This is a document to go with the design pages to show some of the thought process I used to come up with my initial design decisions. I included most of my notes and an unmarked copy of the text file I used to keep track of my thought during design. Unfortunately, I removed all my design and working comments I had in the source code. I create these comments as I work to flesh out what I need then remove them once I have finished.

First thing I decided was to create the first application using .NET core and entity framework. That way I could use EF’s scaffolding to prototype quicker. This would allow me to spend more time with the design of the application and not on the UI. And with using migrations I could leave the final database design until after the data model became solid. This also allowed me to wait until a later refactoring to split the view model and the data model.

As seen by the accompanying documents, the next thing I did was to start thinking of those objects I imagined I would be dealing with as I developed the application. Coming up with reasons it needed to exist. What its purpose was to be and what other things were related to it. I do this as a separate design document prior to ever writing the first line of code. This allows me to better understand the design and lets me add, change or delete different aspects so that I do not go down a design that might have to change later. Changing a design document is a whole lot easier to do then code.

Next, I picked one of the design specs and started fleshing out my thoughts on it to come up with the data for it, how I wanted to work with it. Since this was a prototype, I let the tools, EF, dictate how the first design would look like. I could later refactor into a more permanent design as I mentioned above.

After scaffolding the Job and Family in EF with the data model, I started changing that scaffolding to interact. This led me to write supporting methods, create new objects in the design, etc. Once I had all the main item scaffolded to where I could simply add/change/delete their associated database records through the EF, I moved on the working in the billing issue.

I went through a half a dozen refactoring’s before I came up with a basic method for dealing the variables of start, end and bed times to determine each number of hours for the three billing periods, early evening, late evening and overnight. This is method that will refactored into a better OOD strategy. This will eventually drive the finished design of the whole application.

I finished all the UI in EF to work with the data until a final UI could be designed. Some of the refactoring’s that I wanted to do but did not get time to do was to work on DI to uncouple the objects, break up objects to better separate responsibilities and other aspects of the SOLID principles, to change this application into a true object-oriented application. I also did not do any TDD work since this was a prototype. Once the final design is done, that can be started to lead the development of the final application.

As a final thought, I did try to use some things from the language where I could easily fit them in, like LINQ, and awaits for async, EF help with that, and data annotations. There is even more I want to do but time ran out. If there is more information about my design process, I am more than willing to discuss this in depth in an interview.