# *Assignment 3 –State, Package and Deployment Diagrams*

Date assigned: Monday, March 14, 2016

Date due: **Wednesday, March 30, 2016, 11:50pm**

**Learning Objectives**

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

* Draw State Machine Diagram.
* Package and Deployment Diagram
* Identify tasks for sprint planning

To do:

This is an individual assignment. There is no group work involved.

# Part A: Statechart Diagrams

Note: Be sure to use the “trigger[condition]/action” syntax discussed in labs and lectures and proper UML syntax.

All actions are explicit, don’t assume an undocumented action will happen.

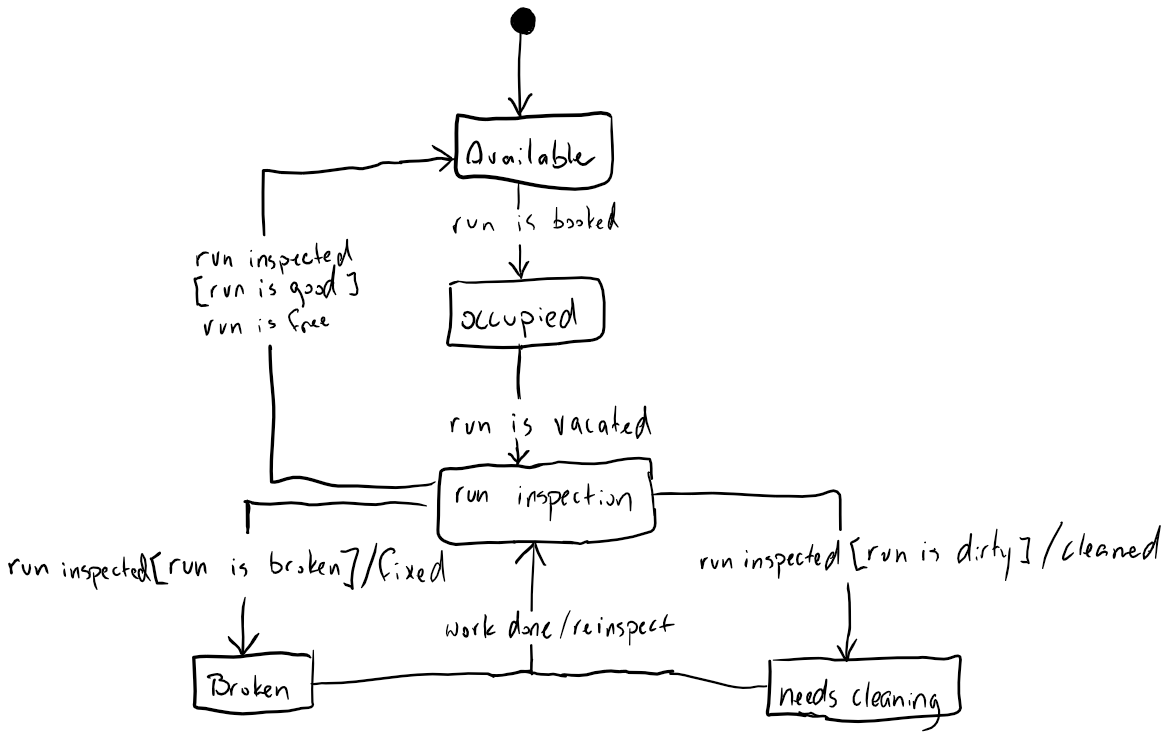
Triggers are a result of a requested action or an asynchronous event.

Conditions are a setup as a result of actions.

1. For the HVK project: Draw the state machine diagram for the states that a run can be in.

|  |  |  |
| --- | --- | --- |
| **Status** | **Meaning** | **Explanation** |
| 1 | Available | The run has been cleaned and is available for a dog |
| 2 | Needs cleaning | The run needs cleaning |
| 3 | Broken | The run needs repair |
| 4 | Occupied | The run is occupied |

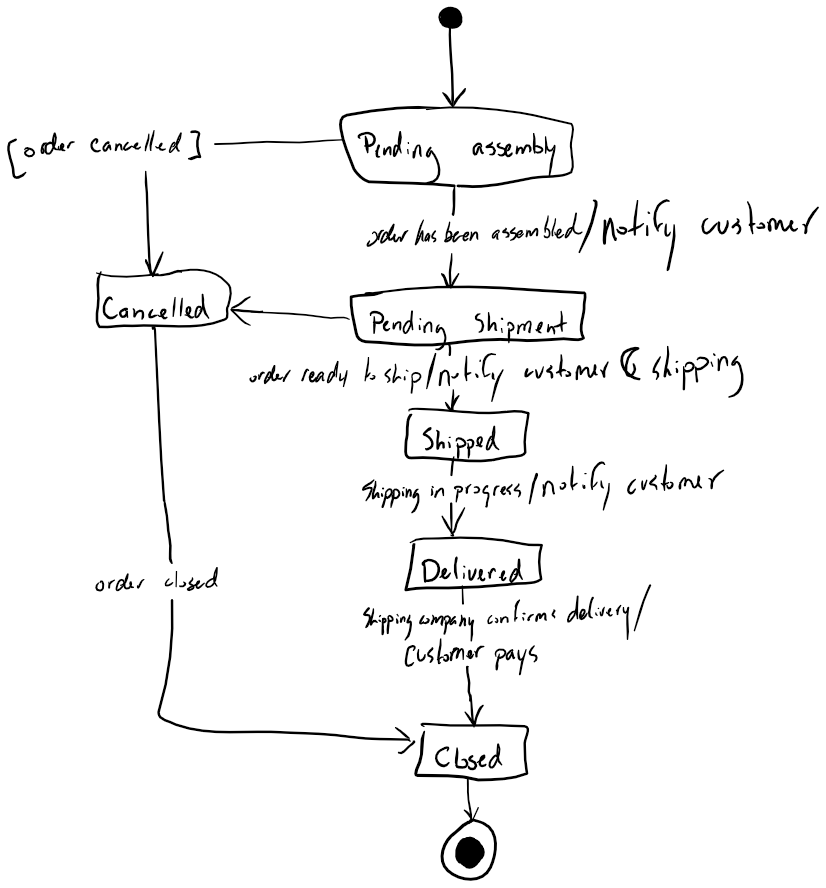
**Insert statechart diagram here:**



1. For the Shopping cart project: Draw a state machine diagram of the states of an Order. An Order can be:
   1. Pending assembly – order is being assembled
   2. Pending Shipment – order is ready and awaiting shipment
   3. Shipped – order is in the process of being shipped
   4. Delivered – confirmation from the shipping company that the product has been delivered
   5. Closed – order has been paid for and delivered.
   6. Cancelled – order has been cancelled.

Orders are created on checkout. You may add more states if you deem it necessary.

**Insert statechart diagram here:**



# Part B: Sprint planning

This is a simulation of sprint planning and effort sizing. In a Sprint planning meeting, the product owner sets the priorities for the sprint, but the Sprint team must balance the priorities against what can actually be accomplished.

1. From your Assignment 2 shopping cart project. As a first sprint, the business owner has set priorities and wants to be able to “Browse Catalog”, “View Details” and “Add to Shopping cart”, and “View shopping cart” and “Checkout” in priority order. Based on the analysis done in the Sequence diagrams and the Class diagrams, you will determine what you and your team can commit to for the sprint task.
   1. Identify which Classes and public methods are needed. Show me a list of Classes and public methods

|  |  |
| --- | --- |
| **Class** | **Methods** |
| Item | * getDetails * checkAvailability * getStock |
| Customer | * removeFromAccount * depositToAccount |
| ShoppingCart | * addToShoppingCart * removeFromShoppingCart * proceedToCheckout * getTotalCost * handleTransaction |
| Catalogue | * loadMoreItems * filterResults * nextPage |
| UserInterface | * displayCatalogue * displayShoppingCart * displayCustomerInfo |

1. Size each task in terms of effort(hours). Analyze each task to determine what pre-requisites you’d need. You many need to add more tasks (such as setups/installs, test design, test data design, test execution).
2. What is your total effort required for this goal? Will it fit if you have a team sprint capacity of 80 hours?

To hand in: a table or spreadsheet showing a list of tasks, for each task – a description, effort estimate and pre-requisite tasks identified (if any), a total of the effort. Conclude with a commitment on what can be accomplished within the sprint capacity. The task list should be in sequence order (i.e. tasks that should be done earlier are at the top of the list).

Note: once you are beyond the sprint capacity of 80 hours, you do not have to continue identifying and sizing tasks.

Note: in reality, teams don’t estimate in hours. We’re only doing this now as a learning step. You will learn about more advanced story points and Agile estimation techniques in Systems III.

**Insert task table/spreadsheet here:**

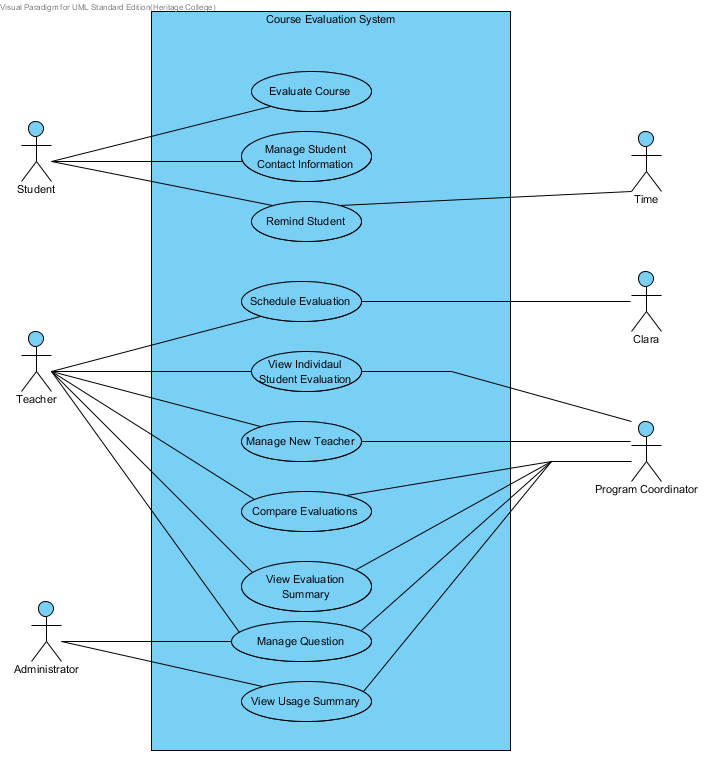
|  |  |  |
| --- | --- | --- |
| **Task** | **Time** | **Description of task** |
| Model / Create Database | 12 | Create the DB for the site to use. Using the right tools, you won’t need to code at all, so this shouldn’t take too long. You just need to make sure the DB is solid and doesn’t have normalization issues or traps. |
| Write test cases | 12 | Need to have the user stories, which presumably we already have since we’ve started the sprint cycle and have a sprint burndown. Writing the test cases should be done asynchronously with the rest of the work. Things should be tested as you go and testing should always be in a day’s work. |
| Design UI | 8 | This is going to be hard since UI design is also tedious. People in my team often feel like they aren’t satisfied with their work after working with UI, so we’re giving enough time to make a nice one, but not enough time to built something that will make us all satisfied with it. |
| UserInterface Class | 10 | The UI class should be relatively simple. The class won’t need to do any work itself, just return all the data and the output of what the other classes are doing. It’s kind of in the middle of the business logic classes and the UI itself. The UI class should be done asynchronously with the rest of the classes because it’s the class that’s going to be calling all of it. When something gets done, the UI class should implement the usage of it and put it into the actual system UI. |
| Item Class | 6 | The Item class is pretty simple, most of what it does is just return information and does very little actual work. The longest thing in here will be the Database lookups for the availability and checking how many are in stock, but that can be a single lookup and just using the results differently. This class is easy. |
| Catalogue | 8 | The catalogue class might be a little complicated since it needs to do lots of reads from the database, but otherwise it’s simple, it just spits out data like Item. |
| ShoppingCart Class | 12 | Shopping cart will be the hardest class for sure. It has the most interactions with the database so it will have bit more code than the others. AddToShoppingCart, ViewShoppingCart and checkout are all part of the shopping cart class, so most of the work will be here. The ShoppingCart also needs to communicate with the warehouse external system to keep the customer up to date on the progress of their order. It will need to communicate with the payment system to handle the actual money transaction as well. |
| Customer Class | 8 | The customer class also isn’t too hard. All it does is return data to the UI. To use in the shopping cart. The reason that customer takes a while is because it needs to handle the customer’s bank account, which means learning how to interact with the external banking system. |

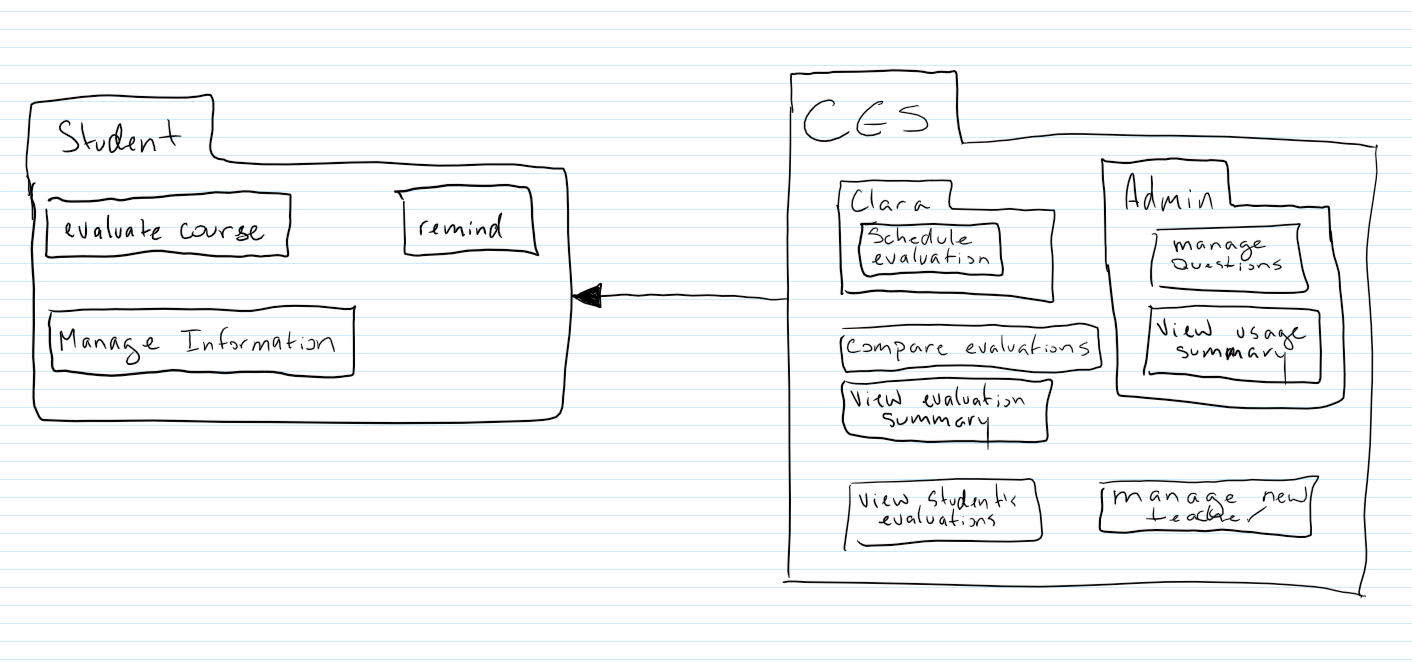
**Declare: What is your sprint goal?**

I think that realistically, you can get all the above working in two weeks with a team of 4, and that should be the goal. There’s a lot of extra stuff that would need to go in after, but getting the basics down now should be the priority. Make it work before you make it work well. The code doesn’t need to be perfect, the DB doesn’t need to be perfect, it needs to work for the time being to have something to show the customer. In the future, it should implement searches through the catalogue and things like updates and deletes to the shopping card and stuff, but for the time being, I think that in this sprint it’s all doable.

# Part C: Package and Deployment Diagrams

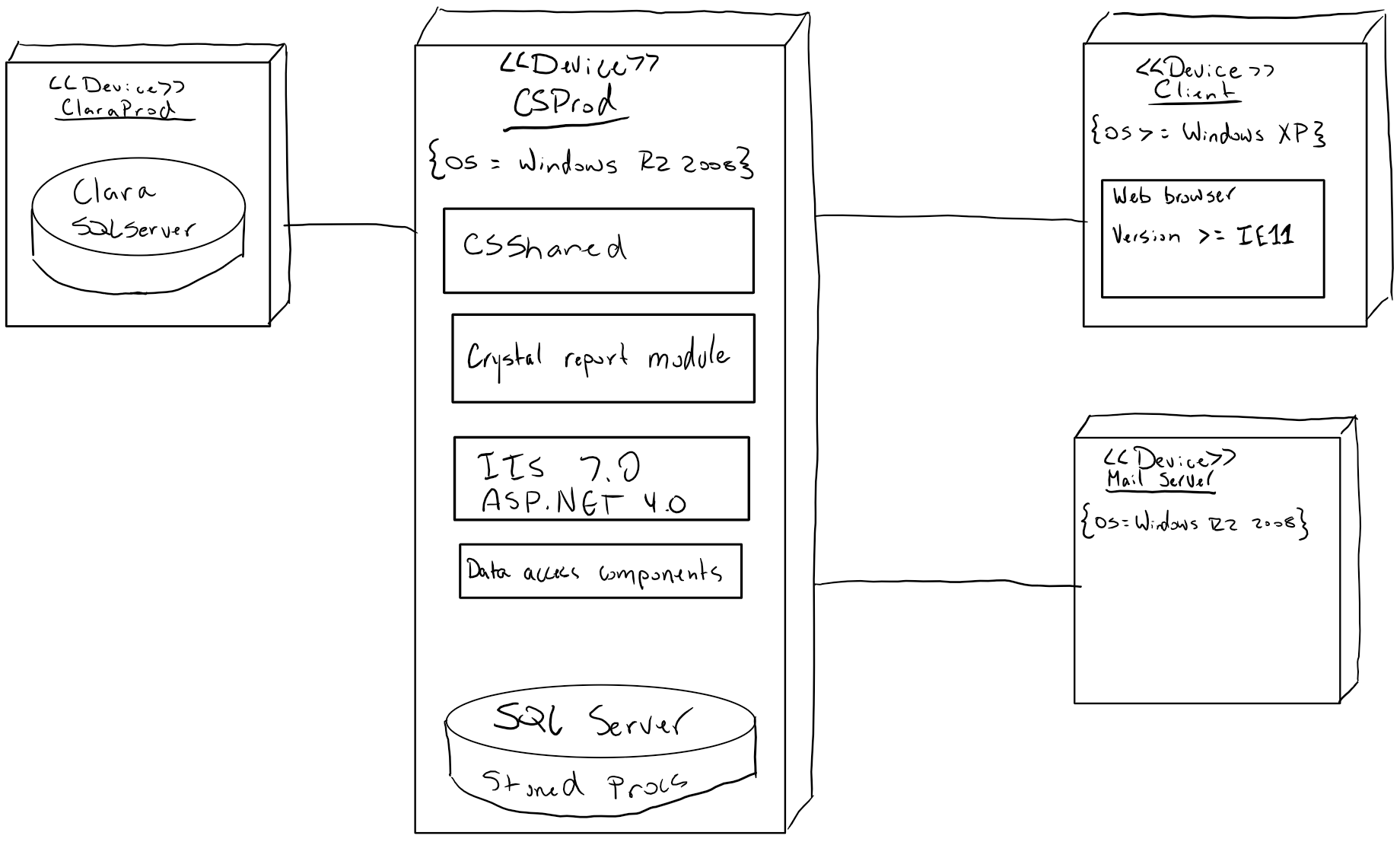
1. Below is the use case diagram for Manage and View Evaluations in the Course Evaluation System. Draw a package diagram from this use case diagram.





1. Complete the deployment diagram (with the contained components) matching the following specification for the Heritage College Course Evaluation System (CES). Remember to include only items that are relevant to the deployment.
   1. The CES uses ASP.NET for the web portion and SQLServer 2008 for the database.
   2. The CES runs on a Web server called CSProd which runs Windows 2008 Server with IIS 7.0.
   3. The CES interfaces to the CSShared application, which also runs on CSProd.
   4. The CES runs a Crystal Report module, which also runs on CSProd.
   5. The client application is accessed over the Internet from a desktop machine running a Web browser. The machine must run Windows XP or higher.
   6. The Web server interfaces to the Clara database which resides on a node called ClaraProd, using SQL Server
   7. The Web client runs in IE 11 or higher.
   8. The Web application uses ASP.NET within the .NET 4.0 or above framework.
   9. The Web application communicates with a Microsoft Exchange mail server which runs on a Windows 2008 server and communicates with CSProd via SMTP.
   10. In order to maintain corporate standards, all access to the database must be done through a separate Data Access Component that communicates with the SQL Server. All access to the SQL Server database schema is done through stored procedures.

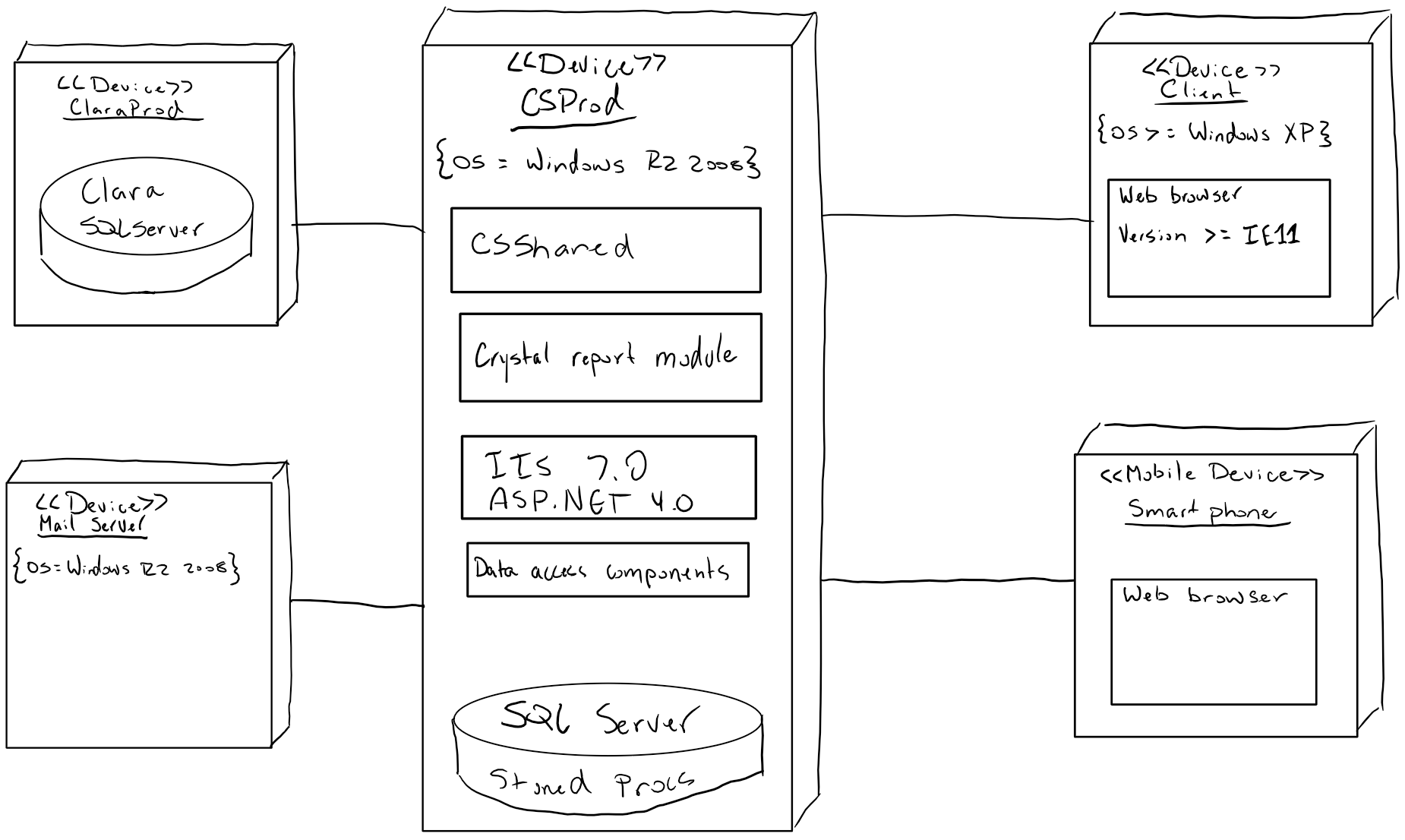
**Insert diagram here:**



1. The college wants to change the CES system so that it can be accessed via Smartphone or mobile device. Copy your diagram and add any new node(s) required to add this functionality.



**Insert diagram here:**



# Part D – Assignment assessment

1. What did you learn in completing this assignment?

In this assignment, I think that the biggest thing I learned was just syntax for deployment diagrams.

1. What did you have difficulty with?

The thing I had the most difficulty with was Part B I think. People always tend to underestimate the amount of work required for building things because people always focus on making things perfect and not making them so that they just work. A lot of other people just took an estimate and then doubled it to be safe, but it’s probably better to make a realistic estimes.

1. What did you do well?

I feel like I did the statechart diagrams well. The statechart wasn’t too hard, but I really thought about the actions, guard conditions and results for all of the lines and I feel like understanding the syntax for them is absolutely key.

1. How many hours did you spend in completing this assignment?

I think the assignment took me around 6 or 7 hours of work.

1. What took you the most time?

What took the most time was Part B. I ended up doing it once from the perspective of each of the main actions that could be taken and didn’t think about it from the perspective of each method in the different classes, so I had a lot missing and ended up re-doing it.

Mark Breakdown:

|  |  |
| --- | --- |
| Part A – Statechart |  |
| HVK Runs | 10 |
| Shopping Cart Order | 10 |
| Part B – Sprint Planning |  |
| Spreadsheet showing list of tasks, description, effort estimate/total, | 10 |
| Tasks are consistent with Class and Sequence Diagram of Assignment #2 | 5 |
| Sprint Goal commitment, Task list are sequenced and sized reasonably | 5 |
| Part C – Package and Deployment Diagrams |  |
| Package diagram | 10 |
| Deployment diagram | 10 |
| Deployment diagram + Mobile | 5 |
| Part D – Assessment, English, properly handed in | 7 |

**To submit**

* Update this document with the requested updates and submit to Moodle.