2nd Year Project Document

**Title:** Vortex rental server’s

**Students:**

Wesley Meade X00156077

Gracjan Kucaj X00155111

Mark Leonard X00155073

**Supervisor:**

Patricia Magee

Student Compliance with Academic Integrity

All students are expected to complete their courses in compliance with University regulations and standards. No student shall engage in any activity that involves attempting to receive a grade by means other than honest effort, for example:

1. No student shall complete, in part or in total, any examination or assignment for another person.
2. No student shall knowingly allow any examination or assignment to be completed, in part or in total, for himself or herself by another person.
3. No student shall plagiarize or copy the work of another and submit it as his or her own work.
4. No student shall employ aids excluded by the instructor in undertaking course work.
5. No student shall knowingly procure, provide, or accept any materials that contain questions or answers to any examination or assignment to be given at a subsequent time.
6. No student shall procure or accept assignments from any other student from current or prior classes of this course.
7. No student shall provide their assignments, in part or in total, to any other student in current or future classes of this course.
8. No student shall submit substantially the same material in more than one course without prior authorization.
9. No student shall alter graded assignments or examinations and then resubmit them for regrading.
10. All programming code and documentation submitted for evaluation or existing inside the student’s computer accounts must be the students original work or material specifically authorized by the instructor.
11. Collaborating with other students to develop, complete or correct course work is limited to activities explicitly authorized by the instructor.
12. For all group assignments, each member of the group is responsible for the academic integrity of the entire submission.

**N.B to be filled out by each member of the team.**

By including my name in the form below, I declare that I understand and will abide by the University Regulations and Policies covering Academic Integrity. I accept that each member of the group is responsible for the academic integrity of the entire submission. I will retain a copy of this agreement for future reference.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name (1):** | **Wesley Meade** | **Student No.:** | **X00156077** |
| **Student Name (2):** | **Mark Leonard** | **Student No.:** | **X00155073** |
| **Student Name (3):** | **Gracjan Kucaj** | **Student No.:** | **X00155111** |
| **Student Name (4):** |  | **Student No.:** |  |
| **Module title:** | **Year two project** | | |
| **Programme Title:** | **Computing-Software Development** | | |
| **Date: DD/MM/YYYY** | **27/04/20** |

**Table of contents:**

|  |  |
| --- | --- |
| Page 4 | Introduction |
| Page 5 | Iteration one report |
| Page 8 | Iteration two report |
| Page 10 | Iteration three report |
| Page 12 | Contribution |
| Page 13 | Conclusion |

**Introduction:**

This project is a rental server website which is aimed at all kinds of user’s for example businesses, gamers or even hobbyists. It includes many different features you would expect to see on a rental server website or even rental site for other products, such as a ticket based/live chat customer support system, the ability to customize the components within your rental server and the ability to extend the time of your rentals. Everything on the site has been tested comprehensively and works as intended by us. Even user’s security is considered, and any payment details added by the customer is encrypted. As a group we are very proud of the quality of the site and are very proud to present it. More detailed information on systems and work we have done over the three iterations are on the next few pages. And developer contribution is available in percentages on page 12.

**Iteration One: Report**

**Wesley Meade:**

For this project I was nominated as the “team lead” for the project. For the first iteration it was decided that we would attempt to get as much done as we possibly could and at as high a quality as we could manage within the given time period. My reasoning behind suggesting this method is so that the future iterations can add less important features and be spent perfecting and polishing the site, so it is as high quality as possible. So far, this methodology has gone exactly as plan and we now have a site that is about 70% complete feature wise before entering iteration two.

As for the division of labour we decided to split the work out based on our capabilities, which resulted in myself doing some of the more “bespoke” features. This method of devising labour has been reimagined for the next iteration and tasks have been set evenly and as fair as possible for each member. If my assistance with anything my teammates are stuck with is required, I am happy to assist.

**Cart:**

One of the features I worked on was the cart, it is a session-based cart the same as the previous cart the main difference is with the products being added to the cart. Unlike the cart in our previous project instead of having a product and a quantity of that product each server added to the cart is a unique instance of that server. Which caters nicely for creating the rental for that server when payment is made, and this also made integrating customized servers much easier.

**Payment System:**

The second feature that I worked on was the payment system which had a custom front end and a stripe backend. The front end gives the user the ability to input their debit/credit card information, then they may pay with that information, pay and save that information or pay with an existing saved card. Their card details are saved to the database; however, the card number and card CVC is encrypted using symmetric encryption with a 32-byte key for security this feature works very well. When using the pay with existing card feature their payment details are decrypted however the client never receives the payment details except for the last four digits of the card number so it is secure. When the payment is made a new rental with a unique ID is created using a unique ID generator and the server is moved from unrented servers to rented servers and removed from the user’s cart.

**Homepage:**

I finished off the design and implementation of the main page of the site writing the custom CSS for the page, and the basic view that returns a list of the four prebuilt options we have available.

**Customize Servers:**

I aided Gracjan throughout the implementation of customize servers and solved an issue involving motherboards and CPU’s

**Models:**

I designed and implemented all database models for the site.

All in all, iteration one was a very productive iteration and I am thoroughly looking forward to the next one.

**Mark Leonard:**

In iteration 1 I worked on numerous parts, but my main task was the testing of the project.

I implemented basic unit testing to the main features of the project. For the testing I manually wrote out each test and created a class which contained functions for each value that I was testing. The functions then returned a pre-set value and compared it to the assigned values I had entered. I also manually tested the test cases as to confirm they were working correctly, and I documented the result of the tests.

I prioritized testing the core features as to ensure the functionality of said features were working in the correct manner they were designed to carry out. In iteration 2 I have planned to add more tests such as excepted failure testing to increase the sites performance.

I also helped towards the design of the basic homepage and html. For the design I made use of bootstrap 4. I used bootstrap to create divs and rows to allow for sections on the homepage where the rental servers and future planned features could be easily added. However, it was my colleague Wesley who finished off the homepage.

I redesigned the login and signup pages to match with the customized login and signup pages Gracjan had setup. I used bootstrap 4 again as it was an easy way to edit the entire form in a quick manner. I also added custom CSS to design the forms as it allowed for me to alter the forms by just using the CSS instead of manually changing each input individually. I decided to redesign them as they were just the simple Django forms initially. I helped Gracjan with the customized products section as it was quite a difficult section as we ran into a few issues.

As for iteration 1 I felt I could have done more as such we have assigned a greater amount of work to my workload for iteration 2.

**Gracjan Kucaj:**

For my part of the project I undertook the implementation of a signup/login system as well as the customise product system.

In relation to the signup/login, the work was quite straightforward as I’ve done it numerous times before. I implanted a custom user sign up page allowing for our needs to be met. This signup allows users to add an address as well as their first and last name. This was done through a custom user form that extends Django’s built in user form. I also amended the Django admin views to incorporate these custom fields. This also required me to edit the Django admin file in the users app.

The customise product section was significantly harder. The aim was to allow a user to build their own server from a list of components in our database. The first issue I faced was the fact that not all CPU’s are compatible with all the motherboards. To get around this, the user is asked to select a CPU they wish to use before proceeding with the rest of the customisation process. They are then redirected to the page and asked to fill in the remaining components such as RAM, HDDs, SSDs and GPUs. All the components are fed into the list from the database which allows them to be updated as needs be. Once saved, the current configuration of the server is saved to an unrented server table in the database. This distinguishes it from our pre-built servers and allows the configuration to be added to our cart to be checked out. I encountered a small problem when it came to the motherboards as they were mixing in with the CPU’s and weren’t filtering properly. Wesley however, had a quick fix for this and thus implemented that himself. This section in general would have been significantly more difficult if not for Wesley’s help.

Altogether, I found this iteration interesting as aside from the signup/login system, I was challenged with the customise products, but I feel I learned a great deal from that and should have a much easier time in the future.

**Iteration Two Report:**

**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 were 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. 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 must 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.

**Iteration three report:**

**Wesley Meade:**

From a team standpoint this iteration was a very good iteration. Everyone pushed through and did what they were supposed to do, which was excellent. This iteration was mainly styling, testing, polishing and quality of life changes. Our testing method was writing out test cases and manually combing through all possible scenarios within a feature this was highly effective. Upon finding a bug the bug was fixed and all the tests where repeated until no more issues where found in the feature.

**Edit profile:**

This feature was a feature put on the backlog from iteration two. This is a simple feature that allows users to change profile details, card details and password on the fly.

**Customer support testing:**

Customer support was the largest culprit for bugs as it was rushed on the last day of the last iteration for reasons I expressed in my last report. Most of the issue where with the refund section of the system. These bugs where found and promptly fixed. This part of the project is where most of my time went this iteration as the issues where somewhat complex.

**Rental detail testing:**

Rental detail worked perfectly in terms of functionality which was expected as when I sat down and wrote it, I was taking my time. However, while I was testing, I decided that the feature needed some quality of life changes in terms of the display of the name of the rental I changed it from displaying the unique ID to the name of the server in the rental. This is good as it makes it easier for the user to identify the rental.

**Payment testing:**

The payment system also worked perfectly in terms of functionality. Again, however in terms of quality of life I edited the recording of transactions, instead of recording all products from a cart purchase as a single transaction each product is recorded as a separate transaction. That makes partial refunds much easier in the refund system.

**Expiry system testing:**

The expiry system worked perfectly upon testing and no changes where necessary.

**Gracjan Kucaj**:

This iteration my focus was mainly to style and optimise the sharing server system., as well as running comprehensive tests for edit profile and the sharing system.

Styling the system was not a difficult task since most of the styling was done for the other pages, all I had to do was match that styling. Perhaps the most difficult part of the system was to implement the search bar. The search bar was not overly difficult however I ran into a few problems. One of the problems was that only certain search queries would show results. This boiled down to syntax and it is now fully working as it should be.

Testing the edit profile was extremely straight forward as the system is not particularly complex. There were no bugs that needed to be ironed out either which made this process much easier and quicker.

When it comes to the sharing system, a few bugs needed to be fixed. Mainly, once a server was placed in the cart and saved, and if the user deleted it from their cart without a transaction the server would be deleted from the saved list. This was fixed by creating a copy of the server and saving that. In this case, even if the server gets deleted, a copy of it still exists. The sharing system itself was working correctly apart from minor display bugs that were quickly stomped on.

**Mark Leonard:**

This iteration my focus was mainly to style and optimise the email system, as well as running extensive tests for the homepage.

The styling for the email was not difficult as I had experience working with the email styling from my project last year. Perhaps the most difficult part of the system was to implement the email system to show the correct values when necessary. The email system wasn’t very difficult, but I ran into a couple of issues. One problem was the correct info being shown in the email. The issue occurred from syntax and is working fine now.

Testing the homepage was quite simple as the homepage is not complicated. There were zero bugs, and such made the process of testing extremely easy.

**Contribution:**

**Wesley Meade:**

Home page (100%)

Cart (100%)

Payment (100%)

Customer support (100%)

Edit profile (100%)

Customize server (50%)

Share servers (30%)

Rental Detail (100%)

Expiry System (100%)

All database models

Login styling

Signup styling

4x test cases

**Gracjan Kucaj:**

Customize servers (50%)

Sign up (98%)

Share servers (70%)

3x test cases

**Mark Leonard:**

Back end selenium testing (100%)

Login (98%)

Email System (100%)

1x test cases

Conclusion:

In conclusion we achieved a fully functional server rental site with multiple feature’s that could be used in many different websites. Everything that is there has been tested comprehensively and to us the site is very visually appealing. We are very proud of the website and in terms of the work we have done up to this point is a large achievement. Everyone learned something from the experience for example how to work as a team, leadership skills and at the end of the day programming and problem-solving skills. These things will stand to us in the future. Implementing a server rental site with all the common features of a server rental site was a large challenge, and I can speak for the team and say I’m proud of the quality of the site given the relatively short development time. In terms of future enhancement’s there are probably back end parts that could be optimized more to make the user’s experience more enjoyable. However, I feel like the system is a relatively robust system that could be genuinely used in the future and I don’t think it could have gone any better with the given time frame.