End-Of-Project Report

# Management summary

The project team aimed to have a fully functional ‘Monster Mash’ game produced and tested by the deadline provided. Unfortunately this was not the case due to a few factors. However, before mentioning why the team did not manage to produce a fully working and tested product, it must be said that the whole team worked incredibly well together and supported each other where required. The problems did not lie within the team’s dynamics but within the initial plan and optimisms.

In terms of the overall functionality of the program, the entirety of the back-end code is present and ready to be integrated with the servlets/front-end. However, this is where we encountered difficulty. We underestimated the time needed to implement and integrate the servlets with the back-end code and its difficulty. Although we had most of the back-end including the JUnit tests to go with it quite early, we should have got producing the servlets and front-end earlier alongside the back-end and attempt to integrate throughout the process.

Also waiting on a standard for server-server communication held us back significantly and despite finding two other groups to pair up with to sort out a standard between us, it was still a bit too late to allow our coders to implement the system. When we realised that it would be too hard to get server-server functionality implemented we decided as a whole to focus on getting as much local functionality finished as possible and focus all our resources on that rather than splitting with the server-server implementations.

By the time acceptance testing came along we managed to get the game working with some integration with the servlets. The sign-up servlet allowing new user to create an account worked, along with the login servlet and the home page servlet. Users were able to create an account, login and view their home page (showing their monsters, initially starting with one) as well as viewing the battle requests, breeding options and selling options pages (however, the last three pages listed here were not fully integrated).

After realising that we might not be able to get a fully working game, we tried to shift resources to help aid the implementation however, although more got done, the actual integration of those parts completed by the other members still couldn’t be implemented in the time left. In the end we managed to get a lot of JavaScript for input validations completed but not integrated into the relevant servlets, the HTML integrated into the relevant servlets, but still missing the main integration with the back-end code.

This meant that other than having a user being able to sign-up, login and view some pages, nothing else could be really tested other than the user’s monsters experiencing their full life-cycles and dying (along with replacing the monster with a new one as they only had the one monster). The back-end code was however done along with the JUnit tests to prove the functionalities are there, they just couldn’t be integrated in time.

In terms of the documents, all documents were completed and in a good state, updated with all relevant feedback provided. The maintenance guide and test report within this document will be completed to the level we can with the program we managed to produce.

# Historical account of project

The first thing we did as a group was allocate the team roles based on our abilities and preferences, although this was always subject to change as needed. We had no problems allocating roles although changes did occur shortly after, again with no problems. The first deadline was for the project plan so we got started straight away on that by listing out the requirements and thinking about what looked easy and hard as well as following the project plan structure provided by the relevant QA document.

We produced use case diagrams along with user interface designs for the time; although designs have been updated since implementation began as instructed. It was not hard coming up with the UI designs as we decided to follow the generic ‘facebook’ game layout, nice, simple and intuitive. The risk assessment table was also completed along with the Gantt chart and task allocations.

Once the project plan was submitted, we got started on the next deadline/deliverable, the test specification. We started by deciding how we would test the system, agreeing that JUnit testing the back-end as it was being implemented was a great approach along with integration testing as the system was to be put together. However, due to how the reality of things happened, integration testing went as far as it could with the state of the program at the end. When designing the tests we took the requirements specification and based the tests on the functional requirements trying to cover the obvious bases first then any not so obvious ones afterwards thus producing our testing table.

After submitting the test specification, the harder phase began; the design specification. This is when the team really had to start thinking as up to this point, the work load was not too bad. We started by discussing certain algorithms that would require some thinking such as how would the monster fights be processed? How would the breeding process occur and what would result? We looked at what was provided in the requirements specification and came up with pseudo code for the more significant algorithms of the program to aid with the design and implementation. Alongside that class diagrams were produced to show a more detailed view what classes would be built and what they would represent, what they depend on etc. Sequence diagrams were also produced to help show how the classes would interact in certain scenarios, these also helped us to visualise how to go about implementing certain algorithms being shown.

A very high level component diagram was produced to show how the HTML servlets simply interacted with the back-end and JSON servlets although not much was known about how these would fully interact at the time of design. At this stage of development, the design phase we hit our first stump, the establishment of server-server standards. This was a major hit to our progress as the coders were not comfortable designing a program that might have to be totally reworked to conform to any new standard established. No progress was really being made between groups so we decided to try and get together with another two three groups but unfortunately that came a little late and we ended up conforming with a bigger standard during implementation week, but even then couldn’t get to the server-server phase due to all the other stumps mentioned below.

The design specification was then submitted allowing us to start on producing the prototype for the demo. This is where things started to go wrong. We had one person, Ivan working/learning how to implement servlets and if we could go back, we probably would have had another doing the same as well as it proved harder than anticipated. We were not able to produce a working prototype but simply produced static HTML with CSS to show how the final product would look and how things would transition, not functionality at all. If we would have had some servlet functionality implemented by the time the prototype was due for demoing, the rest of the implementation and integration may have gone a lot smoother.

Christmas holidays came along and we did plan to have the coders work on things over the holiday, but it was then decided that exam stress was building and it would be best if everyone focused on their exams so that is what we did, we postponed working until after the exams, leaving the bulk till integration and testing week.

The way we structured/planned the build was to have Richard to generate the CSS and HTML for the servlets, Jacob to build the back-end (dealing with the serialising of the data/database and all the main functionalities) as well as tackle the server-server interaction from a back-end standpoint, and Ivan to build the servlets and take Richard’s CSS, HTML and JavaScript, along with Jacob’s back-end code and integrate everything together. Richard managed to get all of the CSS, HTML and most of the JavaScript (for validation purposes such as the sign-up validation) completed and Jacob managed to get most of the back-end implementation done and even tackled the server-server implementation. However, the servlet production and integration were the problem and Glassfish was also causing problems. As Glassfish only managed to work properly on Ivan’s laptop for some reason it made things harder.

# Final state of the project

# Performance of each team member

## Richard Gray – rig6

Richard was an invaluable member of our team, always making sure that he was doing something, contributing in some productive way. Richard even wanted to pursue helping Ivan with the servlet production but it was too late by the time we made the decision to assign Richard to helping Ivan. All of the program’s CSS, HTML and JavaScript were produced by Richard and to a good standard in my opinion.

A vital thing to mention about Richard was the amount of support he gave to people when they needed it, although to be honest, every team member supported each other in one way or another. Also Richard was never afraid to ask questions which really helped reduce setbacks as he never waited till the next meeting to ask questions that may have hindered his ability to finish the tasks assigned to him; he would just contact someone who could help straight away. Great performance all round.

## Edward Davies – edd14

Edward was a key member in helping the team stay on track with our targets and especially good at pointing out things most of the team may have forgot or missed out. Not only he one of the main members to design how the system would look and work but he also was a great deputy project leader when I was not around due to being ill. Edward never hesitated to take over when I had to take leave due to my illness even if it was short notice and he never failed to produce some sort of weekly plan for the team in those events. He kept me up-to-date as well which was really helpful and helped me run our team meetings.

Both Edward and Sam were the two main members that focused on our documentation and they did a good job always making sure that we didn’t miss out on anything or clarifying topics as we went along. They both updated the documentation and designs when feedback was given and amended details as we implemented he actual program.

Edward helped me keep the team organised and has great organisation abilities, even questioning my decisions sometimes (good questions an reasons as well). Great performance all round.

## Sam Morrison – sjm16

Sam was extremely proficient at keeping a record of the team’s minutes and actions as well as working with Edward on the documentation. He kept the documentations in the right format specified by the Quality Assurance documents and ensured that all documentation was produced to a high standard (also double checked by Ollie Roe the QA manager). It was very easy to give Sam a task and just leave him to get to it, if he had any problem he would find someone to help him straight away, he worked non-stop over integration and testing week, revamping our documentation with Edward and was quite independent. Great performance all round.

## Jacob Smith – jas32

Jacob was one of biggest contributors, if anything the biggest contributor to the team. He is an excellent coder, very independent, and driven to get the tasks assigned to him done. Jacob was in charge of coding the back-end database and main functionalities to be integrated with the front-end servlets as well as producing the relevant JUnit tests to go with his code. Jacob event tackled the server-server interaction but as time grew close to acceptance testing it was apparent we would not get the product up to that level so we left that implementation to what was there and focused our resources on the integration.

Jacob was the member that suggested using serialisation as a form of managing a ‘database’ instead of using MySQL. At first the team did not understand the reasons behind that decision but once Jacob had implemented his solution and it worked, there was no reason to change the scheme and it proved to cause no problems anywhere else in the system. Great performance all round.

## Ivan Cholakov – ivc

Ivan along with Richard and Jacob was one of three main coders and was in charge of developing the servlets for the system as well as integrating the whole system together. To be honest the integration lying solely on any one person’s shoulders would be too much now it is apparent how hard the integration of different components of systems can be. However, Ivan did extremely well and was very driven to get as much done as possible even though he kept hitting obstacles in his path.

Ivan was also a very independent coder, I could give him a task and he would be of doing it no questions asked (unless there was a problem) and would come in during integration d nesting week the next day and just pick up from where he left off. Ivan wanted to tackle the servlets by himself and he did well, but if we had a second chance, we would have assigned Richard or someone else such as me to aid him with the servlets. We just all underestimated the difficulties of integration. Overall great performance all round.

## Katherine Rose Farmer – krf

Katherine was the team’s main tester, involved in designing the tests of the system, keeping an eye on the testing throughout the implementation as well as being part of the design decisions for the system. Katherine played a key role with me in supporting wherever we were needed as well such as when integration of the servlets came along. Katherine was very good at evaluating Jacob’s JUnit tests and was very hands on during coding week. Great performance all round.

## Oliver Roe – olr1

As QA manager Oliver was in charge of making sure everyone stuck to standards and all of our documents and programs were produced correctly. Oliver was also one of the team members that kept asking questions to make sure that we were not missing anything. Oliver was the member that mainly kept the team on track, if we diverged from the main focus, he would remind the team of our current position and focus. A lot of quality assurance checking was needed for the project and both Sam and Oliver worked well together to split the tasks, with Sam managing document formatting and skimming over documents where Oliver would deal with the thorough checking and critical evaluating of the documents. Great performance all round.

## Chris Savill – chs17

My role as team leader involved keeping the team focused, motivated and working with good cohesion. I tried to keep the team centralised so our direction was always clear and most of the time we were fine. My main downfall was underestimating the difficulties of integration. If I could do the project again, I would assign an extra coder to help with the servlets and have me help with the coding as well. I don’t think I was proactive enough with the coding, although I helped with the design, the test design and the project plan, I should have put my coding skills to use rather than focusing purely on leadership.

As I leader I think I did well making sure to organise meetings, locations, topics to be covered, and running the meetings, however I should of got involved me with the coding. Also during integration and testing week I got ill and my blood condition prevented me from leading properly so I stayed bed bound one day so I could get well for acceptance testing, luckily Edward took charge, however as I leader I should have been present throughout the whole of integration and testing week. On the topic of being ill, I was ill many times throughout the project, which although I know couldn’t be helped; maybe the role of project leader wasn’t best suited for me due to the illness. My gauge of my performance is good as a leader by should of engaged with the coding more especially during coding week and planned for integration more.