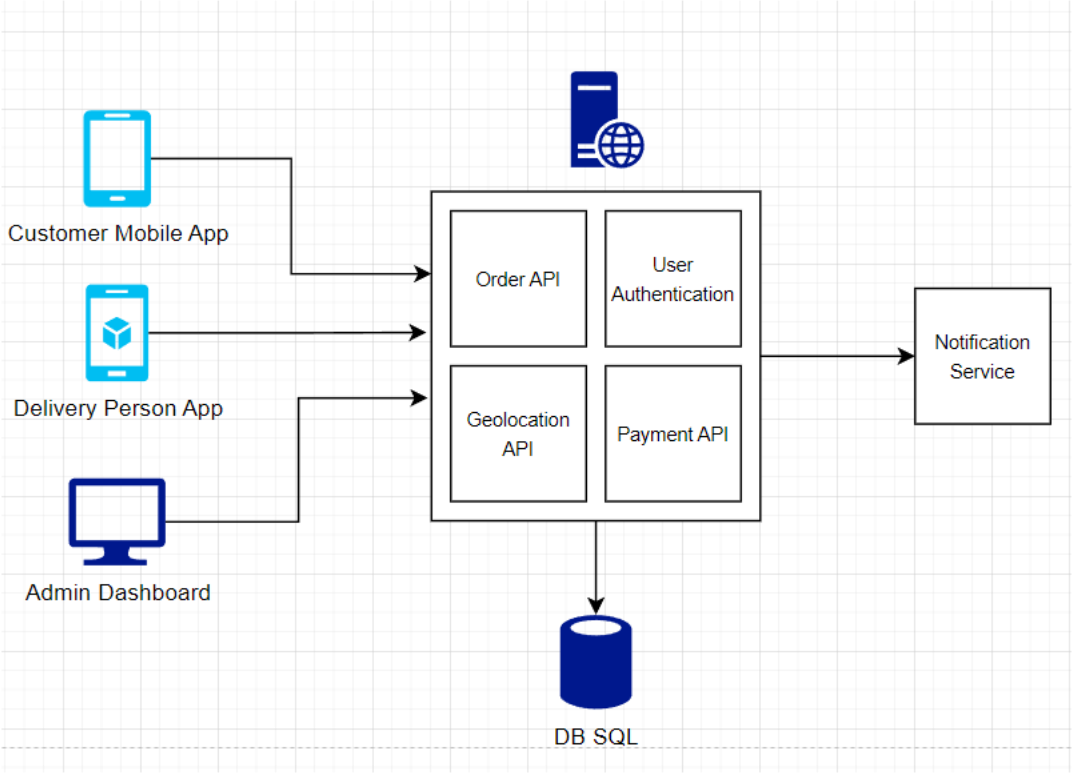
Delivery Application Challenge

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.

   Answer: We will need to develop a Customer mobile app, a Delivery Person mobile app, a Web Admin Panel, a Notification service, and a core service to connect them all. The Customer app will allow users to place orders, track their orders, and pay for them. The Delivery Person app will enable individuals to accept orders, update delivery status, track their location, and manage their work. The Web Admin Panel will allow the business to manage orders, delivery personnel, and customers. A monolithic core application will be responsible for the business logic, serving the APIs for the mobile apps and the web admin panel to manage orders. The Notification service will send notifications to customers and delivery people. This service will be separate from the start, based on the potential need to scale it independently and the possibility of using third-party services.



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.

   Answer: For this prototype stage, I've chosen a modular monolith architecture, as a microservices architecture might be overkill at this point and could introduce unnecessary complexity. A monolith allows for faster initial development, and with a small team, it is simpler to manage. It's also more cost-effective, which is essential for a prototype that's still proving its market viability. By designing the monolith in a modular fashion, we're preparing for a potential transition to microservices if the prototype is successful and the business scales.

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.
   Answer: I would use an adaptation of Agile methodology, a mixture of Scrum and Kanban. The Agile methodology is ideal for a project like this, where requirements are likely to change as we learn more about the market and user needs. Agile allows for flexibility, adaptability, and quick iterations, which are essential for a prototype. As we're testing the market, we need to be able to pivot quickly based on user feedback. Kanban is an excellent way to visualize the workflow and manage the team's capacity. It helps identify bottlenecks and keeps the team focused on delivering value. By combining these two methodologies, we can take advantage of Agile's flexibility and Kanban's visual management to ensure we're delivering valuable features quickly and efficiently.

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

   Answer: I would choose a trunk-based development workflow. This approach is simple to implement and allows for fast integration and continuous delivery. In this workflow, all developers work on the main branch, and feature branches are short-lived. Developers create a feature branch, make their changes, and then merge them back into the main branch as soon as possible. This ensures that changes are integrated quickly and conflicts are resolved early. It also enables continuous integration and deployment, as changes are merged into the main branch frequently. This workflow is well-suited for a prototype project where speed and agility are essential.

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?

   Answer: I believe it may be necessary to add a mobile developer to the team during the development of the prototype. As a Software Engineer, my role would involve leading the technical design and architecture decisions, collaborating with the business specialist to refine requirements, developing the core features, and coordinating with the team to ensure the project's success. I would also be responsible for code reviews and ensuring the quality of the codebase. After the prototype phase, depending on the project's success, we may need to add new members to the team to scale the business. This could include additional developers, a QA engineer to test the application, a DevOps engineer to manage deployment and infrastructure, a Data Engineer, and a Product Manager to coordinate a larger team and drive the product roadmap. These roles would be added as needed to support the business's growth.

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

   Answer: I would consider an automated testing strategy to ensure the quality of the codebase and catch bugs early, especially for critical business logic. This could include unit tests, integration tests, and end-to-end tests. I would also set up a CI/CD pipeline to automate the build, test, and deployment process. This would allow us to deliver features quickly and reliably. Additionally, I would establish code reviews as a standard practice to ensure code quality and share knowledge among the team. Lastly, I would set up monitoring and logging to track the application's performance and identify issues quickly.