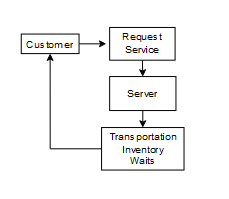
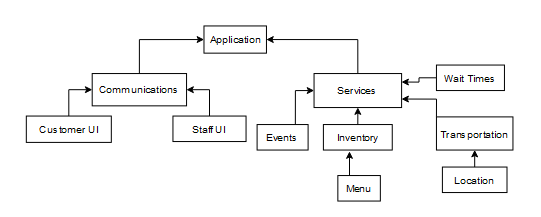


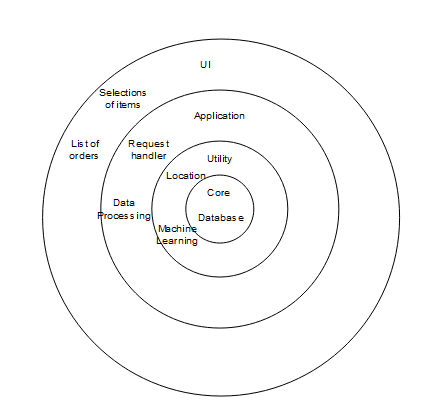
UML Diagram for big picture



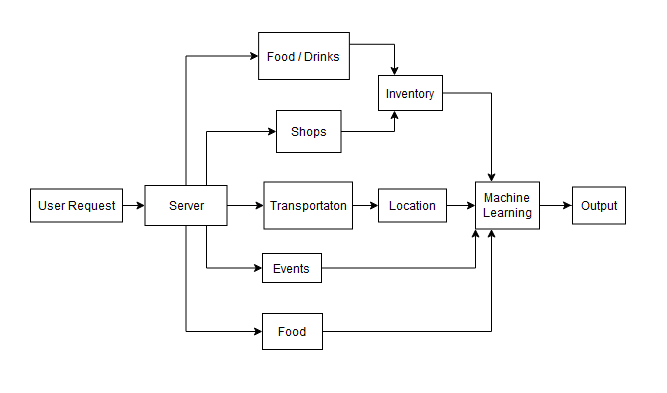
Object Oriented Diagram



Component Architecture



Layered Architecture



Data Flow Architecture

The component architecture shows how components are connected in the entire system. This helps design the architecture through a hierarchy of what component owns which other pieces through direct and indirect connections. This is useful in order to lay out dependencies. This can be approached by working on the smaller branches and working up a single branch to ensure that the entire “module” is completed.

The layered architecture shows how the different components are grouped. It also shows which components depend on other components. The outer layers depend on inner layers. The way to approach this scenario is to work from the inside and progress outward allowing the outer components to add to the inner ones.

The data flow architecture shows how data moves throughout the application. This comes in handy to show which data needs to go where in the system. It also shows how many elements data will need to flow through to get to a certain point in the system.

The advantages of the component architecture is that it allows you to see the connections between the different modules. This is similar to how classes are designed which can make it easier to write classes. The disadvantages are that a designer may not know where to start. Layered architecture is good to show the different levels that the architecture needs by grouping similar elements together. This helps design you design elements that required to build the next component that depends on it. Data flow is good to show how data moves throughout the system to help visualize what components take and send to other components. This design may be harder to split workloads because it isn’t linear on how things are connected.

I believe that component architecture design is the best way because it can help you quickly develop the application starting from the bottom of a branch. This way, a group of developers can each pick a different branch and eventually meet at the end result together.