# Workflow to Design the System

Good — you're speaking at a **systems architecture** level now, exactly where you need to be. Let’s go step-by-step and **break this monster into manageable pieces**.

Here’s **how you can structure the process** from where you are today:

# 🧩 1. Define the Workflow: "From Meeting the User to Diagnosis"

The first thing is to map the **phases** a user goes through.

**User Diagnostic Journey:**

**1** **User Enters**:

\* "What's up?" → capture initial *problem language* in their own words (free text or pick from options)

**2** **Problem Identification**:

\* Classify their issue into a **small number of core categories** (career, burnout, purpose, relationship, etc.)

**3** **Detail Gathering**:

\* Ask smart **follow-up questions** depending on which category they fall into

**4** **Context Mapping**:

\* Pull in their **life circumstances** (student/adult/retiree, working/unemployed, money pressure, etc.)

**5** **Trait Mapping**:

\* (If available) pull in or estimate **nature and trait** influence

**6** **Preliminary Diagnosis**:

\* Match user to **diagnostic profiles** (e.g., "Mild Burnout," "Misaligned Career," "Existential Drift")

**7** **Direction Setting**:

\* Suggest initial **pathways** toward solutions (e.g., "Explore Aligned Work," "Reignite Meaning," "Build New Identity")

⠀

# 🧩 2. Content Elements You Need to Build

Here’s **what needs to be created** in content form:

|  |  |  |
| --- | --- | --- |
| **Element** | **What it is** | **Purpose** |
| **Problem Categories** | 20–30 clear "buckets" users fall into | To quickly orient diagnosis |
| **Sample User Language** | Examples of how users express each problem | To train AI recognition |
| **Follow-Up Questions** | Specific questions tied to each category | To refine and deepen diagnosis |
| **Life Circumstances Tables** | Structured factors: student, career phase, finances, etc. | To add diagnostic context |
| **Trait Influence Mappings** | How different natures/traits affect problems | To make diagnosis more personalized |
| **Diagnostic Profiles** | Defined "problem-solution" templates | To structure outcomes for users |
| **Direction Paths** | Clear first steps depending on profile | To guide user toward solution |

# 🧩 3. How to Build It Practically

(**This is the step-by-step work order you give yourself and Sean**):

**1** **List Top 25–30 Core Problems**

\* Career confusion

\* Burnout

\* Lack of meaning

\* Want to start something new

\* Feel stuck after graduation

\* Etc. (We can brainstorm these now if you want)

**2** **For Each Problem:** a. Write the **User Language** ("I feel burned out" vs. "I'm tired of everything" vs. "Work is draining me.") b. Write 5–10 **Follow-Up Questions** (e.g., "How long have you been feeling this way?", "Is it your role or the whole field?") c. Note **Key Circumstances** that affect it (e.g., "is employed", "under financial pressure") d. Predict **Nature/Personality Factors** that might correlate (e.g., high Providing Nature overworked) e. Create a **Diagnostic Label** (e.g., "Moderate Burnout due to Role Misalignment")

**3** **Make Tables:**

\* Problem Categories Table

\* Language Examples Table

\* Follow-Up Question Table (linked to problem)

\* Circumstance Factors Table

\* Trait Factors Table

\* Diagnostic Profiles Table

\* Direction Paths Table

**4** **Deliver to Sean** as:

\* Structured CSVs / spreadsheets ready to import

\* Plus a **logic flowchart** showing how initial answers → classification → follow-up → diagnosis → pathway

⠀

# 🧠 BIG PICTURE:

You're creating a **multi-layered system** that maps:

**Problem → User Language → Questions → Circumstances → Traits → Diagnosis → Solution Path**

# 📋 Immediate Next Step Options:

**1.** Brainstorm the **Top 25–30 problems** (and rough groupings) **2.** Sketch sample **User Language** for 5–10 problems **3.** Draft sample **Follow-Up Questions** for 5–10 problems **4.** Start setting up the **tables** so we have a structured plan for Sean

# 🚀 In short:

* **This is doable.**
* **You already have all the raw material.**
* **We just need to formalize it into a "Xavigate Diagnostic Engine" structure.**