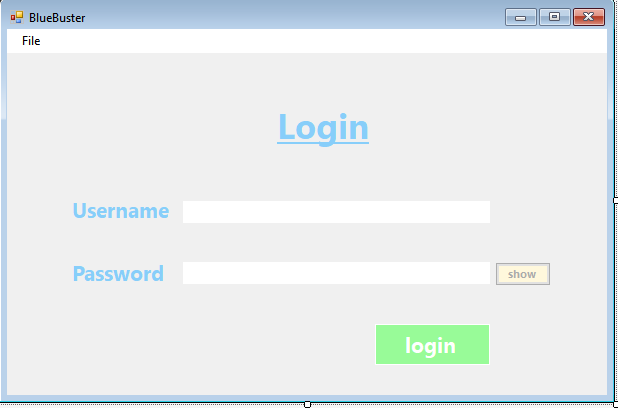
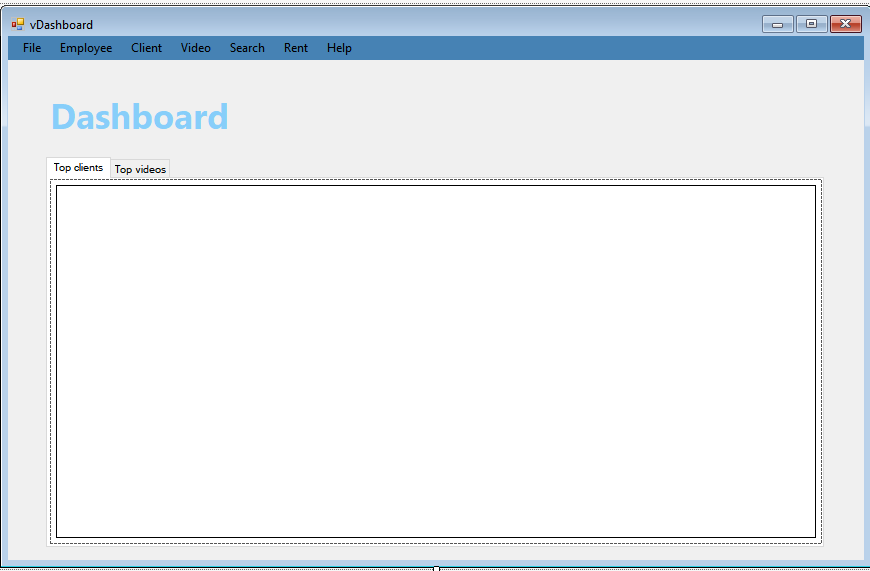
The video store project that was assigned to our team included a culmination of all we have learned throughout the course. To have been able to work and complete this project, one must understand some basic concepts of mySQL, vb.net and object oriented programming.

The end product works just like your typical video store software application. An employee or admin attempts to log into the chosen software, gets greeted with a dashboard displaying the top clients by number of rentals and top videos by most viewed. The majority of the options remain at the very top within a menu strip. These options include, logout, exit, create/update/delete clients, employees, videos, quick search, advance search and rentals. The options however are not available to all employees and have various restrictions according to the users access level. Admin, for example, remains at an access level of four, which can utilize the entire software as he/she sees fit with zero barriers. However, an access level of one is painfully limited to display information and the add feature exclusively. There are a total of four access levels.





Moreover, in this report I will go over a few points of importance. I will briefly touch on how the project was meticulously planned and how it eventually worked towards the end of production. I will provide a class OOP diagram as well as a database diagram, including various screen shots of the application. Also, I will lay out the problems we all fortunately encountered and how we inevitably solved them. Finally, I will go over the applications end result and what we could of done different in order to smooth the divided work load.

My group consisted of Nehal, Jose and me as active leader which was chosen by default, might I add, given to the large concurrent number of students. I came to realizing this project like I do all programming challenges since the end of block 2 - One layer at a time. This project like all others can become overwhelming if looking from its full representation. By chopping part of the application into pieces, it then becomes much more manageable. Also, by laying out a strong foundation in which others within the group can build upon, it not only made it much easier to comprehend but also got the ball rolling.

I instantly took initiative and created a folder that was shared between all members of the group. I then created folders within the project which contained some helpful information such as tasks, references, style and backup. Task was created with the intention of having the members of this group work on individually without duplicating or interfering with others planned assignments. I like to believe that it was used quite modestly and was helpful in providing information on current and finished sections of the application. The Style text was created to inform and remind members of this group of the current theme the application would overtake and provide clarity as to what it should resemble. Another very important text file I created was labeled abbreviations. This particular text file contained all proper vb.net abbreviations and was absolutely necessary for all members to view in case of confusion on naming conventions. I can remember during my first project where proper naming convention was never mentioned and therefore became an upscale problem throughout the creation of that application. I made certain my team did not encounter such an issue by providing this document and both members of the group followed the proper abbreviations perfectly.

Furthermore, once the pr-planned documentation was set, I then began creating the client, user and video class. All class properties were set to the appropriate data types of the database tables which was by default given to us. I then created various views such as login, new password, dashboard, and employee. Afterwards, I assigned different tasks to each member according to their preference. Jose worked on creating the video add/edit/delete view while Nehal chose to work on the client add/edit/delete view. I was taken by surprise by how quickly and efficiently both had completed their assigned work. Once all views were done, I tasked both members to work on their own part of the DBManager class functions associated with their section of the program in order to effectively start working on the back end part of the application. From there on out, all members worked independently. More information regarding member specific tasks will be shown towards the end of this report.

Nihal Discussion

We first discussed how to work on our project flow and then created a model for all the pages to show exactly how another pages looks. Then split the development component after discuss about how to create user-friendly software and add a few swages. When it was over, we had some problems with different structures and how to integrate layout using different frames. when it finish designing after that again group discussion regarding coding part. during coding part, I faced a lot of problems during the coding phase, but my team leader helped me fix my mistake every time after and before class.

Conclusion We did manage to finish the project. We tried to full fill all the requirements for coding and also a validations and some swage too.

JoseDiscussion

What happened throughout the project was that our team leader, George Nikou, went to vacation in California for Blizz-con. To be honest, it was extremely relaxing and it was very lucky for me to have been assigned to him as a team member. The main reason is because since he was in vacation for two weeks, I was also able to take a week off… except that I had to come to class for attendance only since I had no question for either Nima or George Boursiquot, in which case also went for a two-week long vacation.

Usually for every lab I encounter a form of road block at some point. However, during this project, it was a breeze compared to any other module or block. This was the case because my team leader, George, was very helpful by sharing his code for the ‘reader’ and ‘auto-populate a form’ part of his CRUD. By sharing his function/sub for those parts, I was able to study and learn how to do it myself without having to stop and figure out why x or y does not work because of a logical error.

Finally, during our project for Block 2, I was very lost throughout the whole project. Our team leader was using beans and foreign keys while I had no idea why we were using them. This project however, thanks to Frank, my team leader last block, I was able to understand very clearly why those two were used. In addition to that, during this project, we had someone who pulled their weight and actually wanted to contribute more instead of pushing the work to someone else part of the team.

As mentioned before in this report, I was able to avoid many problems and road-blocks. Thanks to our team leader, George, who split the whole project in three parts: Employee & Rent, Client & Login and Video & Search. Each team member did about 30% each since every case was worth 5% of the project.

Even though I was able to avoid many problems, I did encounter some here and there. For example, while trying to understand the reader part of this project for auto-population of a form, I tried to make my own function for about 1 hour and tried to understand how to do it with google and stack overflow. In the end though, I looked at how George did it for clients and saw that he was completely done with his CRUD by the 3rd day of the project. Thanks to that, I simply corrected what I was trying to do (which made no sense compared to what was supposed to do). In addition to that, another problem I had to encounter was the image load from a URL. The main issue I had was that I tried to make the function myself without knowing that George Boursiquot already gave us the function. The moment I found out he gave us the function though, I tried to add it to our auto-populate function. I am still not sure why it did no work but the way I fixed that was to store the URL in a textbox for the list form and then call the image load function in the same form. In which case it worked like a charm.

Finally, the other issue I encountered during this project was trying to fix one of our team member’s mistake for changing the database by adding a foreign key and our team leader not making it Auto Increment. That was not the only issue though, the status was not working for us meanwhile for the member who made the code, it was working just fine. We figured it out by looking at the video Bean and found out that for some reason, the status in the constructor was a Boolean. We know that Booleans only take 1 or 0. But when submitting the form, for some reason, it was giving -1 for the status and it was preventing us from adding the filled form to the database. So in the end, the easy fix was to change the Boolean to a String.

Firstly, one of the first biggest issues we encountered within just the first week of the project was rather frustrating and in fact a huge time waste. Apparently, saving our work while having different versions of visual studio can cause an error which leads the projects designer to be recreated and when finally launched, render the program completely unusable. This problem had me quite frustrated for a time seeing as I had to re-paste all our content into a new project and force us to change our approach in how we work together. The new main project folder was used by only me while Jose and Nehal had their own main folder they worked from. We all unfortunately had to work this way in order to avoid possible corruption of occurring again. I also informed all members to code on Visual Studio 2017 as this was the version we were initially informed to download. Whether this would have fixed the problem, I am not certain, but it certainly eliminated one potential error.

Secondly, the types of errors I witnessed within our project were quite minimal and was rather easy to fix. I specifically had gotten some errors regarding property types not matching with the database table types. I had also seen some members encounter problems with adding a new entity due to not having too many or to few parameters in their class constructors when creating an object of that particular class. Like I said however, these mistakes did not slow us down and was rather simple to diagnose.

Lastly, not much of an error so to speak but rather a confusion I myself had a somewhat difficult time trying to differentiate the differences between using a data table, data set and data reader. At first, these all seemed very similar and proved to be beyond confusing. However with time, I slowly was able to identify which was more ideal to use for the current situation.

In conclusion, the video store project application was an overall success. All of it’s features were fully functional and behaved the way expected. I solely believe that all members of this group learned a great deal and that the process of creating this application allowed us to gain confidence in our ability and function better within a working team environment. Some of the minor issues regarding the application are linked to mostly having incorrect validation or no validation at all. Certain areas of the application could have been more user friendly as well. The quick search for example, which primarily took a single input could of taken one parameter field as oppose to having two mandatory fields with unnecessary validation. These were minor issues, however, issues that would pose no problem and could of easily been fixed given had we been given additional time.

