**How can I reuse some knowledge and ideas from TM352 EMA in my project?**

First off, the tech stack. EMA is using React Native for the frontend, which is perfect since I’m doing the same for TM470. They're also using Expo for mobile optimisation, and that fits right into my workflow too. On the backend, EMA is using Node/Express and SQLite, which seems pretty solid and reliable for data persistence. I definitely need to keep the way they handle in-memory data in mind for demos, but for TM470, a more persistent solution is what I'm after.

Now, let’s talk interface. EMA has a straightforward flow from login to accessing main features, all powered by React Navigation. I think I need to simplify the navigation in TM470, but make it similar to that. A clean, easy-to-follow interface could really improve the user experience, which is what it’s all about.

In terms of structure, EMA does a nice job with its registration and permit booking process. But my maintenance application prototype for the TM470 will have many more functionalities. So I think I should centre it more around the way inspections and checklists are completed in different applications of a similar type. Maybe I could get hold of MaintainX or another type of digital form completing application.

However, the way the TM352 EMA is comprised of a user-driven flow is something I could reuse in my TM470 maintenance application prototype.

This means my application could have a similar user-driven flow, and I can draw inspiration from the TM352 EMA to streamline the processes in TM470. For instance, I could add a user-friendly dashboard where engineers can quickly pick their tasks, similar to how berthing permits are displayed in the Berthing Booking app.

When it comes to data models, EMA emphasises managing QR codes alongside permits, whereas TM470 is more about inspections and their outcomes. One thing that stands out is how EMA digitises permits. Maybe I can integrate some form of digital tagging for inspections in TM470 to assist traceability.

Looking at the core flows, the Berthing Permit Booking system's approach to registration and booking appointments has some clear steps, which might inspire me to simplify the inspection process in my TM470 app. I could present a list of actionable inspections right after logging in, reducing clicks and making the user interaction smoother.

Now, let’s touch on business rules. EMA's dynamic forms for permits could really translate well into TM470, since my checklists might vary based on the type of inspection. I could definitely consider implementing adaptable, template-driven checklists that meet specific needs, just like EMA does.

Persistence is another area where EMA’s in-memory structure works great for demos but isn’t what TM470 needs. I need to ensure SQLite allows offline data capture, so engineers can carry out inspections without an internet connection. That’s a must for fieldwork!

Switching gears to security, EMA takes a somewhat relaxed approach with light authentication. However, I’ll need to ramp up the security for TM470, incorporating hashed passwords and role-based access depending on whether someone is an engineer or a manager. I’ll take some best practices from EMA but will definitely need to elevate the security measures for the sensitive data we’ll be handling.

Reporting is another consideration. EMA focuses on permits and QR codes, but TM470 requires more robust reporting capabilities, like generating lists of inspections and their results. Examining how EMA structures its outputs could really help me figure out how to approach reporting in TM470, especially in creating easy-to-access visualisations and documentation for managers.

Finally, about assessment fit, TM352 emphasises a clear design-to-implementation process, which is super valuable for TM470. I’ll need to incorporate comprehensive testing and justification of my tech choices, so taking a structured approach from EMA could serve as a solid framework to ensure TM470 meets its goals effective and ethical.

So, all in all, there’s a ton I can learn from the EMA project that could really make TM470 shine—whether it’s from the tech stack, simplifying user flows, dynamic forms for inspections, or efficient reporting strategies. I just need to prioritise these insights and adapt them for the unique goals of TM470. That could really help in creating a solid user experience for reproducibility.