persona Module) – LLM assists in guiding scenarios, providing adaptive feedback, and enriching dialogue ³⁰ . Potential for a custom-trained LLM on anonymized data ³¹ .	Fully implemented (core functionality). The AI persona module is robust: multiple therapist personas with detailed profiles (e.g. CBT, behavioral, etc.) are integrated ³² ⁶⁵ . An LLM (OpenAI/Anthropic) generates persona responses and session insights in real-time. The design is modular for future custom models. This aligns closely with the patent’s vision of AI-assisted sessions.
Anonymous Login & Data Anonymization – No PII required for user access; store data without identifying info ⁴⁶ ³¹ .	Not implemented. The system currently uses standard user accounts (username/password) ⁵¹ and stores personal client information (names, DOB, etc. in <code>ClientData</code> ⁵²). There’s no special anonymized or guest login mode evident. Data is not automatically anonymized in the database.
Distinct Practitioner & Client Interfaces – Practitioner dashboard for managing assessments and viewing results; client interface for scenario participation ⁵⁵ .	Partially implemented. The practitioner side is well-supported (login, client management, session controls, analytics dashboard in docs ¹¹). The client experience is described in guides, but there is no separate client app – sessions are facilitator-led. The role separation exists in data (facilitator vs. client IDs ⁵⁷) and docs, but a standalone client UI is still in development.
Extensibility (VR, Voice, Integrations) – Ability to expand into VR environments, enable voice interactions, integrate with third-party systems for feedback, etc. ² .	Foundation only (not realized yet). Unity front-end provides a path to VR but no VR content is present. Voice features are alluded to in documentation ⁶² but not implemented in code. No specific third-party integrations are hooked up, though an API architecture is in place to allow future connections.

Table Source References: Patent claims are summarized from the provisional application ¹³ ² and related sections. Implementation status is derived from analysis of the SmartSteps_PROD repository – e.g., scenario UI in the web prototype ⁴ , persona JSON definitions ³⁴ ⁶⁶ , analysis code stubs ⁶⁴ , authentication flow ⁶⁷ , and documentation for user roles and features ¹¹ ⁶² .

Highlights and Gaps

In summary, the **Smart Steps platform’s development progress is uneven across features**. The **AI persona module is a highlight**, closely mirroring the patent’s ambitious use of LLMs to facilitate therapy sessions and extract insights. The **core application framework (Unity + FastAPI)** is in place, which bodes well for achieving cross-platform deployment and future VR/voice capabilities. The system already handles multi-persona chats, session data management, and basic reporting, indicating a solid architecture.

On the other hand, several **patent-promised features are lagging** in implementation. The **scenario-based content** – the very starting point of the three-phase process – is not fully functional, which is a critical gap. Without the scenario phase, the subsequent query and evaluation phases rely purely on open-ended conversation, which may limit the structured outcomes the patent aimed for. The **psychometric evaluation**

rigor is also missing; current analytics do not yet provide the kind of validated measures (like Rasch-based scores) that would give practitioners objective progress metrics. Additionally, the **privacy model** in the live code contradicts the patent's emphasis on anonymity, representing a potential compliance risk if not addressed (especially since sensitive psychological data is involved).

Technical Risks & Considerations:

- **Fidelity to Patent vs. Reality:** There is a risk that the software may fall short of patent claims in the near term. Features like anonymization and advanced metrics will require significant additional work or research integration. This discrepancy could lead to stakeholder disappointment if expectations aren't managed, or even intellectual property concerns if patent examiners seek working examples of claims.
- **Complexity of Scenario Integration:** Implementing dynamic branching scenarios (especially in VR) is non-trivial. It will demand narrative design, state management in Unity, and possibly AI-driven scenario adaptation. The team should ensure the robust AI persona doesn't overshadow the need for **structured scenario content** that maps to measurable outcomes.
- **Data Handling and Privacy:** The current design collects personal data and likely will accumulate sensitive conversation logs. **Lack of anonymization** and unclear data retention policies pose ethical and legal risks (e.g., GDPR, HIPAA compliance). This needs alignment with the patent's promises – perhaps by enabling alias-based client profiles or by stripping identifiers from analytic exports. The security review steps noted in Phase 8 plans ⁵⁴ must be executed in practice, not just on paper.
- **Reliance on External AI Services:** The heavy use of OpenAI/Anthropic APIs means the platform's core functionality (AI dialogue and insight generation) depends on external services, which involve costs, latency, and data transfer outside the system. Over time, training a **proprietary model** (as the patent suggests) could mitigate this, but that is a large undertaking. In the interim, careful caching, provider switching (already supported), and possibly on-premise model options should be considered to reduce dependency risk.
- **Extensibility Challenges:** While the architecture is extensible, each extension (VR, voice, integrations) will bring challenges. VR will require UX redesign and possibly **motion or environment design expertise**. Voice integration will require accurate speech-to-text and perhaps text-to-speech – ensuring the AI's responses can be vocalized in a human-like manner. Third-party integration must be done securely and in a generalized way (the planned REST API helps here). These are not insurmountable but will need dedicated effort and possibly partnership with specialists.

Despite these challenges, the **project's strength lies in its modular design and ambitious vision**. The current repository shows a lot of thoughtful engineering (for example, the persona system's depth and the comprehensive planning documents). The areas that are incomplete are well-known in the roadmap (scenarios, metrics, etc.), so there's awareness of what remains. If development continues at pace, we can expect the gap between the patent's described system and the actual product to narrow over time. Right now, however, there is a clear distinction: **the implemented platform is an AI-driven conversational therapy assistant with basic session tracking**, whereas the patent envisions a **full-fledged assessment platform with scenario simulations and quantitative psychometrics**. Bridging that gap will be the focus of further development.

1 2 3 7 8 13 14 15 16 17 18 19 29 30 31 46 47 55 56 **Provisional Application.pdf**

<file:///file-VTkR6mXn88u7MwWDRqDXdT>

4 5 6 **smart-steps-web-app-final-version.html**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/smart-steps-web-app-final-version.html

9 34 35 36 **professional_therapist.json**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/resources/ai_module/config/personas/professional_therapist.json

10 65 66 **behavioral_analyst.json**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/smart_steps_ai/config/personas/behavioral_analyst.json

11 32 33 51 **FacilitatorGuide.md**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/Docs/Deployment/FacilitatorGuide.md

12 58 59 60 62 **SessionParticipationGuide.md**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/Docs/EndUser/SessionParticipationGuide.md

20 21 22 23 24 25 27 28 44 45 64 **analyzer.py**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/smart_steps_ai/src/smart_steps_ai/analysis/analyzer.py

26 57 61 **SessionData.cs**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/Assets/Scripts/Models/SessionData.cs

37 39 **STEP_PLAN.md**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/smart_steps_ai/STEP_PLAN.md

38 40 **STEP_PLAN.md**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/STEP_PLAN.md

41 49 50 67 **SmartStepsApiClient.cs**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/Assets/Scripts/SmartStepsAI/API/SmartStepsApiClient.cs

42 **smart_steps_ai/src/smart_steps_ai/api/routers/conversations.py**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/smart_steps_ai/src/smart_steps_ai/api/routers/conversations.py

43 **app.py**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/smart_steps_ai/src/smart_steps_ai/api/app.py

48 **SmartStepsAIManager.cs**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/Assets/Scripts/SmartStepsAI/SmartStepsAIManager.cs

52 53 **ClientData.cs**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/Assets/Scripts/Models/ClientData.cs

54 **PHASE8_PLAN.md**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/PHASE8_PLAN.md

63 **ss_unity/Assets/Scripts/SmartStepsAI/API/SyncClient.cs**

https://github.com/Loflou-Inc/SmartSteps_PROD/blob/60583c6d6dba6a479a289476f4744f9af648e49b/ss_unity/Assets/Scripts/SmartStepsAI/API/SyncClient.cs