

COMP10120 Final Report

James Peach

May 9, 2014

1 Tools

In the creation of **FlipUp**, our group project, we used a variety of tools. Some were used to help us create the website: IDE's and version control applications etc. And some to help run the application: **PHP**, **Bootstrap** etc. Below I will discuss the main tools we used and how they enabled us to create **FlipUp**.

The website was served from the **Apache HTTP Server**. Licensed under the **Apache** licence, version 2 this tool was the glue that would listen for the web requests and manage the actions of finding the correct script to execute and marshal the response back to the client. This service is a standard amongst the online web-hosts and due to the fact that it is cross platform and open source and would allow us to deploy the finished application easily without issues of consistency between local and production environments.

We used **PHP** for all the server side processing of taking the data from the database and putting it into the HTML template the **PHP** language enabled us to organise the code into distinct separate parts. For each page there was a section to parse any parameters from the request, a section to gather the data from the database and a final section to print out the resulting template with the data.

This separation was key because it enabled us to separate work out amongst the group members, someone would be tasked with creating a template for the page, someone would then make this HTML into a **PHP** template with dummy function calls at the top of the file, and finally those function calls would be then be implemented by another person. This separate nature allowed us to run several of these parts in parallel and then to have some time to stitch the separate parts together. The database was implemented in **MySQL** another open source project. This robust data

store allows us to easily query the data from the separate tables and with the use of the relations between those tables, return the processed results. The flexible structure of the database engine allowed us to easily store all the data that we needed to produce the statistics we wanted on our pages. The website design itself was built upon yet another open source project: **Bootstrap** from **Twitter**. This HTML and CSS template system allowed us to create a simple, good looking website that also worked on mobile devices without any changes needed from us. the **Bootstrap** system proved a success with all the group members both experienced and inexperienced to quickly be able to produce a great looking page. This enabled us to spend more time working on the parts of the project that would take longer.

We used **GIT** as our version control system. This tool was a great help in the production of the app and in organisation of the code. as this allowed us to share our changes with others very easily and also allowed us to work on the same file at the same time with only minimal efforts needed to combine those changes later on. Most of the tools discussed above I had experience with before this course unit, however the experience of working as a group on a project with a deadline took this knowledge into a different context. This was an excellent opportunity to show myself how different things are when you do them as a group as supposed to working on a project alone.

2 Principles

Over the course of this project we have learned about lots of different principles that we have utilised to complete the app. There are lots of layers between our application and the hardware it runs on, the app is available on the internet for instance there are different protocols that the browser uses to first request the page and then to render the content. The use of these principals allows ideas and concepts to be built upon to create new applications if there was no toolchain to build upon then we would be forced to *reinvent the wheel* every time we wanted to create a new application.

In this project we made use of lots of different principles of the web I will mention the ones I found most interesting. We used a completely **JavaScript** based question creator and viewer. this means the server is completely oblivious to the meaning of the **json** string it holds. The two interfaces are then run client-side only and don't require any input from the server, thus allowing the page to seam more responsive and to relieve the server of some work. We also used the principles of **ajax** (asynchronous JavaScript and XML) to allow the user to stay on the question page and the

next question is loaded in the background and then swapped into place on the page. This allows the user to see quicker responses and for the quiz to flow creating a nicer user experience.

We also used some project structure principals in our project. Arguably the most useful was the MVC principal. The separation of these parts as mentioned before in the Tools section allows for the team to work on the separate parts more easily. and lets the different be developed simultaneously. This can take a little getting used to and is not always easy to start thinking this way. However after getting over this hurdle the decisions of how to access data and how to display it become more clear and can often help make the project easier to change and easier to understand. All these things come about because of the structure it provides, for example, if you needed to change the number of rows displayed in a table. you already know where the code sits, you can easily find the code responsible and change it. It also makes testing these changes a less error prone process as you know exactly what effects your change is going to make then you restrict your tests to where they are most required. Our team made good use of this principle and it helped us to distribute the work between the members which made the work simpler and more manageable.

3 Skills and teamwork

Teamwork, organisation of tasks, planning, delegation and teaching were some of the skills that we used collectively to complete the task given to us. The teamwork both acted as a help and a hindrance, as most of us have experience working alone but working in a team after some acclimatisation soon helped us to complete the challenge. The team started off very ambitious and with lots of effort and small demos and ideas but this quickly became stale and not much got done after the great start. But after we realised the deadline was approaching fast we then set upon more frequent meetings and more work done at meeting because we developed a tendency to divide tasks into small sections and with meeting more than a week apart the project did not see any progress made as a whole. To solve this we had mammoth meetings where large sections of the website were created from scratch and with the immediate help of others in the group people learned and developed quickly both the project and their skills. Working in a team has its pros and cons, one of the things I had to adapt to was the delegation of tasks. I often don't want to let others work on the project as I feel that, while they will find a solution, it wouldn't be the way the way that I would

have completed it. This is something that I am getting better at with the teamwork that we have done but still poses a barrier between me and project completion. Early on in the project we all went out and tried to complete a section of the website in whatever way we wanted. We then had group meetings to look at the results and decide which solution was best and how we were going to fit this into our solution. This way we found the best tool for the job. It also afforded some time for us to get warmed up and to learn new things. This helped us to distribute knowledge between the group as there was always an expert that could teach the others.

4 Future

There are usually lots of things to improve on when looking back after you have finished something and this project is no exception. As a group we have all improved our relationships and become better team members as a result. The ability to listen, and more importantly, to cooperate within the group has meant that if we did the same project again and changed nothing else, the project would be finished earlier. With more time and effort given to improve / add more features. Also part of the reason the project wasn't exactly how we planned it and was completed with many planned features missing, was that we did not give enough thought to deadlines. Small parts of the app were given a whole week and massive parts of functionality were to be done within a week. In the future we need to both plan out the project in more detail and be more realistic with the tasks we assigned ourselves and also be more forthcoming if we needed help to meet deadlines. These changes should make the next project more successful and something we can all be happy with.

With all the experience gained from completing the project and evaluating performance, hopefully we will be able to foresee the issues I have identified here as we are working and have the knowledge to fix them as we work.

With reference to the setting of deadlines and the division of work I propose that the tutor has more of an active role within the group, enforcing deadlines and the proper and equal delegation of work between the members. This is not to say that the tutor should completely take over the running of the project as this is fundamentally a group project, however, I think there would be an overall better quality of product at the end. Should the tutor do this then it has the possible downside that the group may become over reliant on the tutor, however, if this was properly handled then the positives

outweigh the negatives.

If I was to give some advice to next year's class on completing their project then it would be to quickly find an idea that all members of the team agree on and then start on a small core part and leave all extras till later. Try listing features you would like to include perhaps assigning each member authorship over a particular feature and make it their responsibility. Once the core of the application is built i would then suggest to integrate their idea into the app with the help of the other group members. Separating the features into these separate smaller parts will result in having a more complete product at the end of the development because there will be lots of different features to the project.

Overall to conclude, I had a lot of fun creating a full application from the ground up in a new group with full authorship on all decisions. I know from experience that this is not always the case and is in some ways more difficult as you always have changing goals as you are the ones who created them. However I still think that I gained a lot from the experience and would definitely do it again!

-James