What the project achieved (what parts of the program work and what parts do not; which documents are in a good state and which are not)

The project achieved a working user interface, it was very easy to use with all the buttons being very clear and obvious to what they do. The web application was very easy to navigate with the site ID header, a bar along to the top with all the options, friends at the bottom right of the screen and any interaction button at the center / bottom of the screen for fight or breed etc. The issue's we had with the UI is that when you log in using your username and password it takes you back to the home page and the options along the top do not load until you refresh the page. The other was the UI once logged in each page had to work with you having at least one monster if you did not have any monsters each page shows a null pointer as we always look for the bottom of the list. We did have a Boolean value to set any monster in the position 0 to be invincible and will not age and thus solving this problem but for some reason it stopped working in the final version. Money when buying or selling monsters would be updated in the database but would not update on screen, this was because it took the value from the database when first loaded the side bar. If the user was to log out and in the money would be updated. Given more time this error would have been solved.

For the back end of the project we had everything in place for the algorithms for breeding and fighting trading and all the persistent class to work with the database. The only issue we has was with the database as the one we had tested did not work in the web application so we had to create a new one in the integration and testing week, this meant we had issues with sending request and creating friends and thus stopped us from fighting with friends and breeding and trading. So by the end we just had to set it up internally and fight your own monsters and breed your own monsters just to show it works. For the log in we set up regular expressions for the username and password, we had one for email and then to send the user an email welcoming them to the game however the regular expression for this did not work and neither did the sending of the email so we had to comment it out. Given a bit more time we would had fixed this issue. The Friends List works as long as the friends were in the database, we could not get the requests actions to work before the hand in this is the same for all request actions. For server to server communications be cause we had delays with the database we implemented as much as possible into our code and all we needed was to set up another computer and have their IP address and it would send a request will all the information needed. However we did not have time to set it up with another group but the code is in place ready to use.

The documentation has been checked over by Amy and Sam and is all at a very consistent standard. The only document we had a bit of problem with was the Maintenance Manual this was due to a lack of communication and lack of time to properly check over the document before we had to hand it in.

What difficulties stood in the way of project completion and how they were overcome,

The big difficulty we had was at the start of the integration and testing week, on the Tuesday we got told that Aiman had dropped out and we were an able pair of hands down. This meant we had to share out the jobs Aiman had to do, everyone was very willing to help and we shared the jobs out evenly but this still put strain on everyone to now do all there jobs as well the extra stuff as well. Through out the project it was very hard to get in contact with certain members, we tried to resolve this by setting up a group on Facebook for everybody to put posts and questions and for myself to set the tasks for everybody and we all agreed to this. However this still had problems where people did not look at it or just ignore the comments, this delayed certain work and we had issues with this to the end. When in the integration and testing week we had difficulties when putting everybody's code together as each bit of code’s logic is different as each person made it is different. We overcame this by pair programming by putting the member with myself and we made controller classes that deals with all queries the user could have, this worked really well and saved a lot of time.

How well the team performed

Overall the team worked very well together, commitment, communication and willingness to do the jobs and get them done. Commitment on the whole from everybody was very good having most members at every meeting and doing the work when it came to the deadlines. Communication in the group at times was a bit lacking however we always got the job done. Everybody did talk very well in group meetings with ideas and general discussions, no one argued about differences in opinions or ideas we all worked together to complete the project. One of the main reasons the team did so well was the willingness to help each other and willingness to offer to do a job and for that job to be done.