Case

This case is a fictional one and has no connection to Bain's current or past projects.

Delivery apps have boomed during this pandemic and we have a new client that don't want to be left behind in this trend and we plan to launch our own delivery company. For this, we have decided to start with a working prototype, put it to the test for a few months to see if it manages to gain traction and finally, if the test is successful, go all the way to production. The functional prototype, although, it will be basic, must allow customers to place orders and delivery people to do their work.

All the necessary software for this will be developed by us and you will be the Software Engineer in charge of leading the development team. Your team will initially be composed of:

• A Dev designer: Frontend design and development specialist.

• An Impact Lead: In this case, you can consider that it is someone specialized in the business and that will help you so that what you develop has a real impact.

• You, as Software Engineer and project leader.

• You will also have the support of a Data Scientist and a Machine Learning Engineer, They will not participate in the entire process, but you can ask them for help on specific issues.

Next, we would like you to answer a few questions to understand how you would approach the development process. Don't worry if you feel the questions are too open-ended or there isn't enough information to answer them. They are built precisely that way so that you can make all the assumptions you deem appropriate.

Questions

Answer these questions. Maximum 150 words per question.

1. Tell us what pieces of software you think are necessary to develop for the working prototype and how they are related. We call each application (web, mobile or desktop), each API, each batch process that can be deployed independently a piece of software. Support yourself with a diagram if you think necessary.

Web: React
Mobile: React Native
Backend: Node.js with RESTful APIs for authentication (JWT/OAuth), authorization (RBAC), batch processing, and frontend data delivery
Database: PostgreSQL for storing user data, orders, deliveries, etc.

Batch Processing: Bull and Redis for handling delivery updates, order assignments, and notifications

The frontend applications (web, mobile) will interact with the backend to fetch and display data like orders, statuses, and user info. They authenticate users with JWT/OAuth, which is validated by the backend. The backend handles incoming requests, communicates with the database, manages authentication, roles, and controls batch processes. Batch processes are responsible for updating order statuses, assigning deliveries, and sending notifications, triggered either on demand or on a schedule.

2. Tell us about the type of architecture you chose for question (1). Monolithic? Micro-services? Any intermediate? Other? Comment on what you based to make this decision.

I would choose the Modular Monolithic architecture. Given the small team and the goal of developing a functional prototype, this architecture offers several benefits:

- **Speed**: fast development architecture that can handle cooperative development, the team can focus on creating features quickly.
- **Simplicity**: minimal complexity to manage and deploy, the modular design allows the team to maintain a clean code.
- **Faster iterations**: iterates quickly based on client feedback, making it easier to attend new requirements.
- **Scalability later**: it can be scaled up to microservices post-production considering that the team will grow, and the application requires scaling.
- **Avoiding over-engineering**: microservices would be too complex for the initial prototype and small team. It is better for scalability, however the complexity of the architecture would lead the team off-track. The focus should be on delivering the intended application the best way possible with the tools and resources we got.

3. Describe the work methodology you would use for development. It can be some known methodology (Scrum, XP, RUP), an adaptation, or a mixture between several methodologies. Whatever your experience has shown you works. Tell us why you think this form is appropriate for our problem.

I would choose Scrum. It is a proven agile methodology that I have successfully used before and still use a lot. Even without a dedicated Scrum Master, the team lead can guide the team effectively.

By following the ceremonies - Planning, Daily, Review, Retrospective and Refinement – we can increase the productivity and minimize rework, also giving space for an organized inclusion of new requirements, features or use cases. This way we can refine a backlog in the start of the project, prioritize tasks and assign responsibilities for each sprint. This way we will be aligned with the Product Owner, and the client and tech team will have visibility of what's being done, what needs to be done and if the development is going as expected to deliver the application.

4. Describe the workflow you would use to collaborate using Git. As with (3), you can use something familiar or an adaptation.

Communication is key. The team must inform each other of what is being done, if something will impact others' development then it is our responsibility to communicate via internal group chat like Teams, Slack or other.

Collaboration would be via Git, since it is easy to use, and natively integrated with some IDEs, such as VSCode.

The workflow would follow the best practices of Git and a set of guidelines to ensure consistency. We would use Pull Requests to make sure the code is reviewed before merging into the main branches (dev, qa, uat or prod). Each development would be done in a separate branch, with small commits to avoid breaking the code and reduce conflicts, making rollbacks easier if needed. Finally, every development would need to have a valid test with evidence to make sure the requirements are met before merging.

5. Do you think it is necessary to add any extra member to the team during the development of the prototype? What would your role be? Do you think it would be necessary to add new members after the prototype phase? When and why?

I believe the team can deliver the prototype with the right tools and an aligned deadline. However, I would add a QA tester and a Scrum Master to the development phase, this would improve productivity and ensure proper roles distribution.

During the hypercare phase I would add one or two junior members, to assist end-users with their questions and organize feedback, helping in a fast product improvement.

In the post-production phase, considering the product is doing well and growing, a curatorship team would be beneficial. We could use the support of the Data Scientist and bring a BI analyst to provide metrics and monitoring dashboards. Additionally, a couple more developers to help delivering new use cases while maintaining the current features and if scaling is necessary, they would be great to do it properly.

Finally, a DevOps engineer to manage the infrastructure, guarantee service availability, and automate CI/CD pipelines.

6. What other considerations would you have to make the development process robust and efficient?

To make the development process robust and efficient I would follow these key considerations:

- **Best practices**: writing clean code, following the guidelines provided and providing documentation of the work being done.
- **CI/CD**: automating tests, deployments and reduce human errors.
- **Communication:** maintaining an open communication within the team to identify and resolve issues quickly, especially through scrum routines.
- **Stakeholder alignment:** regularly aligning expectations with stakeholders to reduce miscommunication, rework, and maximize value delivered.