**Product Requirements Document: Data Analyst Client Simulator (DACS)**

**1. Introduction**

* **1.1 Purpose** This document outlines the requirements for the Data Analyst Client Simulator (DACS), a web application designed to provide a realistic and challenging environment for aspiring data analysts to practice their skills, build their portfolios, and enhance their communication abilities.
* **1.2 Scope** This PRD covers the requirements for the Minimum Viable Product (MVP) of DACS, focusing on core functionalities necessary to demonstrate the value proposition and gather initial user feedback.
* **1.3 Definitions, Acronyms, and Abbreviations**
  + **AI:** Artificial Intelligence
  + **DACS:** Data Analyst Client Simulator
  + **LLM:** Large Language Model
  + **MVP:** Minimum Viable Product
  + **PRD:** Product Requirements Document
  + **UI:** User Interface
* **1.4 Overview** DACS will simulate client interactions for data analysts using an AI-powered client persona. The AI will generate scenarios, provide data, answer questions, and introduce unexpected challenges, mimicking real-world client engagements.

**2. Project Overview**

* **2.1 Project Name:** AI Client Simulator for Data Analysts (DACS)
* **2.2 Project Goal:** To develop an AI-powered web application that simulates realistic client interactions for data analysts, providing a dynamic and challenging environment to practice their skills, build their portfolios, and improve their communication abilities.
* **2.3 Target User:** Primarily aspiring and entry-level data analysts, students in data science programs, and potentially data analysis teams for training purposes.
* **2.4 Value Proposition:**
  + **Realistic Client Simulation:** Provides an immersive experience that mimics the challenges and unpredictability of working with real-world clients.
  + **Portfolio Building:** Enables users to create diverse and compelling data analysis projects by responding to realistic client requests and scenarios.
  + **Enhanced Communication:** Develops data storytelling and communication skills through interaction with a dynamic AI client.
  + **Accelerated Learning:** Offers a safe space to experiment, make mistakes, receive feedback, and learn from them, accelerating the development of practical data analysis expertise.

**3. Features (MVP)**

* **3.1 AI Client Persona Engine**
  + **3.1.1 Description:** Core AI agent that embodies a specific client personality (e.g., "pragmatic"). The AI drives the scenario, generates project requirements, provides data, and interacts with the user.
  + **3.1.2 Requirements:**
    - The AI must be able to maintain a consistent personality throughout the interaction.
    - The AI should be able to generate realistic project requirements based on its persona.
    - The AI should be able to engage in a natural language conversation with the user.
    - Initial development will focus on a single, well-defined "pragmatic" persona.
* **3.2 Synthetic Data Generation**
  + **3.2.1 Description:** Creation of realistic datasets (initially CSV format) tailored to the generated scenarios.
  + **3.2.2 Requirements:**
    - The system must be able to generate datasets in CSV format.
    - The data should be relevant to the generated scenario.
    - Users should be able to specify basic parameters like data size, data types, and potential anomalies.
* **3.3 Dynamic Scenario Generation**
  + **3.3.1 Description:** The AI client generates project descriptions, objectives, and initial requirements based on its persona and the chosen dataset. The scenario evolves based on user interaction and choices.
  + **3.3.2 Requirements:**
    - Scenarios should be logically coherent and engaging.
    - The AI should adapt the scenario based on user input and decisions.
    - Initial scenarios should be relatively simple, focusing on common data analysis tasks.
* **3.4 Web Application Interface**
  + **3.4.1 Description:** A user-friendly interface for interacting with the AI client.
  + **3.4.2 Requirements:**
    - **Chat-like interaction:** A chat window for communication with the AI.
    - **Data request:** A mechanism for users to request specific data or clarifications.
    - **Deliverable submission:** A way for users to submit their completed work (e.g., uploading a file or pasting in code/results).
    - **Project display:** Clear presentation of project descriptions, objectives, and data.
    - **User authentication:** Basic user login and registration (optional for MVP, but recommended for future development).
    - **Responsive design:** The interface should be usable on different screen sizes (desktop, tablet).
* **3.5 Simulation of "Unpredictable" Elements**
  + **3.5.1 Description:** The AI client introduces unexpected events or changes requirements during the project (e.g., shifting priorities, new data discoveries, feedback requiring revisions).
  + **3.5.2 Requirements:**
    - The AI should be able to introduce at least one unexpected event or change per project.
    - These changes should be logical within the context of the scenario.
    - The frequency and complexity of these events can be adjusted in future iterations.

**4. Technology**

* **4.1 LLMs:** OpenAI API (GPT-3.5/GPT-4) or similar for the core AI client persona engine, interaction, and scenario generation.
* **4.2 Synthetic Data Generation Tools:** Synth, SDV, or similar.
* **4.3 Programming Language:** Python.
* **4.4 Web Application Framework:** Flask or Django.
* **4.5 Frontend Development:** HTML, CSS, JavaScript.
* **4.6 Framework (Potentially):** LangChain to structure and manage the LLM interactions.

**5. Project Timeline**

* **5.1 Phase 1: MVP Development (1 month)**
  + Focus on core features (as described in section 3).
  + Create a functional prototype with a single client persona.
  + Develop a basic, but usable, web interface.
* **5.2 Phase 2: Beta Testing (Timeline TBD)**
  + Gather feedback from a small group of target users.
  + Iterate on the design and functionality based on feedback.
* **5.3 Phase 3: Expansion and Refinement (Timeline TBD)**
  + Add more client personas, data types, and scenario complexity.
  + Refine the web application based on user feedback.
  + Implement features from the "Future Considerations" section.

**6. Success Metrics**

* **6.1 MVP Completion:** Successful development of a functional prototype with a basic web interface within the one-month timeframe.
* **6.2 User Engagement (Beta Testing):**
  + Number of users interacting with the AI client.
  + Frequency of use.
  + Completion of projects/scenarios.
* **6.3 User Feedback (Beta Testing):**
  + Qualitative feedback on the realism of the AI client.
  + The usefulness of the tool for skill development.
  + Identification of areas for improvement in functionality and UI/UX.
* **6.4 Personal Satisfaction:** (For initial phase) My own assessment of the tool's value for portfolio development.

**7. Future Considerations (Beyond MVP)**

* **7.1 Wider Range of Client Personas:** Develop a library of diverse client archetypes with unique personalities and interaction styles (e.g., "analytical," "skeptical," "hands-off").
* **7.2 Advanced Data Types:** Support for various data formats (e.g., JSON, SQL databases) and larger datasets.
* **7.3 Multiple Deliverables:** The AI client can request different types of deliverables (e.g., reports, dashboards, presentations).
* **7.4 Enhanced Web Application UI:** A more sophisticated and visually appealing interface with features like:
  + Data visualization tools.
  + Progress tracking.
  + User profiles.
* **7.5 Integration with other Data Analysis Tools:** Potential for the AI client to interact with or provide data to other data analysis tools (e.g., Jupyter Notebook, Tableau).
* **7.6 Monetization Strategy:** If the project moves beyond personal use, explore options like a subscription model or usage-based pricing.
* **7.7 AI-Generated Feedback:** Implement a feature where the AI provides feedback on submitted deliverables, focusing on areas like data analysis methodology, communication clarity, and presentation quality.

**8. Open Issues**

* **8.1 Specific LLM Choice:** Final decision on whether to use OpenAI API or another LLM provider.
* **8.2 Synthetic Data Tool Selection:** Evaluate and choose the most suitable synthetic data generation tool.
* **8.3 Detailed Design of "Unpredictable" Events:** Create a more comprehensive list of potential unexpected events and changes for the AI client to use.
* **8.4 Beta Testing Plan:** Define the specific criteria for selecting beta testers and the process for collecting and analyzing their feedback.

This PRD provides a solid foundation for the development of the DACS project. It will be a living document, evolving as the project progresses and new information becomes available. Remember that as both developer and project manager, I will continuously iterate on this document and the product itself, using feedback and insights gained during development.