# *Lab 10 – Package & Deployment Diagrams*

Date assigned: Monday, March 20, 2017

Date due: **Monday, March 20 14:50**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Learn to divide systems into packages
* Build UML Package Diagrams
* Build UML Deployment Diagrams

To do:

Save this document as a Word document named **YourUserName\_E21\_L10\_Package.docx** in your 420-E21 folderin your home drive. The document will hold your answers for your lab.

# Packaging strategies

1. In class we talked about various ways to create packages for a system. They were: by architectural layer, by subsystem, by functionality, by deployment. For each of the following situations, determine what way or ways you would use to package the system and why.
   1. Your company provides a series of Java libraries that can be used to create business and accounting applications for other parties. The library system you provide is in the form of an Application Programming Interface (API) for various accounting processes such as Accounts Receivable, Accounts Payable, Audits, Budgeting, etc. There are two levels of the system that can be purchased, one for basic accounting and one for advanced accounting. Your company provides a user document explaining how to use the classes and methods in your library, but does not provide details on the inner workings of the classes nor does it provide source code.

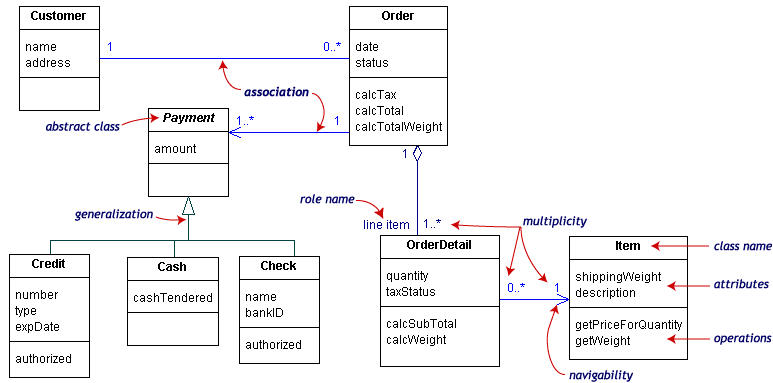
By deployment - There are multiple different complete systems that work in here, with different ones being a step up above another. They’re each their own system, one that builds functionality off of another.

* 1. An alumni system for the college that allows alumni to register with the system to maintain contact with the college and college events and to keep in touch with other alumni. The alumni use a web application to register, log in, set up their profile, read college news and search for other alumni. The college has an administrator for the system who uses a desktop based application to add and remove news, update alumni, administer forgotten passwords, email alumni and run various reports.

By Access Control/subsystem - There are multiple distinct subsystems, each of which would have use cases, so this would be packaged by each of those subsystems.

* 1. A co op system for the college that provides students, administration and employers different functionality. Students can register for co op and confirm their eligibility. They can also search jobs and apply for jobs online using the system. Administrators can confirm students’ eligibility, confirm employers, review jobs, confirm applications and appointments and run reports. Employers can register as employers, add jobs and review applications. Employers and students use a web-based interface and administrators use a desktop application.

By Access Control/subsystem - Student, admins and employees are all different subsystems that have their own subset of use cases based on who has access to what.

* 1. A system based on the following class diagram which models a customer order from a retail catalog. The central class is the Order. Associated with it are the Customer making the purchase and the Payment. A Payment is one of three kinds: Cash, Check, or Credit. The order contains OrderDetails (line items), each with its associatedItem  
     

By functionality - Different types of functionality without it being different subsystems.

* 1. An ordering system wherein the system can be used on many different platforms. The system takes in orders and stores them in a customer database system. The orders are sent to a shipping warehouse which checks the stock database to see if the ordered items are available. The orders are also sent to the accounting system which interacts with the bank and the customer to receive payments. Although the system will only be released initially for the web, there is a requirement to make the system dynamic enough to be implemented on other platforms. As well, the customer database and the stock database may be at different locations and may be different databases. As the company grows they expect to change the database at either location.

By Architecture layers- shows differences between the DB layer, logic layers and UI layers.

* 1. A system consists of 5 discrete pieces of functionality. A customer may purchase any or all of the pieces of functionality; however, they must purchase and install them in order; that is, Package A, then Package B, then Package C, then Package D and then Package E.

By Deployment – The different packages can be installed at any level, but it works it’s way down to the basic functionality and build up from there. The deployment builds off of itself

* 1. You are producing a system to be used within your company. The system has 5 independent modules. All 5 modules of the system must be released at the same time. The system will be released iteratively and incrementally in a series of releases.

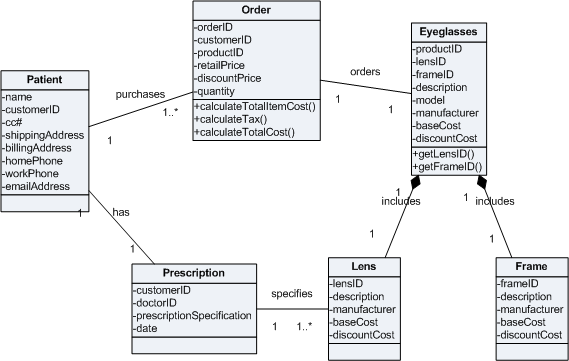
By functionality

* 1. A web application integrated development environment (IDE) which can be purchased with different functionality. For example, you can purchase the complete package which includes 6 different compilers, a debugger and a packager; or they can purchase the professional package which only includes 4 compilers and the debugger. The following packages are also available: standard, introductory, basic and demo.

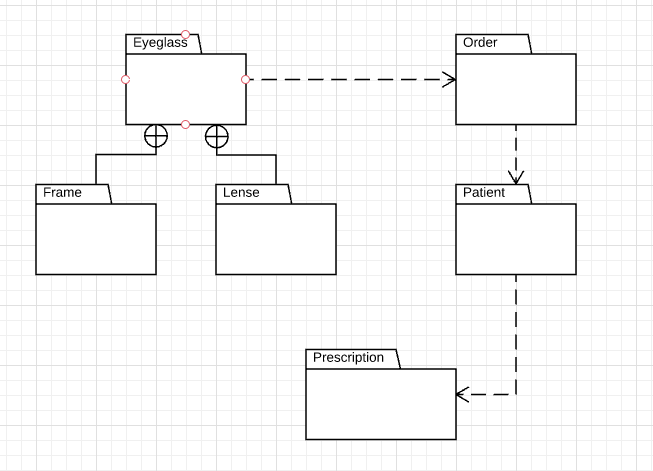
By deployment – Different levels of functionality. Some things will be released in certain releases, but not in others.

# Package Diagrams

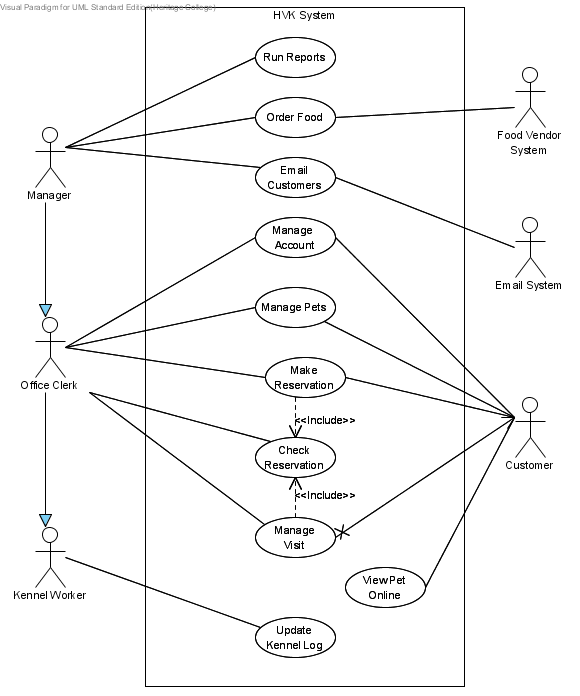
1. Given the following class diagram for Purchasing Eyeglasses, complete a class package diagram in Lucidchart and paste it into this document. Give appropriate names to the packages. Remember to include dependencies.



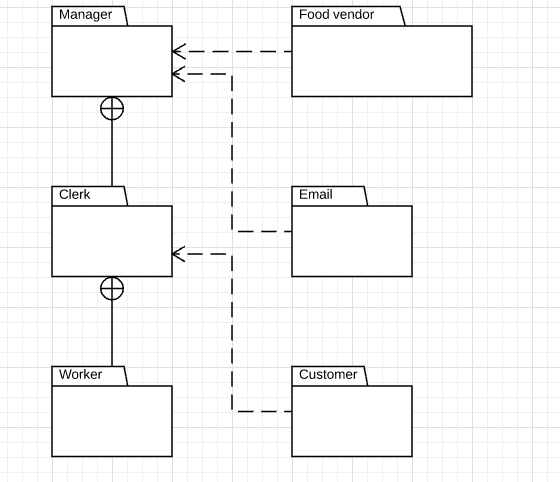
**Insert diagram here.**



1. Given the following use case diagram for Happy Valley Kennels, complete a use case package diagram in Lucidchart. Give appropriate names to the packages. Remember to include dependencies.



**Insert diagram here.**



# Deployment Diagram

1. Create a deployment diagram for an system that consists of:
   1. Web pages running on Windows 10 machines, on the latest version of the MS Edge Browser. This is the direct interfae with the end user. The client side code runs the typical HTML5 JavaScript, jQuery, bootstrap ASP.NET client-side combination.
   2. Web application Server. Serves up the Web pages, also runs the Server side ASP.NET portion of the application. This runs on a Dell PowerEdge R230, Host name csprod1.cegep-heritage.qc.ca. It is running Windows Server 2016 with the latest ASP.net framework.
   3. The web application code accesses an Oracle 11.g database running on csoracle.cegep-heritage.qc.ca, a Windows Server 2008 running on a HP ProLiant BL460c Gen9 Server Blade.

Be sure to show the network, physical box, OS and software environment in your diagrams. Also, show the high level protocols between deployment components.

**Insert diagram here.**

To Submit

When you have completed the assignment, upload the **YourUserName\_E21\_L10\_Package.docx** document to Moodle.