Analysis of Requirements

The Server Rental website has a number of requirements for its systems that need to be addressed. Each system will require different methods and solutions to problems it might present. This document is aimed at addressing these systems and going through ways for these said systems to be introduced without causing issues that will halt the development cycle of the project.

# User

The user system will require various factors in order to work properly. Firstly, the user will sign up, providing an e-mail, username and password. These will be stored in the site’s database. These are required in order to tie a user to an account. They also ensure that only the user will have access to their account. Additionally, the user will need to provide a billing address and payment details in order to process purchases made on the site. These will also be stored on our database. Payment details will be encrypted.

# User profiles

Each user will have a unique profile. The profile will allow the user to change their password, e-mail address, billing address and payment details ( if the user wishes to save the details). These changes will be linked to the database and will update the existing tables. We will not keep records of these changes, hence there will be no ability to revert changes once they are made without re-entering the changed details. The user’s profile will also display the servers they have currently rented, their expiry date and any other details regarding it. This will also be stored on our database, mainly between the users table and the rented servers table.

# Admin

Similar to the user, the admin will provide an e-mail, username and password that will be stored on our database. The main difference between an admin account and a user account is that an admin has certain privileges that a user will not have. These privileges include the ability to add or remove products from the store front as well as update existing product descriptions, prices etc. This will be achieved through the use of the Django admin website that is by default provided by Django upon the creation of a project. The admin website might be edited in order to provided functionality that is necessary for our website to run.

# Shop

The shop will require many factors for it to work. Firstly, a table of products from the database will be linked to the shop, allowing us to put products on sale. The display of the products will include the component name, price and specifications, which will all come from the component table in the database. The shop is also going to have a selection of pre-built server available to rent, these pre-built servers will be stored in a server table. Aside from pre-built servers, the shop will allow users to build their own customised servers from a list of components. These components will have their own table in the database. To build your own server, the user will fill the components from drop down menus linked to specific components. Once submitted, the custom-build will be added to the server table. The user will also have the option to share their custom-build server with the store. This will take the custom server from the server table and place it into the product table allowing it to show up on the store front. Component availability doesn’t really matter in this situation as the servers are not yet rented. This will be dealt with in the cart system.

# Cart

The function of the cart is to take products from the shop and create an order for the user. This cart will be session based. The cart will require several functions to operate. Details on these can be found within the use case descriptions. The cart will also require the Stripe API in order to process payments. Additionally, a PayPal API could be introduced to offer an alternative way to pay. Once an item is added to the cart, and the user wishes to proceed with checkout, the site will look to the database to see if the components within the server are in stock. IF not a message will appear and alternatives will be offered. However, once the check is successful the cart will proceed to the payment page. This page will require the user to enter payment details if not saved within their profile. Once the order is complete, the user will receive a uniquely generated order number. This order number will tie the server to the user. The server is then moved from the server table into the rented server table. An expiry date is added as well as the name of the user currently renting the server. However since a user can rent more than one server, and many users can rent a specific configuration, the order number is going to be the primary key linking the user to the specific server.

# Support Tickets

This system will offer support to the users from admins. This will require a support ticket table in the database. Each ticket will have a unique ID and an e-mail associated with it. Additionally a short description of the issue will be attached. These tickets will be stored in the database for the admin to pull out. Once an admin pulls the ticket, they will have the e-mail of the person with the issue, allowing for easy contact. The admin can then deal with the ticket, be it technical support within the server, refunds, or general queries. For refunds, The admin would have to manually expire the server which would move it from rented servers table into the server table. The admin would also then issue a refund manually. A log of tickets and interactions between the admin and user will be kept for documentation. Once the ticket is resolved, it will be moved from the support tickets table to the completed tickets table.