1. A) ID card swiper for an office in Siebel – This interface is encountered when one wants to get inside a locked office or conference center inside Siebel. Usually access to these rooms is restricted and permissions are given on the icard which is then used to swipe into the office. This card swiper interface is thus needed to insure that only those with proper permission get access into the rooms. The design of the card swiper is affected by its purpose: it is build into the doorknob of restricted doors so that people will not miss seeing it when trying to get it, it contains two LEDs at the top so that blinking red signals the door is locked and a green signals the door has been opened, and the icard can be swiped and read by a magnetic strip easily.

The horizontal swiping of the icard makes it easier for people to hold the door knob and open it faster at the green light. I think that the designer did a good job in using the LEDs on top of the card swiper to signal whether the door is open or not. This will help people who have hearing problems and can’t hear the lock being undone to know if the door has opened or not. Also, the colored light system prevents the user from having to check whether the door opened after every swipe or not (this would get really annoying if the swipes did not work). I also like how the designer of the system build the swiper vertically instead of horizontally. This is a practical use of space because if people had to swipe the cards horizontally they would not be able to have enough space. Also, having the swiper vertically makes it really easy to open the door in one swift movement. Another think that I liked about this interface is that the swiper is located right on top of the door knob. This is a really intuitive place to put the swiper because everyone will see it when they go to open the door and will realize why it is there. If the swiper was located anywhere else, it would not be that clear why the door is not opening by simply turning the door knob.

Although I like the idea of the designer using a swipe system to check for permissions, I feel that with swiping there can be a lot of errors of the swipe does not go through. I would redesign this so that there would be a scan system in place instead of a swipe system. This way, the user would only have to hold the card up to the door and let the interface check instead of having to do the work of swiping and not having the swipe go through. Another thing that I dislike about this design is that the way to position the card before swiping is not clear. There is no way to know which way the card has to be positioned in order to be read by the machine. I would propose to fix this design by putting a clearer engraving of how to position the card into the swiper. Right now there is a little engraving on top of the swiper but it is not intuitive at all. I would make a much larger drawing and put it next to the place where you have to swipe so that there will be more room for the engraving and the picture can be bigger and easier to understand. The last thing that I dislike about this interface is that despite having the red and green lights, if something goes wrong when the user swipes their card, there is no way of knowing what happened. I would redesign this so there would be a LED display that would say informative things like “sorry no permission given” if the user did not have permission or “flip card over and try again” if the user was
putting the card in the wrong way. This way, at least the most common swiper errors can easily be fixed and the user is not left to wonder exactly what went wrong.

B) Ford Focus lights – The lights for the Ford Focus are located on a panel to the left of the steering wheel. The context for using this interface is whenever it is required to have your lights on when driving; drivers of the Focus have to use this interface to turn on lights. Because the lights are needed only at specific times, I think that the designers of this interface thought it would be a good idea to put the lights off to the side so the interface doesn’t distract the driver when it is not needed.

I like how the design is simple to use to just turn on the light; there are three settings, off, dashboard on, and headlights on. So, it is pretty easy to turn on the lights for the car. I also like how there are pictures above each setting to show which button does what. This is good because it prevents users from becoming confused as to which state the lights are on when they move the buttons. This will prevent a lot of trial and error on the user’s part. Also, I like how this interface is off to the side. It is helpful not to have so many things next to the steering when driving so I think that this is a good thing in the Focus’ design.

A weakness that I see with this design is that right next to the light interface there is another interface that controls how bright the lights will be. This is a weakness because it requires two different switches to turn the lights onto full power. It is not exactly intuitive what the extra interface is for because it is just a scroll and there seems to be no connection between the two. However, if you use the original interface to turn up the light switch without turning the scrolling interface all the way up, the lights will not turn on. I feel that this can be redesigned so that there is only one button to turn on the lights and control the brightness of them. Perhaps instead of having three switches, the original interface could have two more for a dimmer and brighter headlight. Another thing that I noticed was a weakness about this design is that the beam headlight control is on the other side of the steering wheel. I thought that this was really not intuitive at all because all of the headlight features for the car should at least be in the same place. If I were to redesign this, I would put the beam control next to the buttons to turn on headlights of the car or would make it part of the button interface itself.

C) Sony Ericsson Mobile Phone – This interface is designed for any person that uses a mobile phone. The interface is encountered when users of the mobile phone want to make a call, check voicemail, send texts, or use any other feature of the phone. Because the audience of the mobile phone is huge and diverse, the interface for the phone has to be very intuitive and should be very easy to use for people of all different backgrounds.

A strength of this particular interface is that when a call is missed or the phone is charging, there is a voicemail/text a picture representing each one of
these different items that shows up on the front part of the phone. This is a really good design because the user of the phone will not have to flip it open to check if they have any missed calls or not, they can simply see this by glancing at the cover of their phone. This makes checking missed calls easier and can save time if the user is in a rush. Another good thing about this interface is that there are shortcut keys to easily check voicemail and turn off sound. This is good because it saves time for the user by not requiring the user to search through menu items to find these common tasks. Another good thing about this interface is that there is a centralized button that contains all the different options for the phone. Once this button is clicked on, it shows a screen with 9 different options that are differentiated by pictures. This is better than just listing out what the different options are because the pictures might appeal to more people and can help them find what they are looking for faster.

A weakness of this interface is that there is no intuitive button that ends a call or starts a call. Most phones either have a green or red symbol on the phone button that signals the start and stop of a call but this one does not. This makes it harder for the user to realize which button to press to start/end calls. I would change this design by simply coloring part of the button that starts a call green and the button that ends a call red. Another weakness of this design is that it doesn’t provide an easy button to access the call archive. In order to access the call archive, one has to go under the main menu button and then find the option for calls. I think that since this is a really common thing to look for, there should be a button on the phone for this shortcut. Also, another thing that I dislike about the interface of this phone is that in the call log, when one selects a call in order to see more information, the call is automatically placed. I think that this should be changed so that before the call is placed a question should show up asking if the user wanted the call to the placed. This will avoid people making unnecessary calls.

2. Interview 1: With this person, I discussed how productivity is affected when there are too many applications opened up on a laptop and the laptop freezes. The person thought that when there are too many tabs open in IE or Firefox, the “is not responding” state was really unhelpful because there was no way to know what exactly went wrong. They thought that a good idea would be to have more feedback on why the browsers sometimes freeze up and give this error. I also think that for laptops/desktops there could be a desktop gadget that shows which apps are currently open and how long they have been open for. If an app is open for a long period of time, the user can easily check to see if they still need it open or can close it so it does not take up space and cause the laptop to freeze up. Another problem that this person noticed was that when downloading files, if a certain format is not available on your computer there is little instruction on what to do to eventually get the file to open. They thought that it would be good if a script launched a website that would contain a sort of reference on while types of programs open up which types of files. I thought that this was a good idea because it is pretty annoying when you go to download items and find out that you don’t know how to get the file opened.
Interview 2: With this person, I discussed diagnosing hardware failure in desktops. They said that it was really frustrating to figure out what the problem is when hardware malfunctions because there is no clear interface that deals with this. For them, if there was a better system for detecting hardware failures, dealing with this problem would affect their productivity greatly. One possible way to fix this problem would be to have a more informative registry of hardware problems located somewhere on the desktop help functions.

Interview 3: With this person I discussed the toolbar in Windows. This person really did not like the grouping of the taskbar because they felt like when they had a lot of files open for a certain type of program, it would take too long to look for the file that they wanted to open. They thought it would be a better idea to have a folder open up with each of these files instead of the common list function that opens up in the taskbar. This person also did not like that in XP when you have items open in the taskbar there is no way of looking to see what the item was about. They really liked how in Vista you can just mouse over the item and a little screen shot opens up with more information about the program.

3. Correctness Log : 1 = correct try 0 = incorrect try
   Correct: 1 Time: 43
   Correct: 0 Time: 33
   Correct: 1 Time: 26
   Correct: 0 Time: 11
   Correct: 1 Time: 14
   Correct: 0 Time: 11
   Correct: 1 Time: 9
   Correct: 1 Time: 8
   Correct: 1 Time: 10
   Correct: 1 Time: 9
   Correct: 1 Time: 6
   Correct: 0 Time: 8
   Correct: 1 Time: 7
   Correct: 1 Time: 5
   Correct: 1 Time: 6
   Correct: 0 Time: 7
   Correct: 0 Time: 6
   Correct: 1 Time: 6
   Correct: 0 Time: 7
   Correct: 1 Time: 7
   Correct: 1 Time: 7

Please see attached excel sheet for graph and equation and Alphabet.java for the source code. The learning constant for this graph is .69. The average learning curve is about 0.5 so this person has a faster learning rate for this test. I noticed that the person took a really long time (respectively) for the first run of this
program. This is to be expected because this time accounts for getting used to the program and what it is asking for. After the first 6 trial, the results became faster and started to flatten out. This is again, to be expected because the person had by this time established a pattern in typing in the letters and could type them in without having to go through the alphabet each time.

4. a) \[MT = a + b \log (A/W + 1) = 548 + 420 \log 6 + 1 = 1727.09 \text{ ms}\]

b) Dynamic MT
\[= a + b + \log (A/W + 1) = 548 + 420 \log 14*4 / 14 + 1) = 1523.20\]

Static MT
\[= a + b + \log ( A/W + 1) = 548 + 420 \log 14*8 / 14 + 1 ) = 1879.36\]

50/50 = .5 * 1523.20 + .5*1879.36 = 1701.28 ms
75/25 = .75* 1523.20 + .25*1879.36 = 1612.24 ms
90/10 = .90* 1523.20 + .10*1879.36 = 1558.82 ms

Minimum choice time for the dynamic menu is when there is only chance that the dynamic menu will get selected. This is 1523.20 ms.

The probability split that would cause maximum choice time is when only choices from the static menu will get selected. This is 0/100.

c) Hick’s law does not take into account any sort of error terms. So, when applying Hick’s law to real world tasks, we cannot account for any sort of error or delay that might have occurred.