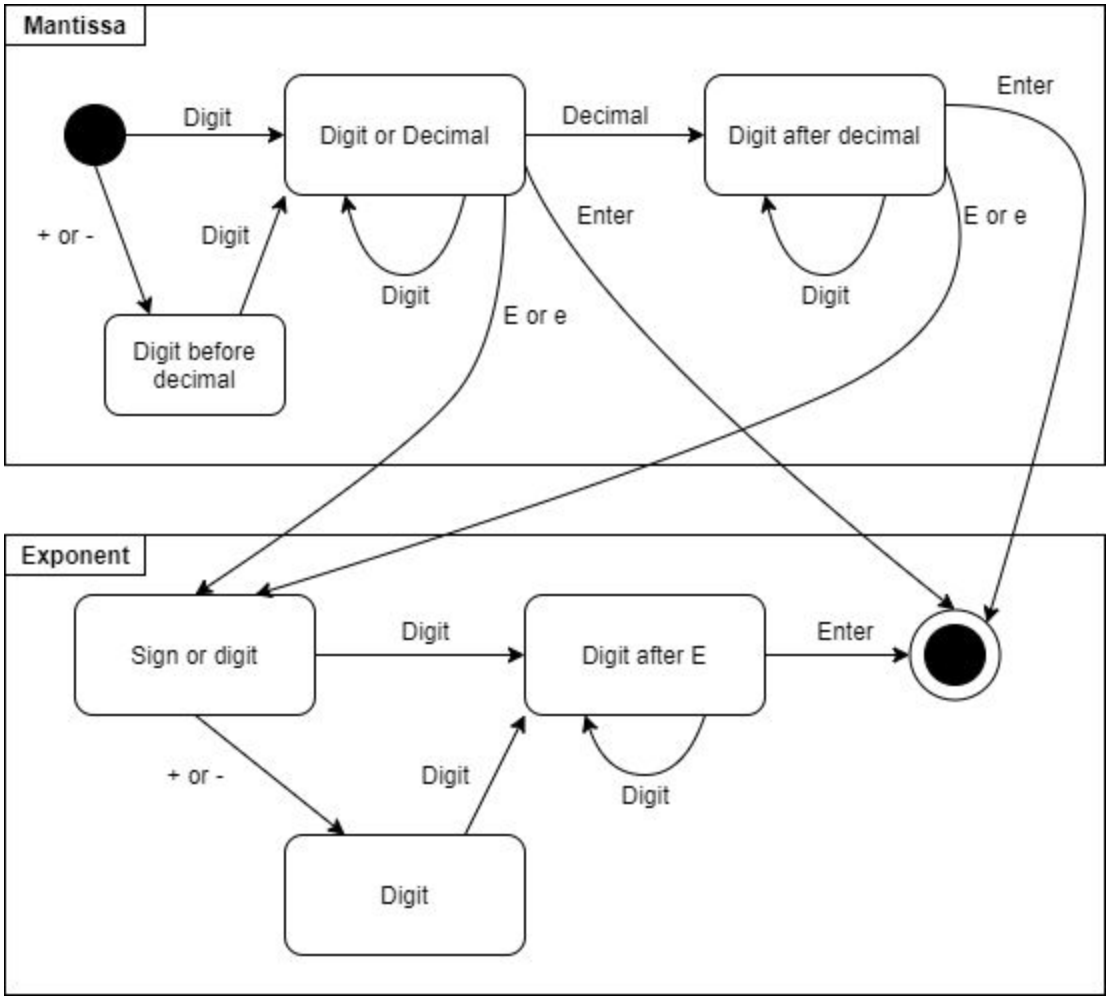
5.1 Component-based architecture deals with pieces of the system as components that provide services to one another. A service oriented architecture is similar, but it's pieces are services usually running from other computers across a common network. A service oriented architecture is just a little more spread out.

5.2 For this Tic-Tac-Toe app, a monolithic architecture is most likely sufficient since it is a small, standalone application. In the case of the user interface, making it event-driven makes the most sense for responding to user events when playing the game for events such as making a move, starting a new game, or checking a score. So the best choice for such a simple app is a monolithic data-centric approach.

5.4 For a chess game against a real player over the internet, we would definitely want to go with a data-centric approach like Tic-Tac-Toe since we would have tables of moves on the board and the subsequent results. Due to the online nature of this application, a service-oriented approach would be useful in spreading the application across multiple devices and communicating via the Internet. So our resulting approach is a monolithic data-centric service-oriented application.

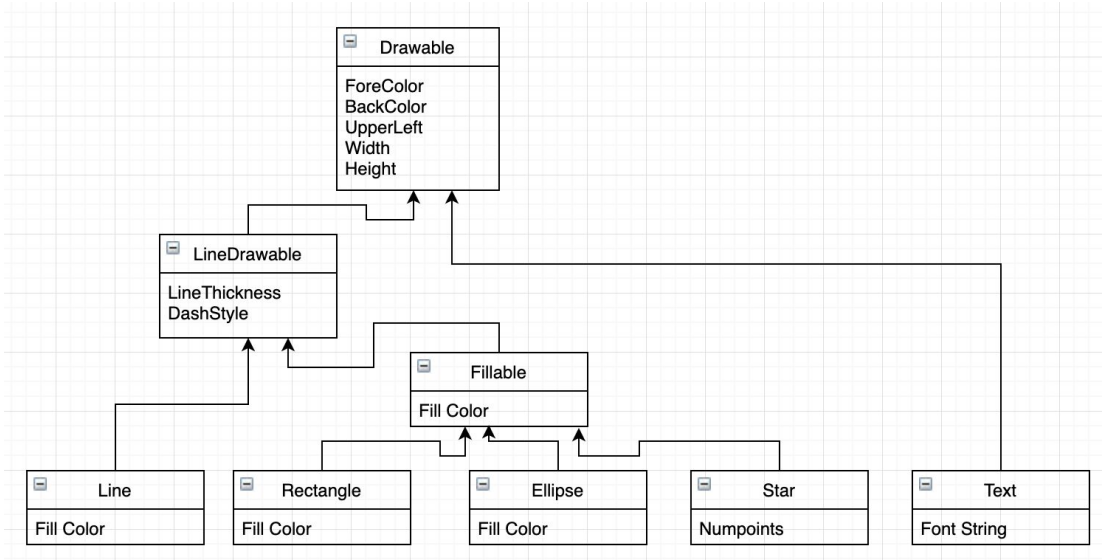
5.6 The ClassyDraw app can store each drawing in a separate file, so it doesn't need a database. Basic operating system tools will allow a user to create, add, delete, and manage all files. This program could add a temporary save state function that saves a users file temporarily in case of crash or file loss.

5.8

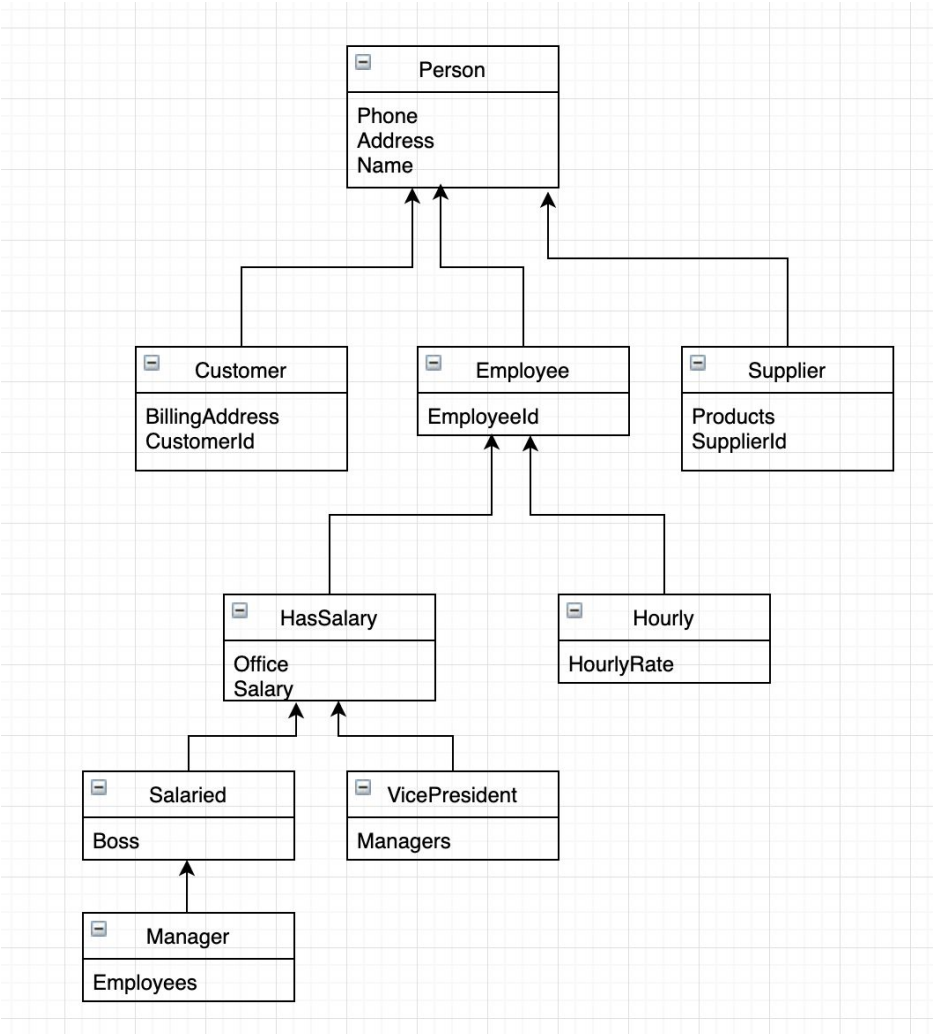


6.1 All of these classes represent drawings or components of drawings. They all share properties used for drawing such as color. They also all store their position using height and width at the leftmost upper corner. Some of these classes vary by the data that is needed in order to draw them. For example, the Text class needs font information and the string to draw. The Star class needs to know how many points to give the star. Rectangle, Ellipse, and Star can be filled, so they need a fill color. The classes that draw lines also need properties such as thickness and dash style. These classes are Line, Rectangle, Ellipse, and Star. The property ForeColor is used by all. The property BackColor is used by all. The property UpperLeft is used by all. The property Width is used by all. The property Height is used by all. The property Font is used by Text. The property String is used by Text. The property NumPoints is used by Star. The property FillColor is used by Rectangle, Ellipse, and Star. The property LineThickness is used by Rectangle, Ellipse, Star, and Line. The property DashStyle is used by Rectangle, Ellipse, Star, and Line.

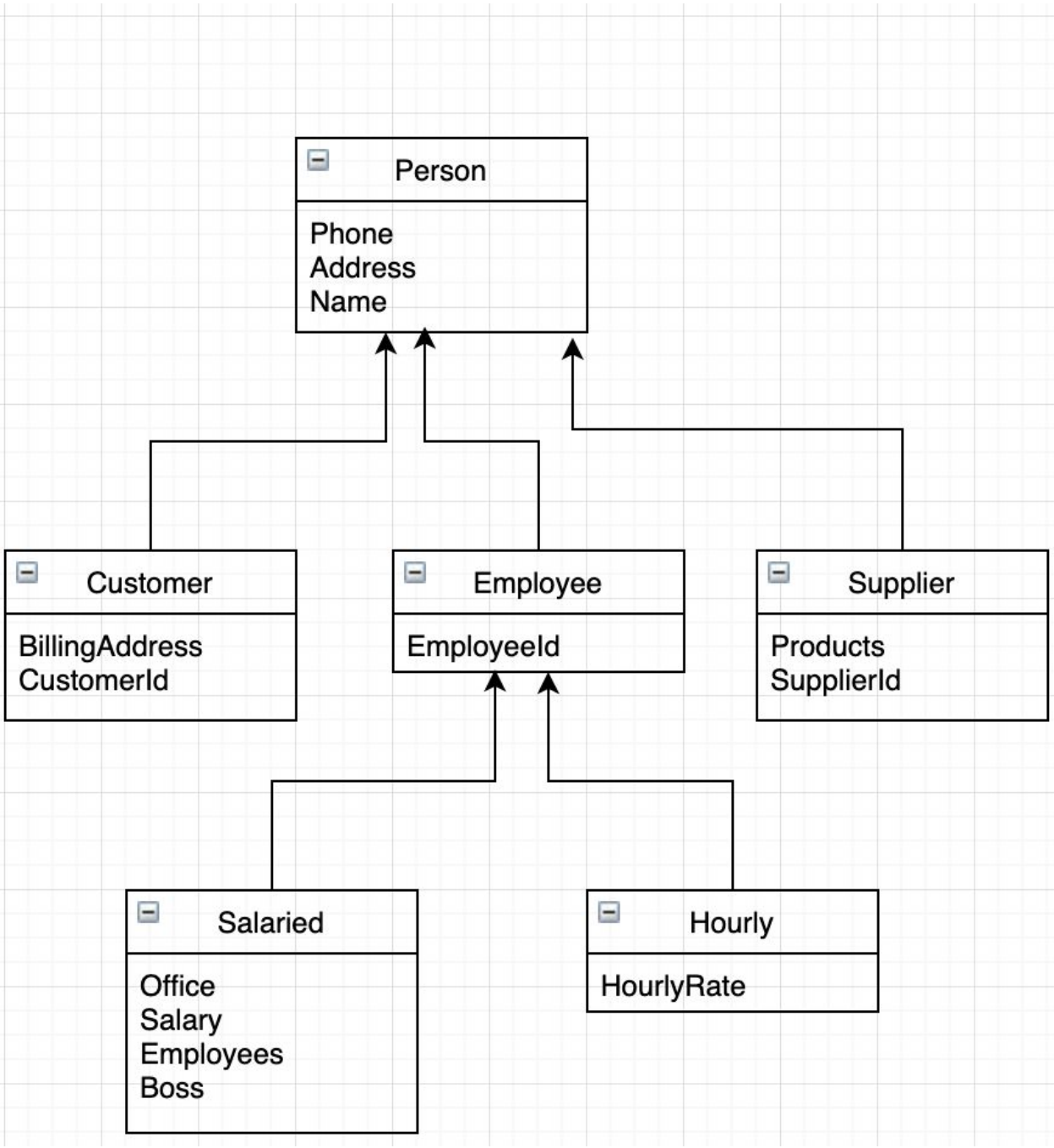
6.2


Drawable
ForeColor
BackColor
UpperLeft
Width
Height
LineDrawable
LineThickness
DashStyle
Fillable
Fill Color
Line
Fill Color
Rectangle
Fill Color
Ellipse
Fill Color
Star
Numpoints
Text
Font String

6.3



6.6 In order to combine the managerial types we could give the salaried class the properties: office, salary, boss, and employees. Therefore the boss property will not be filled for top-level vice presidents that have no one to report to and the employees property will be empty for bottom level employees that are not managers of anything.


Person
Phone
Address
Name
Customer
BillingAddress
CustomerId
Employee
EmployeeId
Supplier
Products
SupplierId
Salaried
Office
Salary
Employees
Boss
Hourly
HourlyRate