Style Agnostic:

1. Some text garbling ('projecyngr', 'work product'), strange use of interfaces, inputs and output represented as components

2. Diagram cuts off at top and bottom. Describes the data architecture but not the system architecture

3. Empty box, disconnected printer. Does not account for save action, table should be connected with body, no mention of images. Data architecture

instead of system architecture again
4. Includes users like a use case, not a component diagram. Additional components that are not needed ("Doctor Management"). At least is a system architecture

5. Double component on the payment service

6. Incorrect connections between components. Data architecture, not system architecture

7. Again, more of a data architecture. Balance repeated across multiple components. Includes unmentioned external system.

8. Shows clear (if somewhat useless) layers between UI, application, and database. Maps the different types of data, not the functionality

9. Repeated / blank components. Represents data but not behavior

10. Seems to be all clumped under "restaurant service" with some data design underneath. Duplication of Client component

11. Missing many "domain knowledge" kinds of components, e.g. service to determine closest delivery center.

12. Data grouped under the aegis of the "Furniture System"

13. This is clearly a database design, not a software design (database was specifically requested in the requirements)

14. Duplicate Store Service, strange...other thing on the left. Some text issues.

15. This one isn't bad (apart from, say "oheck enrd"). Includes functionality as interface names

16. Diagram layers are pretty well defined, although not necessarily correct. Duplicate consumer division

17. Everything nested under contract. Seems focused on data model design, not system design

18.The prompt specifically calls out class design, so it made an OO class structure, not a real system architecture. Might need to change the prompt.

19. All reference the central MyDoctor Application. Very high level

20. Unnecessary cycles. Web user as an interface?

Layered:

1. Direction of calls between layers are not clearly defined, hard to say which way they go
2. Solid start on this one. Doesn't really get into the "setups", but overall a good high level architecture. Clear layers, clear communication between
3. Layers are clear, might be a bit of weirdness in the "Domain Layer"? Some repetition
4. Good layers, generally clear. Maybe missing some mention of the diagnoses. Vestigial interfaces coming out the bottom should go away.
5. No clear layers of interaction here, just positioning. Empty boxes. Overall bad.
6. Some layers, but too "thick". Calls clearly one direction. Strange boxes.
7. Clear layers. Disconnected Customer Repository should be one layer down. Missing bottom of picture
8. Clear layers, pretty standard. Weird connections between components, though, text is a mess
9. There are layers, but generally seem disconnected from components listed.
10. Very bad text issues. Layers seem random, not like they are made with any specific organization in mind.
11. Clear layers, but the components are all weird.
12. Would be better w/out the disconnected external service and database and the duplicated repositories.
13. Layers generally present and clear. Services in one, repositories in another. Use of the component diagram symbols seems strange.

14. No real layers, just a chain of services
15. Layers are listed but don't seem to mean anything. Strange lines going nowhere
16. Nothing about divisions or branches? Pretty clear, though. Just a bit incomplete. Clear layers
17. Looks like inheritance relationships across layers, not interfaces. Strange lines throughout
18. Layers generally clear, cards on the right side are odd
19. Clear layers, the "office" at the bottom is strange
20. This one is just a mess. Boxes across multiple layers? Users in there like a use case? Bad.

Microservices

1. Looks like microservices. Connections between components do not match requirements
2. Duplicate "Setup Service", otherwise not too bad. Few connections among services, each a single responsibility
3. Way too many services for a high-level architecture diagram. Connections between incorrect services (e.g., sentence and picture)
4. Not sure what this does at all. Duplicate doctor service. Unconnected receptionist service. Some longer chains of association that could indicate failures.
5. Not too bad. Order should be connected with Orders?
6. Generally OK. Plot service should be removed.
7. Seems like a data design again, not a microservice design. Duplicate account service
8. Two disconnected graphs, that's odd. Some weird words in there ("consition"). Unclear why some of the connections are the way they are.
9. Duplicate assembly service needs to be removed, but otherwise ok. Assembly service is a single point of failure, but the "manager" services frequently are.

10. Generally ok except for correctness. Should be more connections between services like order and waiter.
11. Duplicate database. Generally not bad.
12. Generally ok.
13. Pretty good microservices. Correctness bad, duplicates of mulitple components, missing connections between them.
14. Some empty components in the diagram. Should be more connections to the database
15. Some services there, but not great. Barrier service functionality spread among multiple services.
16. Duplicate services. Branch and personal banker should connect
17. Shows inheritance, not microservices. Duplicate option service
18. Book database should be inside book service. "Boan"? Oddly connected history service
19. Overall clearly interacting microservices. Multiple doctors, data should be managed in service.
20. Not too bad. Multiple payment. Order should connect to product.

Event-Driven:

1. No event bus listed or representation of events. Strange text artifact on the right.
2. Does have a central event hub. Missing some components related to scenes, setups, etc. Strange connections between "repositories"
3. No events mentioned at all, still just a class diagram.
4. No event bus
5. Central event broker. Some odd connections among other components
6. Some lip service to events, but notation is strange. Really weird changes to the text here.
7. No central event bus. Covers the other types of account ok, strange component at the bottom

8. Disconnected event bus at the top, not used by the rest
9. Includes an event bus as a side component, does not seem used by the main processes
10. There is an event bus and other services, but only one service uses the events.
11. Generally clear event workflow. Connects orders and couriers, at least
12. Overall bad. Strange lines, empty components, no event bus
13. Overall bad. Duplicate components, poorly connected event queue
14. This one is nice. Would include a "client" component, drop the spare database
15. No event bus, odd clouds? Not exactly sure what it all means
16. Event bus doesn't mean anything here. Components have multiple connections to each other, overall odd
17. Event bus seems there just to satisfy the words. No real connection. Disconnected services
18. No event bus. Seems to have most of the pieces, but hard to see connections for a system
19. Not too bad, if we argue the event queue is managed by the central service
20. Event queue not a player here. Hard to say how the system operates. Abandoned, unlabeled interfaces, overall bad.