**REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

A **Customer Journey Map** is a visual representation of the user’s experience interacting with HEALTH-AI — from discovering the chatbot to gaining actionable insights. The goal is to highlight user goals, pain points, and AI interactions at each stage.

**📍 User Persona:**

**Name:** Raj  
**Age:** 25  
**Occupation:** College student  
**Location:** Urban India  
**Pain Point:** Can't always consult a doctor; relies on Google searches.

**📈 Stages of the Journey:**

| **Stage** | **User Action** | **System Response** | **Pain Point Solved** |
| --- | --- | --- | --- |
| Discover | Searches for health AI tool | Finds HEALTH-AI via web/GitHub | Information accessibility |
|  | Onboard | Opens chat interface, enters symptoms | Accepts input in natural language | Usability |
| Engage | Asks follow-up questions | Context-aware generative responses | Personalization |
| Interpret | Sees chart of possible diseases | Visual chart generated | Clarity and trust |
| Reflect | Asks for remedies | Gets home remedy + treatment info | Affordable solutions |
| Reuse | Uses again for different query | Memory-free fresh session | Consistency and simplicity |

**3.2 Solution Requirements**

This section breaks down both **functional** and **non-functional** requirements of the system.

**✅ Functional Requirements:**

* FR1: Accept symptoms or health queries via user input.
* FR2: Process queries through IBM Granite model using Hugging Face API.
* FR3: Generate text output with disease predictions and remedies.
* FR4: Render visual chart output (bar chart showing disease probabilities).
* FR5: Support multi-turn conversations (contextual follow-up questions).
* FR6: Export visualizations if needed.

**✅ Non-Functional Requirements:**

* NFR1: Response time should be under 5 seconds.
* NFR2: System should handle 95%+ common symptom queries reliably.
* NFR3: Interface should be user-friendly and intuitive.
* NFR4: Model communication must be secured (via API token).
* NFR5: Output should avoid overconfident medical advice — include disclaimers.

**⚠️ Constraints:**

* Model accuracy is limited to training data.
* No real-time user data storage (for privacy).
* Internet dependency (due to Hugging Face API).

**3.3 Data Flow Diagram (DFD)**

A **Data Flow Diagram** describes the flow of data between the user, backend services, model inference, and the final output.

**🧭 Level 1 DFD Description:**

1. **User Input:** A user types a symptom or question.
2. **Backend Processing:** Python code parses and validates the query.
3. **Model Invocation:** IBM Granite model (via Hugging Face API) processes input.
4. **Output Parser:** Filters and formats model's output into user-friendly form.
5. **Chart Generator:** Converts disease prediction text into bar/pie chart.
6. **Chat Interface:** Displays both the text and chart to the user.

**📘 System Entities & Processes:**

| **Entity** | **Role** |
| --- | --- |
| User | Sends input and receives responses |
| Python Script | Handles model request and logic |
| Hugging Face API | Provides access to IBM Granite model |
| Granite Model | Generates text outputs |
| Matplotlib | Used to generate bar chart of diagnosis |
| Output Interface | Displays responses in Google Colab/CLI |

**🔄 Summary of Requirement Analysis**

* **Customer Journey** shows how HEALTH-AI guides users through health discovery to insight.
* **Functional/Non-functional Requirements** ensure the system is usable, reliable, and efficient.
* **Data Flow** shows how each component processes and transforms the input to deliver meaningful outputs.