1. High-Level Technical Architecture diagram is attached.

The system is composed of 4 applications each with their own database. The 4 services are:

1. Vehicle & Part Order Management (VPOM)
2. Customer Relationship Management (CRM) – with functionalities of Lead Management, Sales and Service Management
3. Finance & Reporting
4. Marketing
5. For each layer, I shall choose the following technologies

Database: MongoDB

Microservices: NodeJS

Mobile Apps: React Native/ Ionic

Web App: ReactJS / Angular

EventBus: Kafka (interaction among the Microservices)

The reason for choosing each technology is as follows

MongoDB: Fast, efficient, open source, enduring, cross-platform, huge community, works with structured and unstructured data, easy syntax

NodeJS: Fast, efficient, open source, enduring, cross-platform, huge community, is JavaScript

ReactNative & ReactJS: Fast, efficient, open source, enduring, cross-platform, big community

Angular & Ionic: Fast, efficient, open source, enduring, cross-platform, big community

The choice for most of the technologies to be used has been made considering various factors including the company being a startup. Most of these technologies are free and easy to use, quick to work with, fast and efficient, proven and enduring.

Although we can use a variety of technologies since each service is an independent microservice, using correlated languages should help us develop in an easier manner requiring expertise in limited domains.

External factors such as availability of developers could also play a role.

1. In my opinion, Customer Relationship Management should be mobile enabled. A lot of the features involved in Lead management, Sales and Service management may have to be done while not on a desk but chasing people. This is also the most used module, in my opinion.

The other modules involve feature that may not be apt for small screen devices. For eg, a vehicle has hundreds of parts if not thousands. Although categorization can reduce the count per category, it would still be a little too many to choose from on a small screen. Moreover, inventory/ stock of parts is generally at the dealership, where you expect a computer system.

Finance & Reporting involves sensitive data. Exposing such information on a personal device may be a decision for the involved dealership to make. Nevertheless, a case for the module to be shown on mobile can be made. We may use configuration to turn the module on/off on mobile.

1. Mobile application should be built either in react native or ionic to run on both Android and iOS platforms. It will save time and money. One codebase for both platforms is also easier to maintain. Mobile apps can connect to the microservice just like web app via a gateway.
2. Technical skills as mentioned above with some cloud experience. Given that these technologies are so closely related (MEAN, MERN) knowledge on all skills is not required for all the developers. A mix and match approach will do.

Besides, people who start without worrying about the end, who bring clarity, who bring calm yet positive energy are who I would be looking for.

Considering it is a startup, it makes sense to keep capital expenses low. Hence, I propose to deploy the entire solution on cloud. Azure is my preferred choice. Cloud also allows independent scalability, possibility of using cloud functions (for services like Reporting) and so much more besides keeping capex low.