Driver Pass Project

Systems Design By: Christopher Harvestine

**Briefly summarize the DriverPass project. Who was the client? What type of system did they want you to design? What did you do particularly well?**

This project’s challenge was to create a program that helped students who are trying to obtain their drivers licence. The goal of the program was to train students. Our client DriverPass noticed that there are very few tools to train students to pass their drivers test. Given there is a demand for the service and there are very few competitors, DriverPass saw an opportunity for growth in that area. Our client DriverPass had some requirements for their training program. They wanted the student’s to be able to take online classes and have the ability to do practice tests. They wanted the student to have the ability to set up on the road practice driving to help with their real test. Our client DriverPass also wanted the ability for the student to access their data from anywhere, even offline through a download. DriverPass wanted access to all of the accounts in case someone forgot their password for example. They also wanted the ability to log any changes in the system and print out a log report in case there was a problem.

I think I designed the flow of the system pretty well, I was able to have a blueprint of the system in my mind along with the interactions. When I was designing each of my diagrams I tried to make them as accurate to the system as possible. Between the flow charts and the object chart I made, my hope was that a programmer could understand the system I had in mind and start coding.

**If you could choose one part of your work on these documents to revise, what would you pick? How would you improve it?**

Upon reviewing the design constraints I realized I had not included some of them in my final design. I would revise the design document for this reason. I did not design for the downloading of data, I did think about it but I wasn’t sure how to do it. I was worried about having two different data sets and didn’t want to create a bug and override progress that has already been made. I did give the administrator an editing tool, they could speak with a student if there was an unidentified bug and correct it. After correcting the bug the administrator was to log it and the team in charge of maintaining the program would fix the bug in a future update. In order to fix that I would need to do some research on how companies dual manage online profiles with multiple data sets.

**How did you interpret the user’s needs and implement them into your system design? Why is it so important to consider the user’s needs when designing?**

I interpreted the user’s needs in my design by reflecting on my own needs through attending online school. I found the needs to be very similar and therefore based a large part of my design on that idea. I also interpreted the user’s needs by putting myself in their shoes. If I were attending an online program like this, I would want security to be tight. I would want to be assured that my account isn’t going to be hacked and lose my credit card to someone. One of the reasons I created the administrator editing program was for the student. If a student were to lose progress through a glitch or a bug and the company could not fix it, they might quit and lose out on something good for them. The company also does lose out if the student quits but I thought more of the student in that instance. I think it is important to put yourself in the user’s shoes and make the program user friendly because they are the ones who the program is designed for. You of course want to stay within the design constraints laid out by the company, but it doesn’t hurt to add extra features for the customer.

**How do you approach designing software? What techniques or strategies would you use in the future to analyze and design a system?**

Since I have released a video game previously, I have had a decent amount of trials and tribulations in software design. I however would still consider myself an ameteur since I have not been thoroughly trained. This is why I decided to pursue a computer science degree, to add to and fine tune the skills I have. Since I was creating a video game by myself with little knowledge, I got very good at understanding and creating systems. I had to become good at thinking about what would destroy the system or mess up the program, as I would write code for 11 hours straight sometimes. I needed to be able to write large amounts of code and when I hit the debug button, it all had to work. It could be very hard to find bugs that aren’t system breaking, but they do make the program malfunction. This ability came from the fact that I didn’t plan out my process which this class has helped me understand. If I had planned out the process and did those charts, I wouldn’t have needed to code for 11 hours straight because the design already existed, I would just need to implement it. I was creating the design and the code at the same time, since I was constantly creating a complex and integrated system, I was afraid to stop because it could be hard to figure out where I was going or what I was thinking. I would just leave a comment in my code like //Left Off Here Was Trying To Connect This Code To This Object. The next time I picked up the program it would usually take me around an hour to figure out where I was going with my code, and what the relationship was between the objects. This is assuming I could retrace my line of thought, there were times I had to pick it up and just code not knowing where I was going. I am glad I did this as it fine tuned my programming skills, however, pre planning the design and fine tuning it with charts and design documents is definitely the way to go. In my future projects I will implement this. Pre-planning systems design is also important for team related tasks. When I made my game there was no one else. I did not have to communicate my ideas to anyone except myself. Software development is generally a team process, planning through documents and charts will help the team understand the system so they can communicate with each other as well as start coding once the design is complete.