Comprehensive Assessment

Systems Development

Jonnie Klein Quezada
jonniequezada@gmail.com
github.com/JKleinne

Fall 2018

Table of Contents

Introduction

Prior to taking this course, students were exposed, although minimally, to the environment of a developer. However, development is closely tied to designing. Ofttimes, although we were taught to first plan and think algorithmically before writing any code, many of us have skipped that important phase in development. As such, programs written were, most often than not, not optimized and inefficient. We cannot afford that flaw in the business world and thus, we must put an emphasis on project planning and system design. In a program whose goal is to train students to succeed in the industry, practical experience is a luxury that each must have. Therefore, we have been placed in teams where each members' skills, to the best of possible combinations, complement one another. I was fortunate enough to be placed in a team whose primary expertise was in redacting documentation, which is indubitably my Achilles' heel. It helped that we somewhat knew each other. Otherwise, we would have needed an ice breaker activity so that we could identify each other's strength and weaknesses, and how we can possibly cover for them and mesh in a way that maximizes our true potential. One of the challenges I face personally was to adjust myself in not overlooking the designing page. In my previous experiences, notably with coinsave.io, the team had a dedicated designer and thus, the developers could focus on their own work. However, during this course, the most invaluable lesson I have learned is how crucial important project planning is, more than I could ever imagine. There exists a language barrier between developer, clients, and designers. Carefully and meticulously planned projects' goal is to shatter, or at least blur that barrier through the use of UML diagrams, user stories, user cases, etc. Eventually, after our planning, it was time to move on to the

implementation of the prototype. The prototype itself was vastly different from our initial designs, mainly due to the lack of time left, needed for the system completion. As such, due to the time restrictions, we had to choose only a select few features that we would implement. Nearing the end of the semester, many of us had final exams coming up. And so, we had to learn time management skills to adequately balance the implementation with studying. We treaded on a thin rope but that's the life a student lives.

Project Plan

Phase 1: *Deliverable 1*

Now placed in a team, it it now our responsibility to organize ourselves. A group of individuals cannot succeed without knowing each's role. Although we all have been somewhat acquainted during the course of our stay in the program, we frankly do not know each other enough to cover for each other's weaknesses. We wanted to base ourselves on Kant's Enlightenment. We want everyone to have their own voice and freely critique one another. As such, our group functioned less like a swarm of insects, relying on the Queen for direction and commands; no, we were independent enough to know what was best for the group. Additionally, we had to decide on version controlling our project. And so, we decided to use Git, a tool I am fluent with from my time in coinsave.io. Finding a client was indubitably the less daunting task out of the others. We had multiple options presented to us, due to our network with other organizations, companies, etc. Earlier I mentioned we treaded on a thin line between implementation and studying, without the project plan that we have redacted, that line would not have existed and instead we would have to behave how entangled quantum particles would: every possibilities exist simultaneously prior to observation but must settle on one reality once observed. We would have to choose between passing this course or passing our other courses. Fortunately, we were taught at a very early stage in the semester on how to properly organize our project. Professor Sadegh Ghaderpanah taught me the importance of employing a divide-and-conquer approach in programming. A complex task can be broken up into multiple smaller and simpler tasks. This would prevent the developer from seeing nothing but darkness but would lead

him to see light at the end of the tunnel. Similarly, by creating a Gantt chart for our first deliverable, we sliced the daunting task we were facing into manageable pieces.

I would say my biggest flaw is in not seeing the importance of designing is in the development of a system. I have always lead myself to focus on my task and refused to see the bigger scheme of things. My arrogance had left me vulnerable in an industry that demanded near perfection. Consequently, I had trouble adjusting to a project in which I would be participating in the most crucial phase of systems development: *the design.* Eventually, as the semester passed by, I am now much more inclined to spend more than the majority of the time, in the scope of developing a system, in the designing than the implementation. Before, I would have skipped the design process and would instead design as I implement which, looking back today, is a rather fallacious approach. I have grown from a one-dimensional person into someone that now understands the multiple facets required to successfully manage and deploy a project.

As a team of computer science students, we were not fortunate enough to have learned how to use project planning tools. As such, we had trouble creating the Gantt chart at first, but eventually completed it. As presented earlier, we decided to use Git for versioning, but that is another part in which the team struggled. I did my best in giving them a crash-course on Git using only the basic commands. It ended up paying dividends as the team can now properly create a development branch that mirrors *master* now know how to commit and push their changes.

Apart from Git, which I learned on my free time during the first semester, there were not many technical skills I had to use for the sake of the first phase of the project. There were, however, concepts and philosophies that I brought with me from other courses. The *divide-and-*

conquer approach that I had learned from Professor Sadegh Ghaderpanah during Programming I was most useful during the creation of the Gantt chart

Client and Business Domain

Phase 2: *Deliverable 2*

Shortly after deciding that we would choose the Vanier Library as our client, we settled on our first meeting with the client. The client representative turned out to be Haritos Kavallos, an individual I have acquainted with during my time with the Vanier Robotics Club. As such, I felt comfortable with him being the representative because I knew, for a fact, that he would be someone the team will be glad to work with. However, this is a business meeting and not a personal reunion; we prepared our questions that we would ask him in advance. We then gathered more information through our meetings with the librarians and potential administrators of our system. We learned that they did not have any reservation system of any sort, in place. They do have a walk-in same-day (same-hour) booking which again, is not a reservation system.

Personally, I did not feel like there were any challenges I faced on a personal level during this phase of the development. If there were anything I felt that was off-putting, it would be my urge to complete the login system which we all knew needed to be done eventually. However, I eventually saw folly in that as Professor Lebensold stressed the importance of design over implementation. I am most thankful for that lesson as I believe it would have been my death in the industry, had the attitude of skipping design persisted.

The team fared well during the phase and nothing terrible could be said. There were instances in which we could not be all present for a certain meeting with the client or with each other, but we handled it well by covering for one another.

Again, this is far from the implementation phase, so consequently, there were not many technical skills we had to rely upon during this phase. However, our course on Business exposed us to how many different business operates. Thus, we were not particularly struggling to match the client's pace as he talked about the business problem, business domain, etc.

Use Cases and UML Diagrams

Phase 3: *Deliverable 3*

For this deliverable, we had to meet with the client and discuss the current system in place. Fortunately, we were taught about UML diagrams and Use Cases during this course, which tremendously helped us. The UML diagrams that we created with the intent of providing aid in visualizing the design of the system gave us an overview of which we can present to the client with much better understanding and simplicity. Employing Use Cases, we described the interactions between the user and the system to achieve a goal. In addition, we also redacted class diagrams and state chart diagrams to further aid in visualizing the design of the system. It can be said that the majority of the people are visual learners. Thus, it is a tremendous aid to provide a visual representation of the system in order to further extract information from it.

I believe this phase was one of the most difficult one I have faced during the course of the project. Being someone with little to no experience in systems design, I found it challenging to produce quality UML diagrams and Use Cases. However, in every struggle there exists a lesson to be learned. I can now confidently say that I will be much more prepared the next time I will need to produce a visual representation of the design of any given system.

I struggled on a personal level and I believe the team did so as well. We were able to rely on at least one person to provide us with their take since it is their expertise. However, none of us had any experience in producing diagrams and so, we struggled.

We had to learn to use Microsoft Visio, which did not really necessitate a youtube session to learn how to use. We came into this phase with no luggages in hand, but left this phase with the proper knowledge on visualization.

User Stories

Phase 4: *Deliverable 4*

I thoroughly enjoyed this phase, more than any of the other previous phases. It was a mark that implementation was approaching, something that I am still deeply looking forward to, despite my lessons learned on the importance of design. And so, we had a meeting with the client in order to gather requirements of the to-be Vanier College Reservation System. Because of my experience working in an Agile team, I was fortunate enough to be exposed to user stories, which is the description of features the user would like to have in the system, written in the perspective of the user. However, there was more to what comprise the users than just an administrator. We also discussed with the librarians and they helped us redacting user stories which would present the features they would want in the system to our development team.As such, we were re-wrote the 30 user stories that we had discussed with the client on flashcards, with their tests on the back.

I felt this was the phase I enjoyed most compared to the ones prior to this. Furthermore, enjoyment does not necessarily imply that there are not any complications in existence. Oddly enough, I have not faced any challenges during this phase apart from the pain I suffered in my wrists caused by the manual writing of 30 User Stories, implying my lack of writing using pen and paper ever since my admittance to this program.

The same thing can be said for the team. Although there were complications such as Professor Lebensold telling us that our User Stories were too vague and similar, none was

particularly alarming. We managed to modify the user stories, some by completely re-writing them, some by combining the ones similar, etc.

I don't like working during school terms as it will most likely conflict with education and personal projects. As such, I took a break from my job at coinsave.io. However, there are many skills I learned from that in which are most useful to this course. Our development team was closely tied to our client, in fact, the client is itself part of the company as CFO. Using JIRA by Atlassian, they would post User Stories that the dev team would need to implement by a set date. Using that knowledge, I was able to help my team.

Prototype and Client Comments

Phase 5: *Deliverable 5*

Now that we have gathered the requirements for the system, the goal of the fifth phase is producing a sketch of what the prototype would be like. We first had to draw out a sketch on pen and paper and discussed with our client. There were a few minor changes to be made, but the client was particularly pleased with our approach. Afterwards, we moved on to a computer sketch of the design of the system. We used a website builder to produce the sketch and again, the client was surprisingly pleased with the general look and feel of the system. There were again, a few minor changes to be had, such as changing a title, or using a different color scheme, but he was a firm believer in a simplistic design, which is what we ultimately tried to go for.

I consider myself an artist, having played the piano ever since I was young. However, I needed to be a different kind of artist for the sake of this phase. My drawing and sketching skills are not particularly top-notch, in fact, one could argue that an 8 year old has better artistic skills than me in that regard. Thus, I have struggled sketching out the designs and my lack of creativity in art did not help either.

I do not think the team faced any major hurdles during this phase. I could argue that they enjoyed this phase more than I did, being amazing artists. I personally struggled during this phase, but they covered for that weakness with no complaints. There was, however, a complication in which we were not able to meet the client in time to finish the deliverable. Thankfully, we contacted Professor Lebensold and he agreed to delay the submission date of the deliverable, allowing us to meet with the client and gather his input on the matter.

Database Design

Phase 6: *Deliverable 6*

The implementation phase was quickly approaching. I had gathered myself, thinking the time had almost come. All of the lessons that I will have learned in this course and others will finally be showcased. Conversely, before implementing the prototype, we must first properly design our database. Just as I have learned during this course, design comes before implementation. And so, using Visio, the same software used in producing Gantt Charts, UML Diagrams, etc. we decided to take on this important phase. I was mainly concerned with how we would internally structure our database, would we have separate tables for regular users and admins? Or would we just add a permissions column in the users table noting that a specific user is an admin? We then had to decide on the data types for each attributes and we were not particularly sure if we would store studentId as a seven-digit integer or as a varchar.

My main concern reading the instructions for this deliverable was the use of indexes. Although we had seen this during class, I was not so sure how it would be of use for our prototype. Otherwise, everything else went fine after thorough discussions.

We initially had trouble in deciding which types our attributes would take. We first wanted to store studentId as an integer and we made other decisions that, looking back now, seemed rather silly. Eventually, our lack of direction gave birth to a philosophy we employed for the sake of the database design. We decided that any attribute that wont't necessitate any mathematical calculation shall be stored as a string for simplicity. As such, studentId was then stored as a string.

Professor Chebbine's lessons during the Database I course were of tremendous value. We managed to remain consistent in our design thanks to his advice of the advantages in using foreign keys and related tables.

Implementation and Client Comments

Phase 7: *Deliverable 7*

Finally, the moment I had been waiting for had arrived. It was finally time to use my expertise in practice. For simplicity, we divided the group into two sub-teams: *frontend and backend*. Because of my expertise, I was acting as the Full-Stack Developer, overseeing both teams. We decided to use Full-Stack Javascript for this implementation. In other words, we will be using Javascript to code the UI and Javascript to code the server. The reason behind this decision is that Having one language will allow your team to work much more closely since both are related and will facilitate debugging since it is restricted to just one language. Also, Node.js, which is what we use for the backend, fits our needs perfectly. If we were to use Ruby on Rails or PHP, we will taking a massive hit in performance since Node.js is extremely fast in handling a multitude of requests. Our prototype should be able to search, add, delete and update items or group of items in the database. It should also be able to produce a report of the database contents. We managed to fulfill all of those requirements by having our analytics page, calendar and user tables to retrieve data from the database. The administrator will also be able to add new sub-admins which fulfills the add requirement, in addition to students able to register. Administrators will also be able to delete students that are no longer in the College, although the administrator must enter his token due to the restriction we placed in ensuring that the administrator must first verify via Two-Factor Authentication before deleting. This will prevent any accidental deletions.

Additionally, administrators are also able to update student data and print out a summary sheet of all the students in the database.

Out of all the other phases, even more than the user stories, this is the phase I enjoyed the most. There were no challenges on a personal level that I have faced because this is my expertise, my passion and my hobby. I did not feel this as work, but rather as a source of entertainment.

I believe as a team, we initially struggled to be on the same phase, mainly because we were not on the same level of expertise in coding. However, we overcame this by having them focus on coding the simpler parts of the prototype first then proceeding to help with the more complicated ones.

Evidently, I used skills I have learned from my Internet I and Internet II courses for basic html and basic Javascript. However, that would not have been enough. I used my work experience from coinsave.io, where I learned about the Asynchronous nature of Javascript, React.js and Node.js. Using that knowledge, I managed to give a crash-course, although a very quick overview, to my team which eventually helped them overcome the fear barrier in learning a new language / framework.

Teamwork

I am quite used to working in a team and so I did not have much adjustments to make during our term project. I know for a fact that it is better to have a team of experts in their respective fields than have a team of jack-of-all-trades. Fortunately, I was placed in such a team. Although documentation was my inherent weakness, my team covered up for me and although the implementation is where the team struggled initially, I took on the weight and helped the team stand back up. There were many instances in which we were able to cover for each other's weaknesses and I believe that that is the mark of a successful team. There were no complaints whenever one could not attend a meeting, everyone was understanding that there are circumstances which simply spun and cannot be escaped, such as an unavoidable exam. A philosophy that I developed during my time in Air Cadets is that I would rather lead a team of friends than lead a team of strangers. Thus, we all eventually developed a bond, a friendship that will persist beyond the scope of this course. We always tried, in the best of our abilities, to forecast possible complications and adjust accordingly. As such, during our final meeting, our client revealed to us that he was particularly impressed by our contingency planning. Our teamwork also lead to the seamless transition of switching leadership. Because we all knew each, at this point, each other's strengths and weaknesses, we were able to trust the leader in handling contact with the client, producing reports, etc. I appreciate that none of the team members had any hidden agendas hidden behind our backs, something that is inherently frequent in many team projects. In the end, I believe this team was successful not only due to our staggered expertise encompassing many different fields, but also due to our trust in each other.

Conclusion

This course was meant to teach one the steps taken in the creation of a computer system. It was also meant to encourage students in working as a team and learn how a successful team should function. However, the greatest lesson that I have perhaps learned this semester, and I wish I had learned earlier, is the importance of adequate, meticulous and design is to the quality of the system. Without proper design, the system will hold many vulnerabilities that could mean the end of the system. It will be far from perfect and will be far from what the client desires. I believe the added experience in teamwork will help me achieve my goal in eventually participating in Neural Network research with a team of experts. The abundance of documentation that we had to redact during the course of this class will also help me in writing thorough research papers in the future.

Works Cited

Quezada, Jonnie. (n.d.). Deliverable #1.

Quezada, Jonnie. (n.d.). Deliverable #2.

Quezada, Jonnie. (n.d.). Deliverable #3.

Quezada, Jonnie. (n.d.). Deliverable #4.

Quezada, Jonnie. (n.d.). Deliverable #5.

Quezada, Jonnie. (n.d.). Deliverable #6.

Quezada, Jonnie. (n.d.). Deliverable #7.