**Wesley Meade:**

This iteration was a disaster in my eyes. This iteration we moved from allocating work based on our capabilities to spreading the work between us as evenly as possible, I took a small step back since I did most of the last iteration however this was a complete mistake. This iteration had the addition of two of the less important features or as I would say less complex systems to be added. These features being the sharing server system and the customer support system. I have been left on the Sunday 29th disappointed as the customer support system was pushed only this morning at 8am on Sunday 29th and it is completely unusable in terms of this project. I have been asking what state it was in since the start of the iteration and I was assured that it was being worked on. Apparently, Mark was stuck for a week and a half and never said anything about being stuck until this morning at 8am which to me is completely unacceptable. We where hoping to get testing done but that must be put on the backlog now. I now on the 29th after writing this report must fully implement the customer support system from scratch by 10pm and finish the share server’s system. Which this leads to an issue in methodology its either the work gets allocated based on capability and I must do all the work. Or work is allocated evenly, and the work doesn’t get done which then I have to do in a much smaller time frame. At this stage I feel like the project would have been fairer as a solo project. The share server system is working in terms of functionality but is not yet styled hopefully that will be done early next iteration.

**Customize Server:**

This iteration I added the ability to edit a custom server that the customer added to the cart and removed the summary view after server customisation. Added none options for SSD’s and GPU’s.

**Rental Detail:**

I added the ability to extend your rental for up to two years and the ability to end your rental prematurely for a small fee of €200. I also added a second thread in the application which runs asynchronously to the server that checks for expired rentals in the database every two hours and then sets them to expired and removes the server from the database. The only small non-issue with this is that on the closure of the Django run server command via control + c there is no way for me to implement the closure of the thread so when using control + c to shut down the server the terminal will freeze but just closing and opening the terminal again works just as well and this is unnoticeable to anyone actually using the site.

**Payments:**

I added proper handling of errors within the payment system. Now the user cannot possibly enter invalid details and is now prompted with an error message upon doing so.

**Home:**

Added overlay to the carousel

**Customer support:**

Today I implemented customer support in its entirety. I created a list where you can view your transactions and create a ticket based on a transaction and admins can see all transactions. I created a ticket view where you can see all your created tickets and again admins can see all tickets. In the transaction view admins can search through transactions based on the transaction ID. I created a view which allows a user to view his ticket and in that view admins and the user can interact via text chat for customer support. I added the ability for the user to delete their own ticket as well as for admins to resolve it and another to refund the ticket based on the transactions charge ID.

That’s all for this iteration I hope the next iteration will be a more productive team effort.

**Mark Leonard:**

In iteration 2 I was tasked with the implementation of the email system and the support system.

I implemented an emailing system that required the use of an app called papercut. I then wrote a simple emailing static method for the emailing system. I created an emailing system for purchase of a rental and for the creation of your account.

I was also tasked with the implantation of support system. I however due to poor time management and my own ineptitude failed to implement a working support system and my colleague Wesley completed it instead.

**Gracjan Kucaj:**

This iteration of the project, I was responsible for the implementation of the server share system. This wasn’t a hugely difficult system to code however there were a few bumps along the road.

The system works in the following way: a user creates their custom server and when in cart, they have the option to save the server. If a user chooses to do so, the method captures the server details and sets the user\_saved Boolean to true. This way we have a way of separation saved and unsaved servers. The user can then view their saved servers on a designated page. From that page the user can share, unshare and unsave the server. These functions are carried out by one method on a basis of else if statements. The user can only interact with servers they have made. This is achieved through an if statement in the loop that displays the servers. The servers are filtered based on the user email. So, we only show the servers that are both user saved and if the email address on the server matches the current user that is logged in.

My biggest issue was surprisingly the buttons on the pages. Since they all worked of one form, I wasn’t sure how to differentiate each submit on each button. However, the fix was very easy, instead of having two or three separate methods, I could have just one with else if statements. That way the methods merge and become more efficient since I only have to get the server id once, instead of three times, if I was to do separate forms and methods for each.