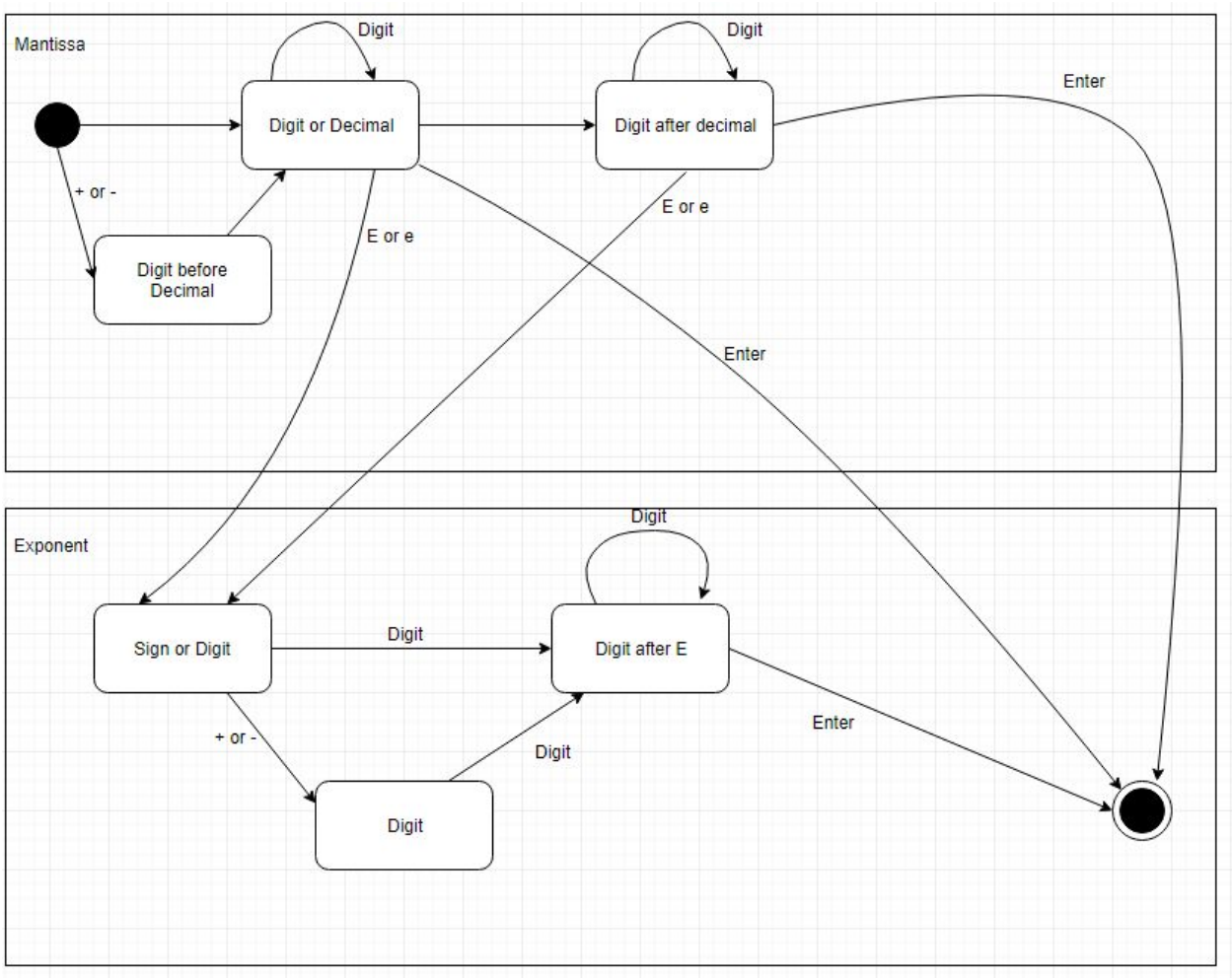
Ronald Uy
CMSI 402 - Assignment #2

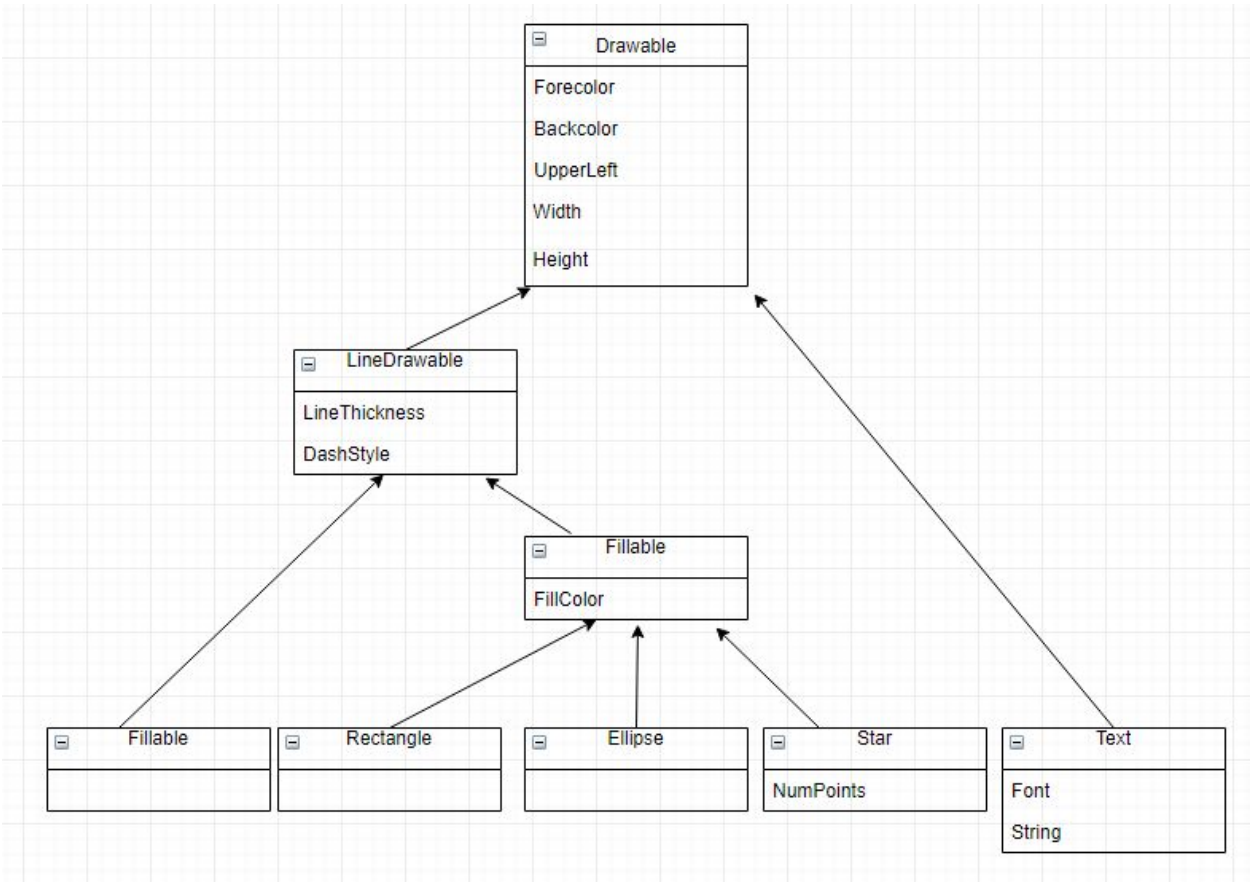
Chapter 5

1. A component-based architecture regards pieces of the system as loosely coupled components that provide services for each other. A service oriented architecture is similar except that the pieces are implemented as services being run on a computer via a network. The big difference is that the pieces of a service oriented architecture are more separated compared to a component based architecture.
2. Based on the description of the phone application, this is a self-contained application with no remote services or databases required, which makes things like a component based and service oriented architectures are probably massive overkill for an application like this. For this application, a monolithic architecture would probably be the best option because it is a relatively small, self-contained app. You could probably also go with a data-centric approach, especially for tic-tac-toe because you could easily build tables and the best responses per move.
4. In regards to the chess program, it would be similar to my response in question 2, that a monolithic rule-based architecture would work best for this type of application. However, things that would have to be kept in mind are the fact that the program needs to exchange information with another instance of the program on the internet, and also that there would be no AI opponent. Keeping those in mind, I believe that this application would still need to be a monolithic rule-based application, but also being service-oriented.
6. The ClassyDraw application can store each drawing in a separate file, so there would be no need for a database really. One possibility would be that the OS tools allows the user to manage the files, in other words, the user can delete/backup copies of files. Another option would be to have the program create temp files while the user is creating the drawing & if the program crashes or ends prematurely, ask if they would want to restore the temp file (similar to how Microsoft Word recovers documents).

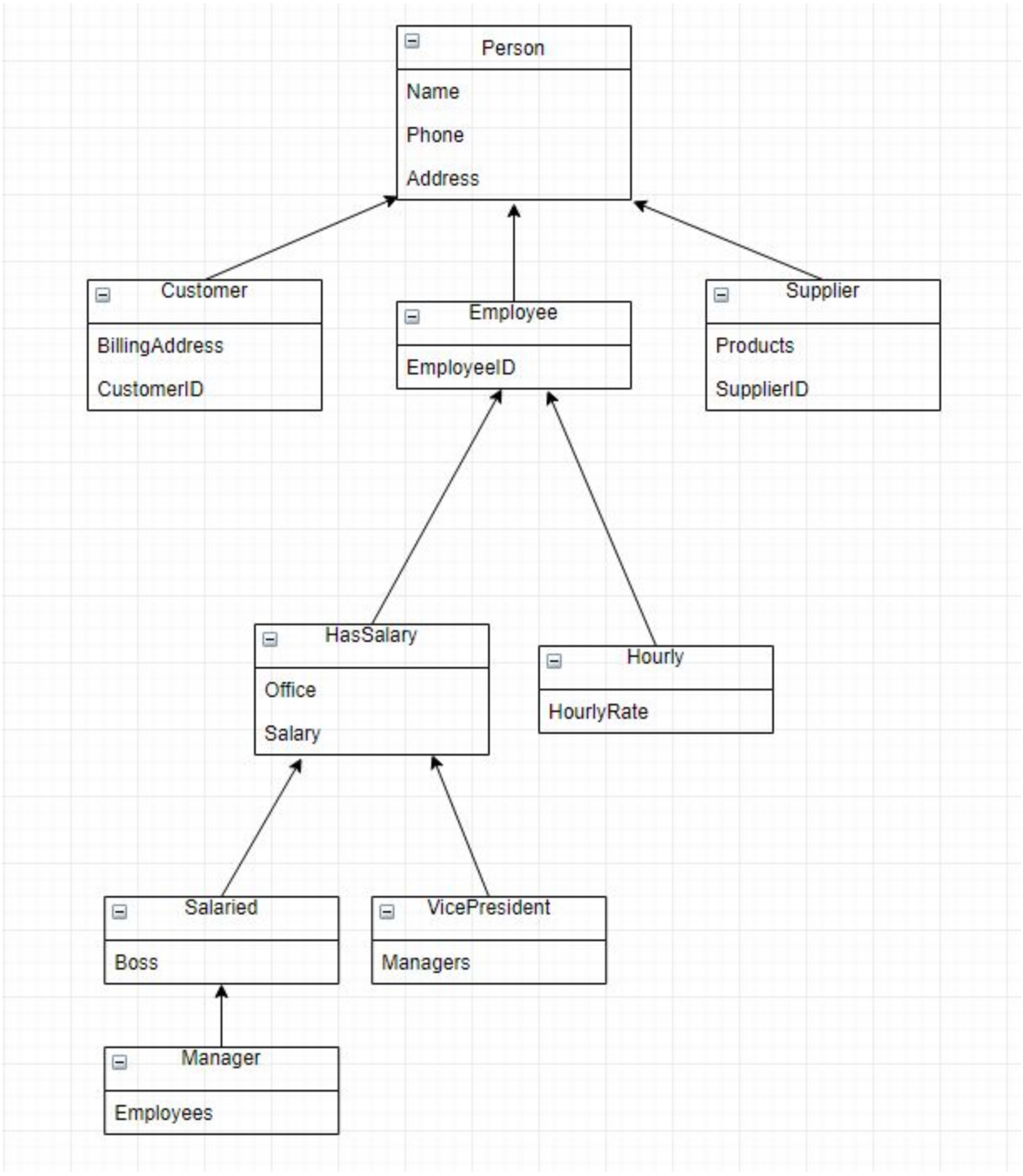
8. 

Chapter 6

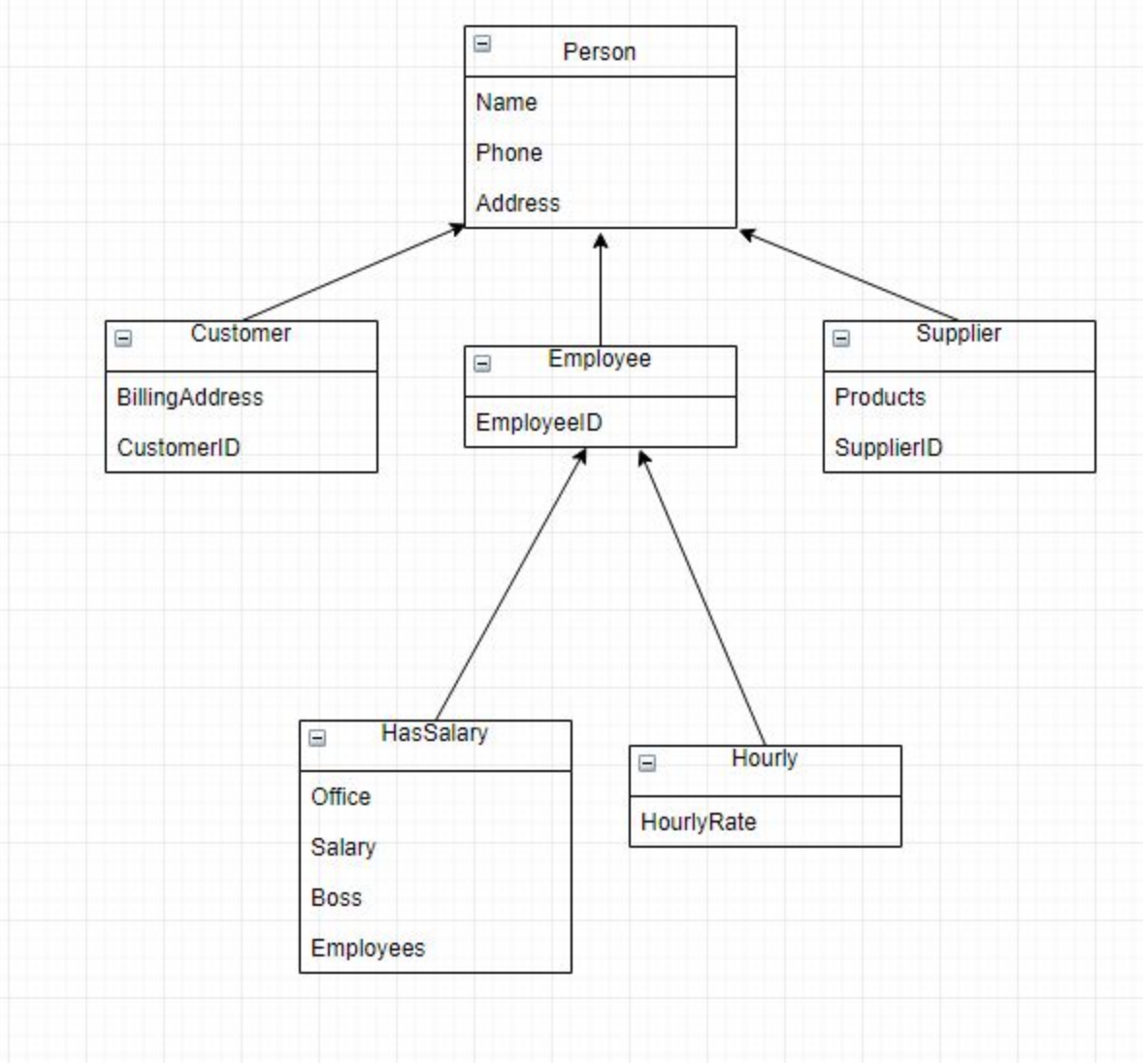
1. The classes: Line, Rectangle, Ellipse, Star, and Text, all represent things that are drawn, meaning that these classes would all share properties (such as foreground & background color). Other properties would include: drawing position, width, and a height. Some classes will need extra data to draw a particular shape for example, and the other classes wouldn't share that data. The Text class for example would need font information & the string to draw things, whereas the Star class needs to know how many points to give the star. Likewise, some properties can be shared with certain classes and not others.

2. 
Drawable
Forecolor
Backcolor
UpperLeft
Width
Height
LineDrawable
LineThickness
DashStyle
Fillable
FillColor
Fillable
Rectangle
Ellipse
Star
NumPoints
Text
Font
String

3.


Person
Name
Phone
Address
Customer
BillingAddress
CustomerID
Employee
EmployeeID
Supplier
Products
SupplierID
HasSalary
Office
Salary
Hourly
HourlyRate
Salaried
Boss
VicePresident
Managers
Manager
Employees

6.


Person
Name
Phone
Address
Customer
BillingAddress
CustomerID
Employee
EmployeeID
Supplier
Products
SupplierID
HasSalary
Office
Salary
Boss
Employees
Hourly
HourlyRate