Final Report

Marc Daaboul

Development Project I

Christopher Elliot

17 Dec 2022

**Table of Contents**

[Introduction 3](#__RefHeading___Toc422568_1244017957)

[Course Goals and The Development Project 4](#__RefHeading___Toc3882_1557039811)

[Purpose of Development Project I 4](#__RefHeading___Toc3884_1557039811)

[Agile Scrum Approach analysis 5](#__RefHeading___Toc3886_1557039811)

[Sprint Review & Sprint Planning Meetings 5](#__RefHeading___Toc1148_1244017957)

[Daily Scrums 5](#__RefHeading___Toc1148_12440179571)

[Sprints 6](#__RefHeading___Toc1148_12440179572)

[Retrospective Meeting 6](#__RefHeading___Toc1148_12440179573)

[Role of APM 6](#__RefHeading___Toc1148_12440179574)

[Team Experience With Agile Scrum 6](#__RefHeading___Toc3888_1557039811)

[Design and Build – Analysis 7](#__RefHeading___Toc3890_1557039811)

[Budget and Estimates 7](#__RefHeading___Toc3892_1557039811)

[Budget and Plan 7](#__RefHeading___Toc1159_1244017957)

[Actual 7](#__RefHeading___Toc1161_1244017957)

[Analyze 7](#__RefHeading___Toc1163_1244017957)

[Design 9](#__RefHeading___Toc3894_1557039811)

[Plan 9](#__RefHeading___Toc1165_1244017957)

[Actual 10](#__RefHeading___Toc1167_1244017957)

[Analyze 10](#__RefHeading___Toc1169_1244017957)

[Refactoring 10](#__RefHeading___Toc3896_1557039811)

[DevOps 11](#__RefHeading___Toc3898_1557039811)

[Test and Validation Analysis 12](#__RefHeading___Toc3900_1557039811)

[Black Box and System Testing 12](#__RefHeading___Toc3902_1557039811)

[Functional Tests 12](#__RefHeading___Toc421638_1244017957)

[Defects Table 12](#__RefHeading___Toc421640_1244017957)

[Usability Tests 12](#__RefHeading___Toc421642_1244017957)

[Exploratory tests 12](#__RefHeading___Toc421644_1244017957)

[Conclusion 13](#__RefHeading___Toc421646_1244017957)

[White Box/Automated Tests Cases 13](#__RefHeading___Toc3904_1557039811)

[Conclusion 14](#__RefHeading___Toc3906_1557039811)

# Introduction

This is a report on how the first couple of sprints went in the Development Project I course. I believe that this report will take me around 9 hours to make considering what the other students have told me how much they did.

# Course Goals and The Development Project

### Purpose of Development Project I

The development project I course is supposed to cover 2 competencies: Developing a web application and collaborate while doing this. The following 2 tables cover what competency elements each activity cover.

|  |  |
| --- | --- |
| **Develop Transactional Web Applications** | |
| **Activity** | **Competency Element** |
| At the start, we had 2 clients present their project to us and we had to take notes on each one. From there we decided as a team on which project we would prefer to do.  Once it has been decided we had multiple meetings with the client of our choice to clarify any uncertainties. | Analyze the application development project. |
| This was mostly given to us since the only thing our team really knew was dotnet and Visual Studio. We also used the nuget packages that we learned about from our other classes. | Prepare the computer development environment. |
| Again based on what we learned from the other classes, we chose to use T-SQL since it integrates easily with the rest of the system.  Setup was mostly done by 1 teammate but input from the rest of us was then added. | Prepare the database. |
| We had assigned tasks to design and create the pages. For me I have edited the heritage logo to match the rest of our theme and created the sets page. Also created a base for the side bar and for the grid that will be used for primarily navigation. | Program the Web interface.  Program the client-side application logic. |
| I did the logic for the test cases functionality. | Program the server-side application logic. |
| We made sure to use pull requests as a team to make sure all the code has been reviewed before getting pushed to our dev branch.  Our dev branch was then getting more tests done on it. | Control the quality of the application. |
| We had to follow the run book and deploy our projects to dev and test servers | Participate in the deployment of the application on a Web host. |
| We had to create documentation for people coming from outside like from our readme and for people already inside the team, like our code standards.  I have helped in creating the run book and the code standards document. | Produce the documentation. |

|  |  |
| --- | --- |
| **Collaborate on the design of applications.** | |
| 1. **Activity** | 1. **Competency Element** |
| 1. We would have meetings with the client and make meeting minutes to make sure all the info was noted down. We would then send those to the client and make sure that all our features are accurate to what the client wants | 1. Participate in the development of the functional specifications. |
| 1. We had a meeting to discuss all the research we had all done individually about which technologies we should use. On my part I presented the options for a compiling service and a text editor on our website. | 1. Participate in the overall design of the application |
| 1. We have all made checks to make sure the database and what the client needs are clear, but there are still some changes that are needed to be done. | 1. Develop the detailed design |
| 1. We have had meetings with our client to get the workflow of the system sorted as well as our general design. We placed them all in a moqups. | 1. Produce design documents |

The course first starts off with getting our teams in the flow of scrum by making us decide on a project. Later we plan the project of our choice by going to meetings with our client to make sure we there are as little uncertainties as possible. Once that is done we started our development cycle and worked on separating tasks and working on them. I believe the development cycle was started too late since we had no time to plan for our sprints other than during it and since we did not have that long to work on the project in the first place. It made us feel rushed to finish the planning and start coding. I think one way to fight this is by placing a little time in between the 2 development sprints to plan for the second one. That way we would have all our stuff sorted more easily. On my side, the technical architecture we have decided on as a team was not an issue for me.

### Agile Scrum Approach analysis

#### Sprint Review & Sprint Planning Meetings

This is the point where we would plan, then review the sprint with the client. I think this went fairly well at the end of it, we had learned a lot from the first couple of meetings with our client. We were able to better understand our clients more and what they would generally need. Like when at the start, he asked us that he ‘didn’t care’ about styling and we took that literally. This ended up costing us some extra time to get priorities sorted.

#### Daily Scrums

Every start of class and end of the last class of the week we would have a scrum meeting to discuss what is and what’s left to be done. I think those went fairly well, but I think we ended up going off course multiple times during the meeting and that cost us a lot of time for those not even concerned with the issue.

#### Sprints

The sprints in general \*again\* went pretty well. We were pretty consistently busy enough to fill the sprint with work. One problem with it was when we did run out of tasks, we ended up scrambling a bit, since a lot of the tasks we had were either ‘full commit or no commit’. This ended up causing some people to switch tasks a lot or doing some lower priority ones. Regarding the sprint length, we chose 2 weeks as a class and I was pretty happy with the length. The only issue with this is that we did not have enough time to plan our sprint and actually do it. It would be nicer if we had a planning sprint in between the sprints we had to plan the following sprint out. I believe it would’ve went much smoother if we did.

#### Retrospective Meeting

The retrospective meeting we had as a group went as expected. It was great to listen to the point of views of the other students and brought up some very good points that I now stand by. One small thing I didn’t like was how long it was. It was a little bit difficult to stay focused the whole time and keep participating since there was some repetition in what was said.

#### Role of APM

I don’t think we had that role defined correctly as a team. The APM really just started and ended meetings and sometimes sent emails unlike how we had it in systems maintenance, where APM’s assigned tasks sent emails and made sure everyone was completing them. This role helped me the most when it came to really understanding the project. I was forced to actually see the top level view of the system we are building and everything that was needed to be done. This made writing documentation, talking to our client and assigning tasks easier.

### Team Experience With Agile Scrum

The work was divided based on what people liked doing. Some people wanted to work on the database and others on the design of the site. This worked out very well at the start but over time we started noticing the ‘bus factor’. Some decisions weren’t optimal for someone working on another section and that ended up costing us some delays. I think one change we would need to do for next semester is make sure there is always at least 2 people working when it comes to making decisions that would end up affecting others.

# Design and Build – Analysis

### Budget and Estimates

#### Budget and Plan

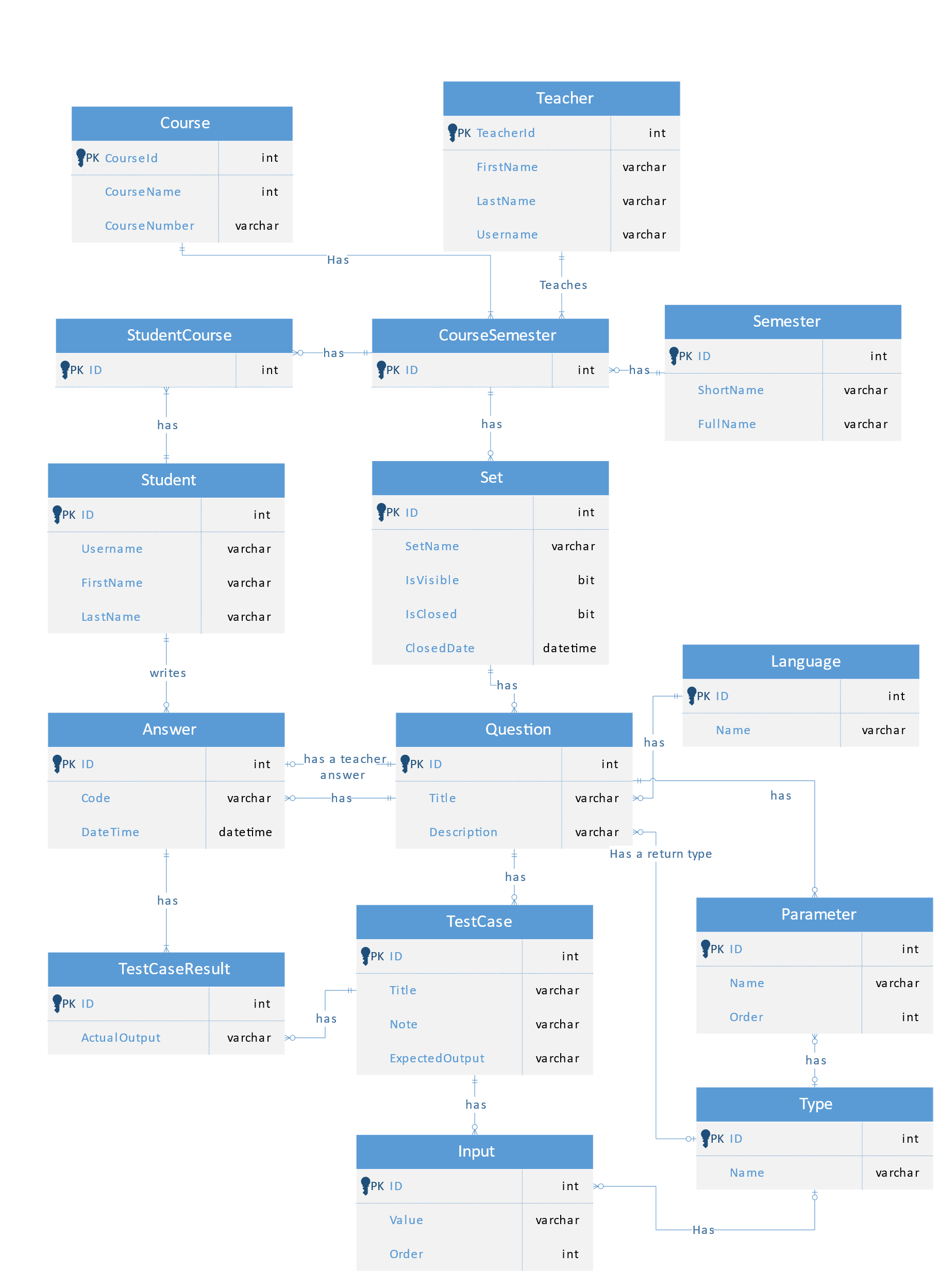
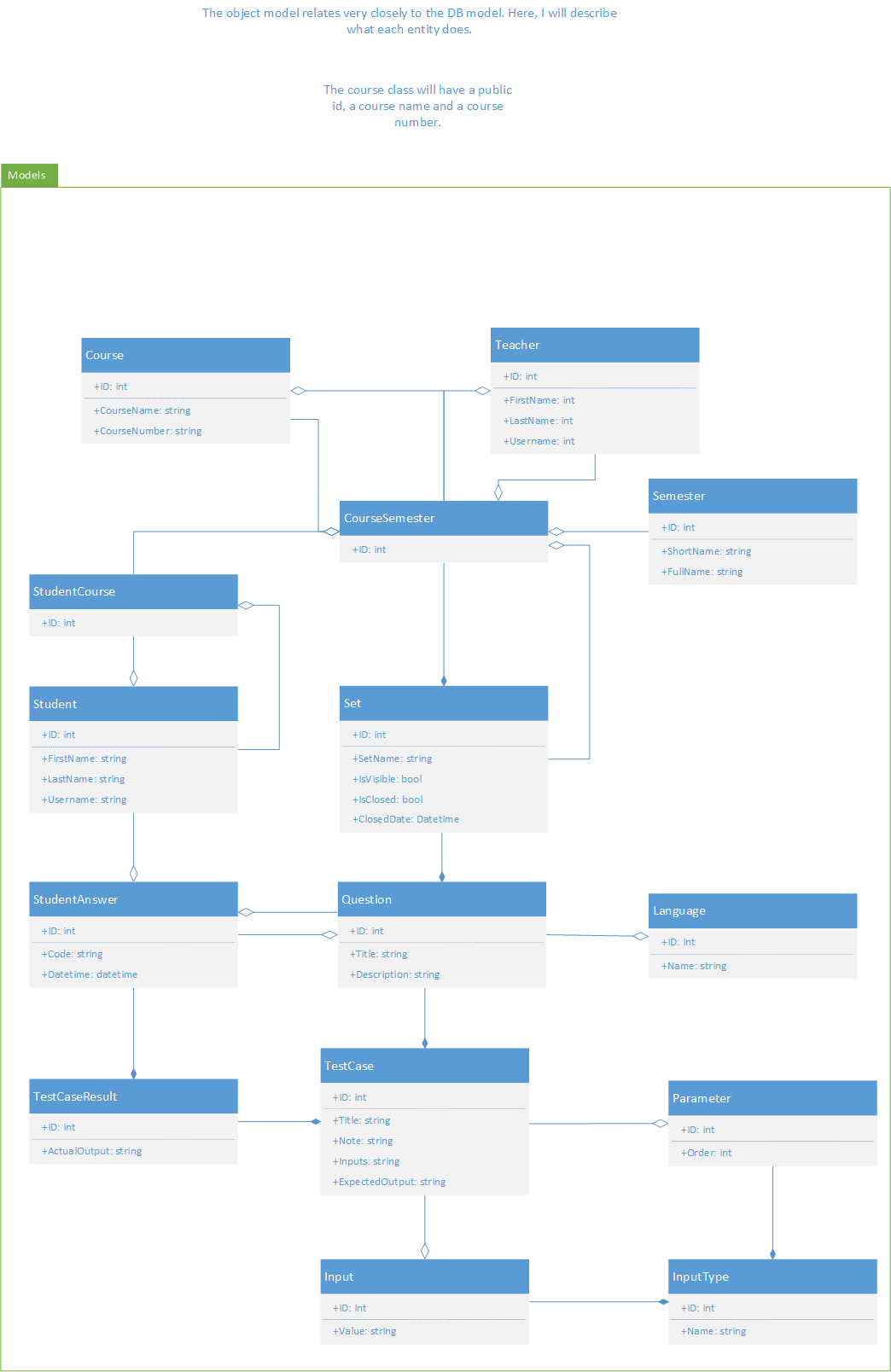
Since we had 6 hours of class and 3 hours of homework time to work on sprints per week and every sprint was 2 weeks worth. We each had 18 hours assigned to us every sprint. That gives us 108 hours in total worth of work every 2 weeks considering we are a team of 6.

#### Actual

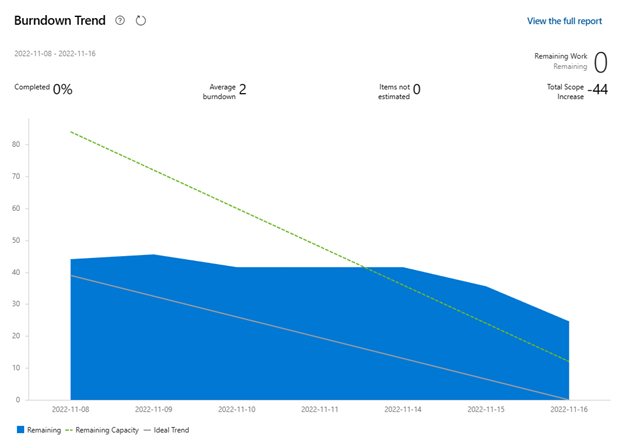
In general, we were usually on budget when it came to the amount of hours spent. Sometimes it went over, but it would mostly go under since we wouldn’t have time due to our other classes.

Generally my tasks usually hit either on the 18 hour mark or went under consistently, but the tasks changed a lot in the middle of the sprint so it is hard to tell.

#### Analyze



So far, I believe our design plan accurately reflects what we have made in our original design. We have noticed multiple flaws in them and will attempt to fix them next semester. I think I would assign more people to review this since not everyone has agreed on the current design.

Figure 1: Development Sprint I burndown chart

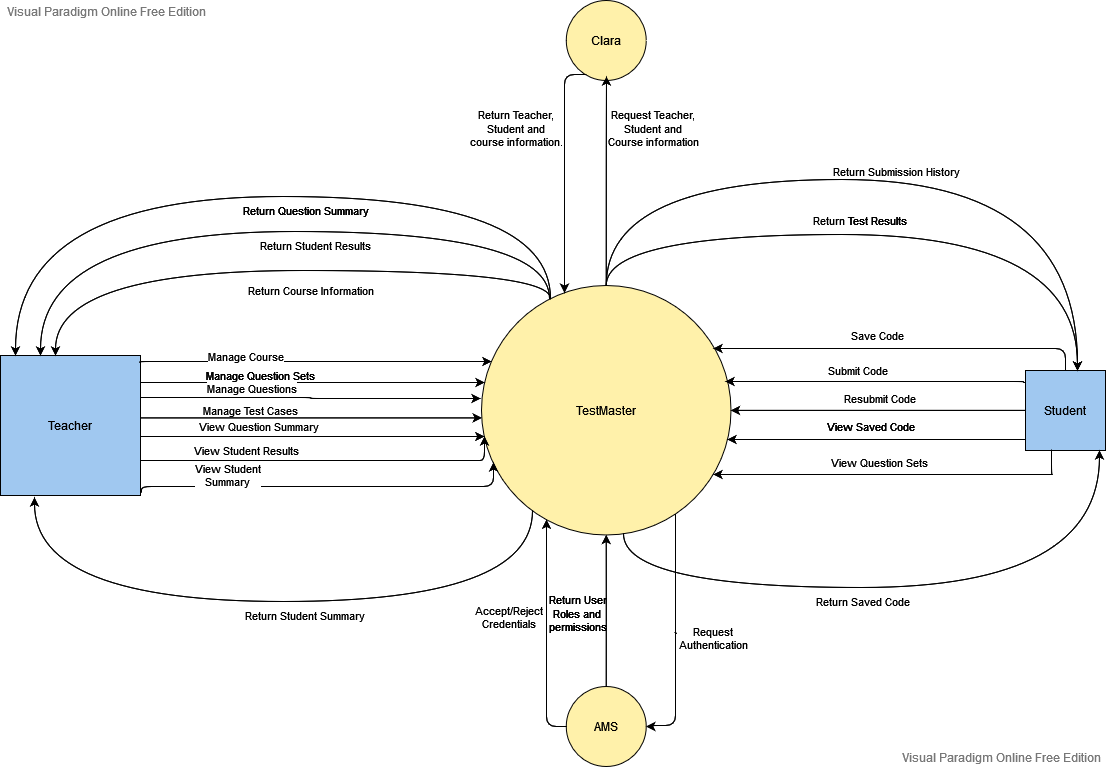
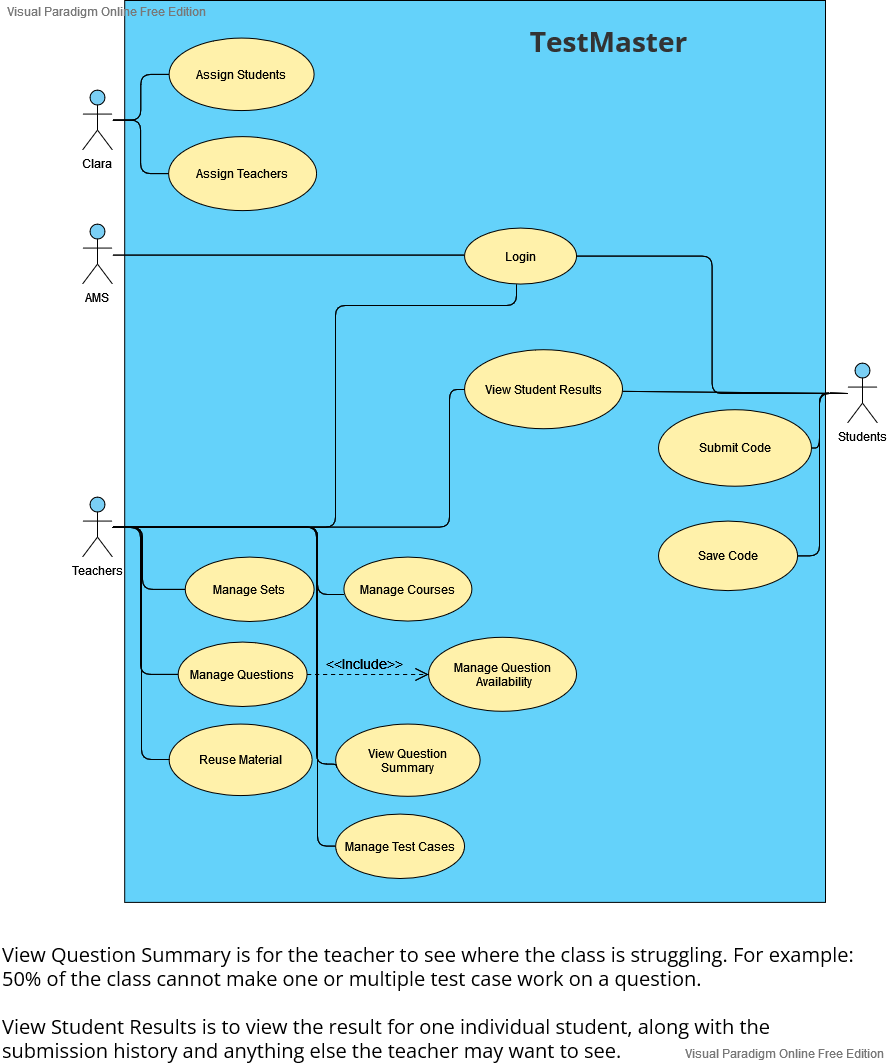
This was our first development sprint and as you can see, we really only completed the tasks towards the end. That is because this was including a lot of setup and we ended p waiting a lot for each other to complete some tasks so we could get started on or own.

Figure 2: Development Sprint II burndown chart

This one is the second development sprint. There is a spike at the start of sprint since we weren’t able to plan before the sprint. But this one the work goes down more smoothly which I think is very good on our part.

### Design

#### Plan



#### Actual

The system ended being very similar to what we have planned it to be. One difference is that we have a separate system altogether for code compiling that TestMaster interfaces with, but it shouldn’t be different for the rest of the parts.

#### Analyze

The original was good enough for the system to be based on. I would have a separate circle named compiling service that takes care of compiling for TestMaster.

### Refactoring

Refactoring is when you attempt to make changes (and most likely with intent to improve!) to the internal structure of code. It is not adding new features, it’s just making thing (hopefully) better. This process could bring benefits like: maintainability, performance, security and scalability.

One improvement that can be done to the database is getting rid of unnecessary tables, right now we have the ‘Input’ table which ends up adding extra work for the programmers for close to zero gain.

Another system we will have to refactor is the code compiling system. It is built in node.js and might complicate deploying our web app (I believe someone got it to work though!). This is not confirmed, but I think maybe a refactor to a different language like C# might be ideal. Of course this will have to be accepted by the team first.

# DevOps

DevOps is just today’s way of bringing the latest and greatest of your application as soon as possible. It usually follows these steps (according to IBM):

1. Planning
   1. see which tasks to move from the backlog to the sprint.
2. Development
   1. complete those tasks. Make sure you test!
3. Integration
   1. (CI/CD) have a ‘master’ code base that has all the changes everyone did. At this stage you could run automated tests as well.
4. Deployment
   1. Runtime tests are done to check the quality and stability of the application. Then release those changes.
5. Operations:
   1. Monitor the release and make sure nothing is going wrong while it is live.
6. Learning:
   1. Gather feedback from the end users and take that into account when going back to the planning phase.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Clients get updates faster | May end up causing the developers to rush a feature since the time limit is not as big. |
| Clients give updates on what they want more easily. | This one is just something I noticed but I think scrum meetings causes the possibility for some people to compare themselves with their co-workers. |
| Chances of something failing reduces. | Clients may ask for too much in a sprint |
| Clients get to react to the market quicker. |  |

I think we should implement the rest of the DevOps ‘stuff’ as soon as possible to deal with as little roadblocks. (for example, Continuous Integration may end up causing us trouble)

# Test and Validation Analysis

IMPORTANT: by the time I started this section the VPN went down, so it is all based on what I remember from the system since it cannot be accessed!

### Black Box and System Testing

We didn’t have a set of tests that should work during black box testing but we usually did make sure the navigation worked and simple CRUD did as well.

#### Functional Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** | **Actual Result** |
| FN01 | check if the validation while creating a question works (all required fields are set) | Creating a question without test cases should not work | crash |
| FN02 | Login should work with student and teacher credentials | Logs in successfully and assigned to the right role | Success |
| FN03 | Create a set | Works with all required fields set | Success |

#### Defects Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Severity | Priority | Steps to Reproduce |
| 1 | Creating a test case without | 3 | 2 | Go to questions page, click on the create question, fill in the required fields except don’t add any test cases. |

#### Usability Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Purpose** | **Expected Result** | **Actual Result** |
| US01 | Any page should be reachable at any point in under 4 clicks (assuming you are already logged in) | Exactly the purpose | Success (I think) |
| US02 | Should look consistent on Firefox and Chromium based browsers | All content should work exactly the same on both browsers | Success (I develop in firefox while the rest use Chrome/Edge) |

#### Exploratory tests

This can’t be done since I don’t have access to the system.

#### Conclusion

From what I was able to remember, we have focused a lot more on workflow and the broken functionality shouldn’t stay that way for long.

### White Box/Automated Tests Cases

TDD is the process of creating automated tests before writing the code. This way we could make sure whatever feature we are adding works as intended. This is also useful later on while adding other feature, you could check if there were any regressions on a previous feature automatically. This comes at a cost of time used to create those tests and thus requires more resources in the short term.

Code coverage is how much of your code is covered by tests, in a perfect world 100% of the code should be covered. Unfortunately, that is unrealistic. I would say in the 50%-80% range should be okay. It is quite wide, but it really depends on the system.

I cannot asses the tests in my system, due to the VPN being down.

Tests should be run before submitting a pull request, and see if your code breaks any of the tests. We did not make it part of our routine to test our system using automated test, but we did run them occasionally.

Even though I wasn’t able to review the tests we did as a team, but I don’t think we did enough of them or integrated them enough in our ‘routine’.

# Conclusion

This course is very well made and I did learn quite a bit on how to deal with/treat teammates better. So I did get enough value from this course to consider it worthy of teaching again (maybe up the difficulty since they got web 3 too easy??). A cool twist that I didn’t mention this before, but as JF said during our retrospective, we could work with the design students for like a semester and create something fun.

It took me 11 hours to complete this assignment, which goes over the original estimate. I can blame that on the fact I wasn’t ‘really’ working on it, and kind of wasted some time just staring at the screen.