Here is the updated plan incorporating your changes, followed by my suggestion on the "one vs. two apps" question.

Updated Dashboard Plan V2 🚀

This plan now includes **role-based views**, meaning the interface changes depending on who is logged in.

Panel / Feature	View for Ambulance Crew	View for Hospital Staff
Panel 1: Action Panel (Left)	**Patient Registration Form:** • A simple form to quickly register a new patient. • Fields: Basic identifiers (Name, Age, etc.). • Action: Generates a unique Transfer ID for the session.	**Patient Registration Form:** • The same form, allowing hospital staff to register patients for inter-hospital transfers.
Panel 2: Map (Center)	Displays the ambulance's **live location** and the **dynamic, fastest route** to the assigned hospital.	Displays the live location and route of any **selected incoming ambulance**.
Panel 3: Information Panel (Right)	**Live Vitals & Predictions:** • Shows the patient's real-time vitals graph. • Displays the immediate AI predictions (Severity).	**Full Emergency Report:** • Shows the patient's registration details. • Displays the complete AI prediction report, just like in your simulation video: - Predicted Patient Severity - Suggested ICU Destination - Predicted Resource Needs

---

One App or Two? My Suggestion

You asked for a suggestion on whether to build one unified app or two separate apps (one for ambulance, one for hospital).

My recommendation is to build ONE UNIFIED APPLICATION with role-based views.

Here’s why this is the better approach for your project:

- **Efficiency:** You manage a single codebase, which is far easier and faster than building and maintaining two separate applications. You build core features like the map and API connections only once.
- **Consistency:** It ensures a consistent data flow and user experience. When you update a feature, it's updated for everyone.
- **Simplicity:** The modern way to handle this is with **Role-Based Access Control (RBAC)**. When a user logs in, the app checks their role (e.g., "Paramedic" or "Doctor") and simply shows or hides the panels and features they're allowed to see. It’s one app that wears different "hats."

Building two apps would double your workload and create unnecessary complexity at this stage. We can design a single, smart application that adapts its interface to the user.

With this updated plan, we now have a clear blueprint. We are ready to start coding when you are.