ITopiaLogo

**Technical Design**

**Client Company:** Plaintech

**Project:** VIRT Team 6

**Authors:**

Cyril Adjei 500695104

Annika de Graaf 500662415

Jast Hamelink 500693975

Abdel Ochan 500661495

Brian van der Raaij 500639928

**Version:** 1.0

**Status:** Final

**Date:** 11 January 2015

**ITopia is subsidiary of the Hogeschool van Amsterdam**

**Versioning**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver.** | **Status** | **Date** | **Author(s)** | **Changes** |
| 1.0 | Concept | 02/01 | Cyril Adjei | - |
| 1.1 | Final | 09/01 | Cyril Adjei | Specified and finished |
|  |  |  |  |  |

**Approval**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Carried out by** | | **Checked** | | **Approved** | |
| **Ver.** | **Name** | **Date** | **Name** | **Date** | **Name** | **Date** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Table of Contents

[1. Introduction 4](#_Toc408763370)

[Purpose 4](#_Toc408763371)

[Scope 4](#_Toc408763372)

[Document Organization 4](#_Toc408763373)

[Audience 5](#_Toc408763374)

[*Itopia resources* 5](#_Toc408763375)

[2. Document organization 6](#_Toc408763376)

[2.1 Network Design Overview 6](#_Toc408763377)

[*Use case scenarios* 6](#_Toc408763378)

[*Activity diagram* 7](#_Toc408763379)

[*Website mock up/wireframes* 8](#_Toc408763380)

[*Front-end-website* 10](#_Toc408763381)

[Website creation 10](#_Toc408763382)

[2.2 Technical Design Overview 11](#_Toc408763383)

[*Archimate model* 11](#_Toc408763384)

[*Database* 12](#_Toc408763385)

[Security 13](#_Toc408763386)

[Backup 13](#_Toc408763387)

# Introduction

### Purpose

The purpose of this document is to outline the technical design of the server setup and installation.

Its main purpose is to –

• Provide the link between the Functional Specification and the detailed Technical Design documents

• Detail the functionality which will be provided by each component or group of components and

show how the various components interact in the design

This document is not intended to address installation and configuration details of the actual implementation. Furthermore this document is a generic Technical Design Document for use by Plaintech. It provides guidance and template material which is intended to assist the relevant management or technical staff, whether client or supplier, in maintaining the project.

Installation and configuration details are provided in technology guides produced during the course of project.

As is true with any high level design, this document will be updated and refined based on changing requirements.

### Scope

The Application Design outlined in this document builds upon the scope defined in the Requirements phase.

### Document Organization

This section should define all the external interfaces. This will be based on a system diagram or context diagram to illustrate the relationship between this system and other systems.

**Network Design Overview:** Describes the overall design of our network configuration and the overall data model for buying VM’s through the website.

**Technical Design Overview**: Describes the overall design of our technical configuration such as the application architecture in in terms of different layers and their relationship to each other.

### Audience

The intended audiences for this document are as follows

* Plaintech NL, Chief Executive Officer (CEO) – Arjen Jansen
* Plaintech NL, Chief Technical Officer (CTO) – Irshad Rampersad, Elmer Hoeksema, Arnim Eijkhoudt, Henk Hoogcarspel
* Plaintech UK, representatives – Wilko Oskam

### *Itopia resources*

* Coaches – Henk Hoogcarspel, Wilma Hoogenhuyze
* Consultant – Various
* Process assessments – Wilma Hoogenhuyze
* Technical Consultancy – TBA
* Project leader – Arjen Jansen

# Document organization

## 2.1 Network Design Overview

We made use of a few use cases scenarios and our activity diagram to be able to create a design of the Plaintech website with the required features and options.

### *Use case scenarios*

**Use Case** Forgotten password  
 **Primary Actor** User

**Preconditions** User is on the website

**Success Guarantee** The user knows what his or her forgotten password is.

**Main Success Scenario**

1. The user clicks on the button ‘’Forgot password’’.
2. The user enters his/
3. The user clicks on send to continue.
4. The user receives an email for confirm.
5. The user opens the mail with a link which direct the user to a page where he/she can change his/her password.
6. The user clicks on the underlined text ‘’ click here for a new password‘’.
7. The user successfully finished the tutorial ‘’ forgot password’’.

**Extensions**

3a. The user clicks wrong

2. The user clicks on ‘’login”.

4a. The user entered an invalid email address.

1. The user edits the email address.

**Use Case** Space per user.

**Primary Actor** User

**Preconditions** User is on the website

**Success Guarantee** The user can address how much space he needs.

**Main Success Scenario**

1. The user logs in.

2. The user clicks on ‘’ manage virtual servers’’.

3. The user addresses how much space is needed.

4. The user gets a confirmation email.

5. The user opens the email with a link which directs the user to a page where the addressing is confirmed.

6. The user has successfully addressed the space.

**Extensions**

3a. The user clicks wrong.

4a. The user does not get an confirmation mail.

1. The user edits the email address.

**Use Case** Adjusting subsribtion.

**Primary Actor** User

**Preconditions** User is on the website

**Success Guarantee** The user can adjust his subscription.

**Main Success Scenario**

1. The user logs in.

2. The user clicks on ‘’account settings’’.

3. The user clicks on ‘’subscription’’

4. The user changes the subscription.

5. The user gets an email with detailed information regarding the subscription change.

6. The user has successfully changes his subscription.

**Extensions**

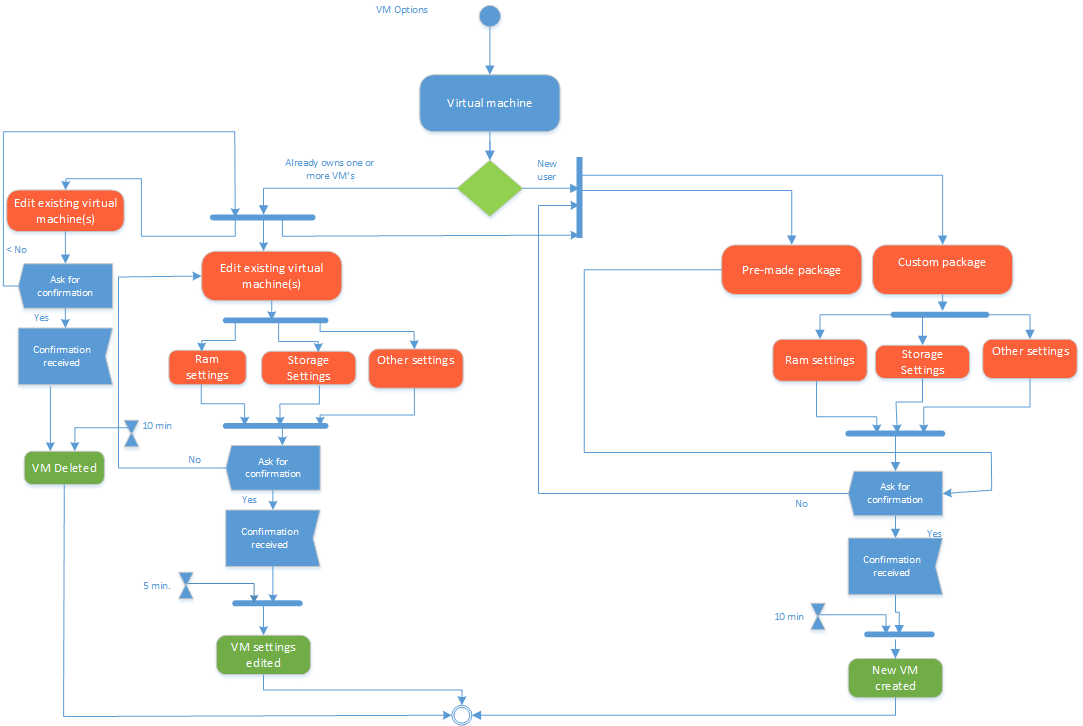
3a. The user clicks wrong.

4a. The user does not get an confirmation mail.

1. The user edits the email address.

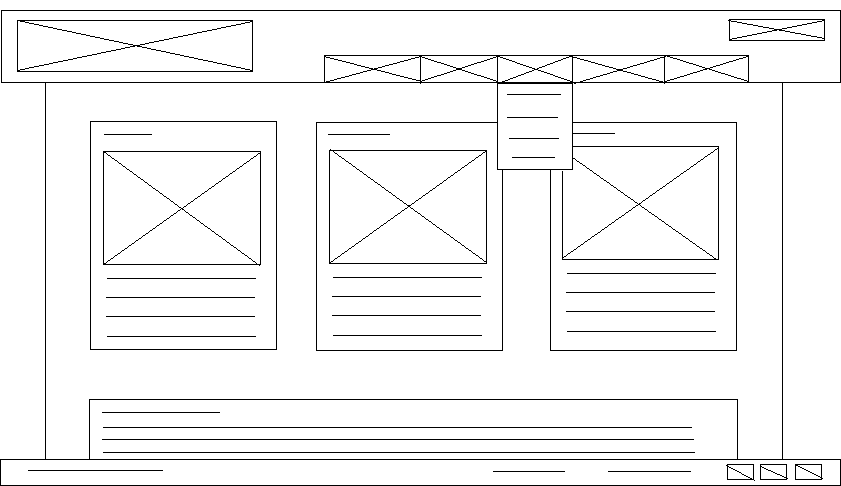
### *Activity diagram*

This diagram will give a visualisation of the steps a customer has to take to do specific tasks on the website.  
We can make out that the requirements of Plaintech are fully included if we look at the process how a VM is created/deleted.



### *Website mock up/wireframes*

*The front page of the website:*



The top part, the header, exists of three separate elements. These elements are the logo, the menu and the login button.

**The logo** is the first element on the top left. This is the obvious choice in terms of placement for the logo as it is usually where people start reading (top, left).

**The menu** is placed at the top as well, so that it is easily to navigate through the website and to the page where the customer is looking for. A feature of the menu is that it will become a drop down menu, so that there are more choices underneath each menu option.

**The login** button will redirect the user to a different page, on which they can login.

More details about the login page will be described underneath the second wireframe.

In the middle of the page, all the content will be placed. This consists of two main elements.

**The three columns** which are going to contain the different service levels Plaintech UK will offer their clients, putting them right at the top of the main content will make them very visible and makes the customers more prone to look at them.

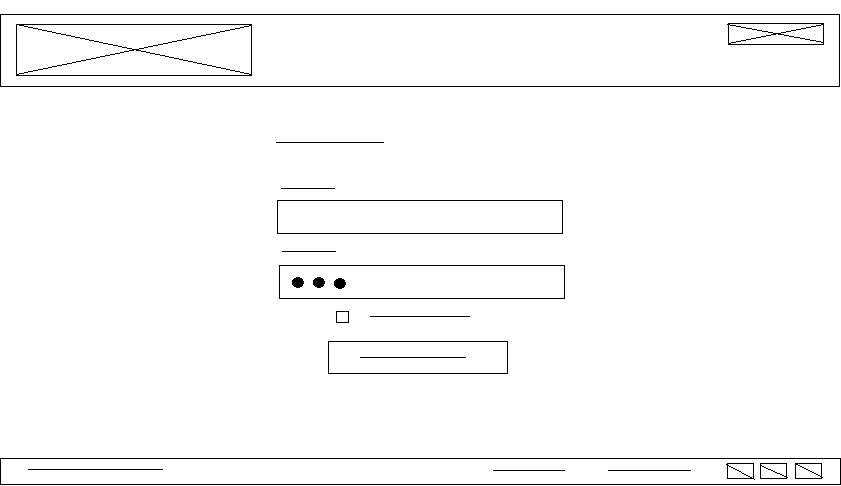
**The main text field** in which information will be presented to the customers.

At the very bottom of the page there is a footer. The footer usually contains the name of the page, copyright details and sometimes more links. The footer in the wireframe consists of two elements.

**The copyright details** are located on the footer so that it will be known Plaintech UK is the owner of all information on their website.

**The (social media) links** are also located on the footer so that people can look Plaintech UK up and connect with them on more webpages.

The login page of the website:



The login page of the website will obviously be in the same style as the front page of the website and all other pages, however this page does not contain the menu as this page works slightly different. The page grants access to the virtual machine as the user will be logged on the user panel here.

Since the **header** and the **footer** did not change aside from the lack of menu, we will only explain the main content.

The main content consists of four elements. The username input field, the password input field, the check box and the continue button.

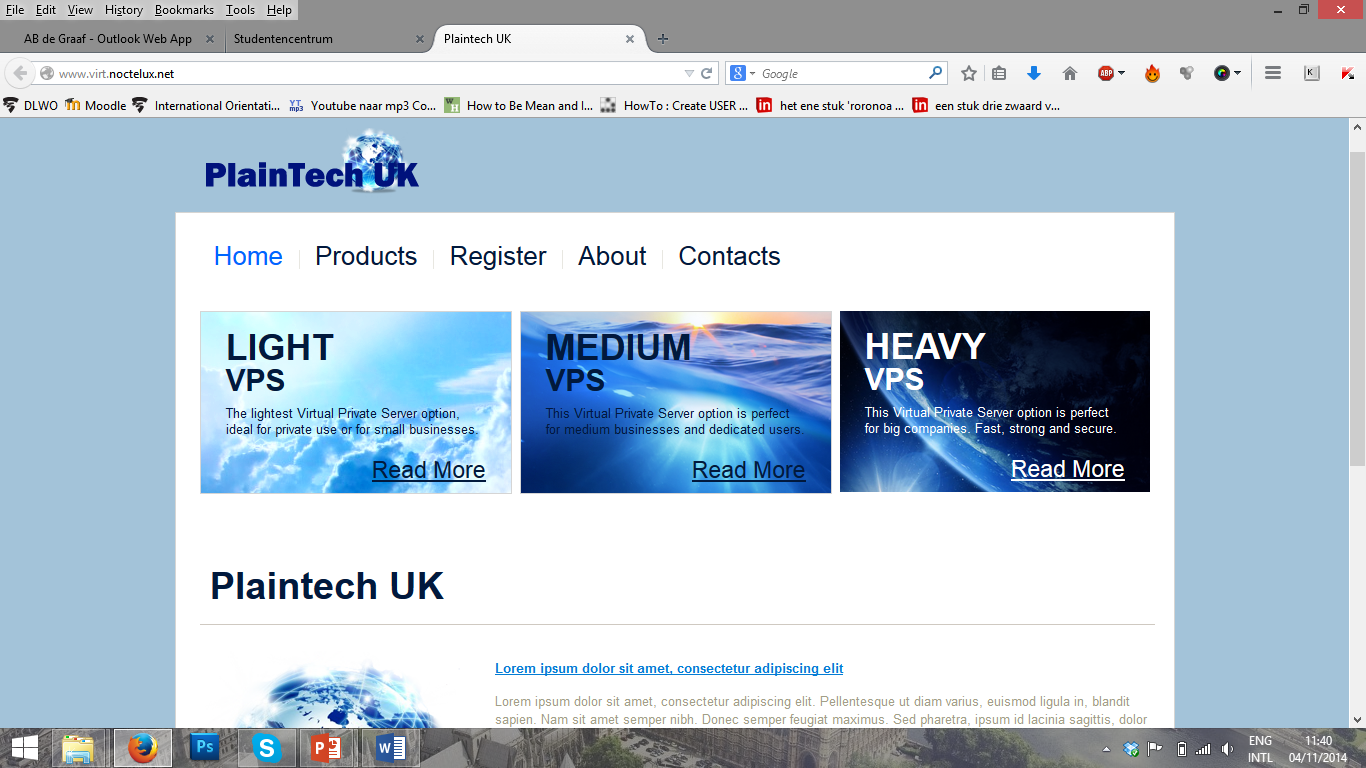
**The username input field** is basically a plain text input box where the user is meant to type their username so that it can be compared to the database.

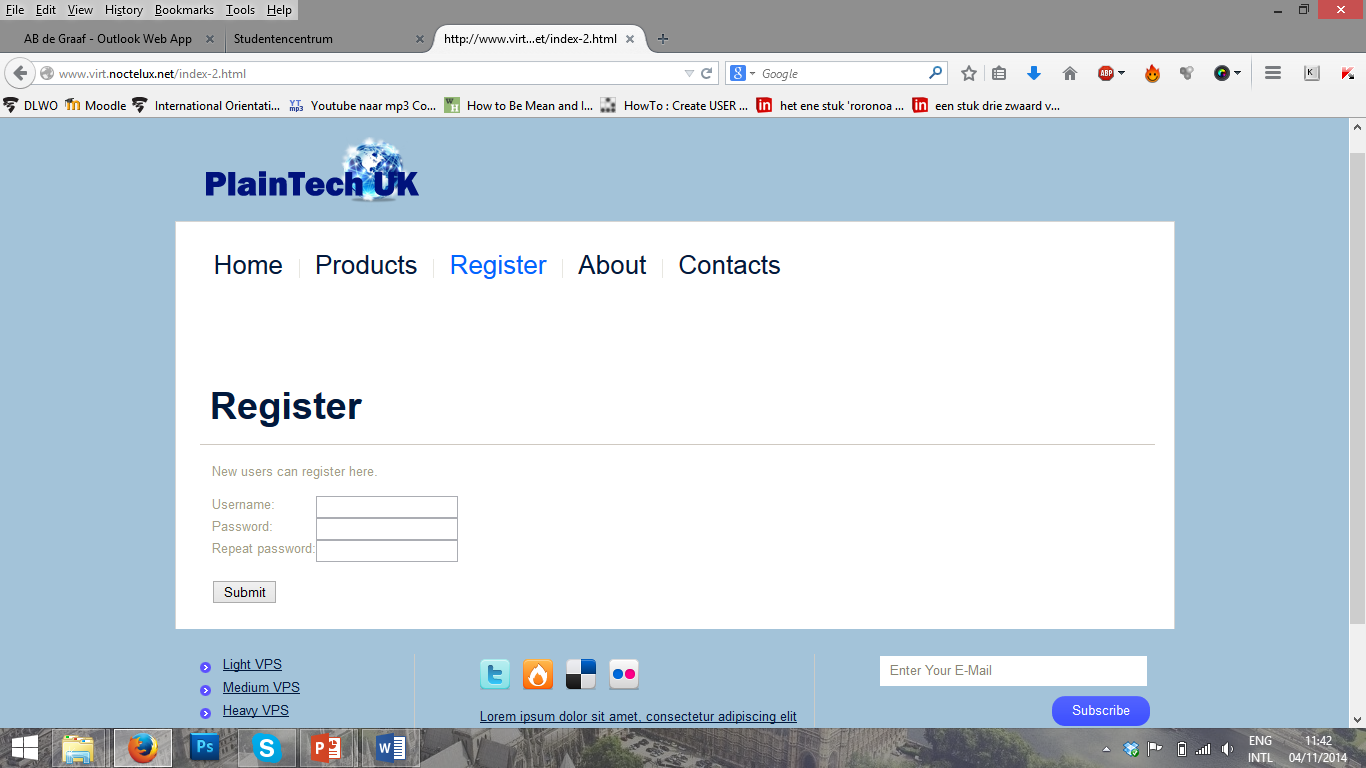
**The password input field** is a text input box where instead of the actual characters it will show dots to protect the user from people that might look at their screen. This is one of the most basic forms of security we will implement.

**The check box** is there to allow users to use the ‘remember me’ option, which will basically allow their browser to remember the username and password for them so that they do not have to put them in every time they want to login, even though this is not recommended.

**The continue button** is placed underneath the other elements since this is the last thing the user has to click on before they can continue. Pressing the button will initialise a connection with the database so that the username and password can be compared to the ones stored in the database. If they are the same, access to the control panel is granted.

### *Front-end-website*

****

****

Current design (11-01-2015)

### **Website creation**

The website is created with HTML.

The layout of the website is mainly managed through .css documents, or Cascading Style Sheet documents.

These documents contain all the details on the style used for the website.

The external style sheet is used by Dojo, which we used for certain elements within the website.

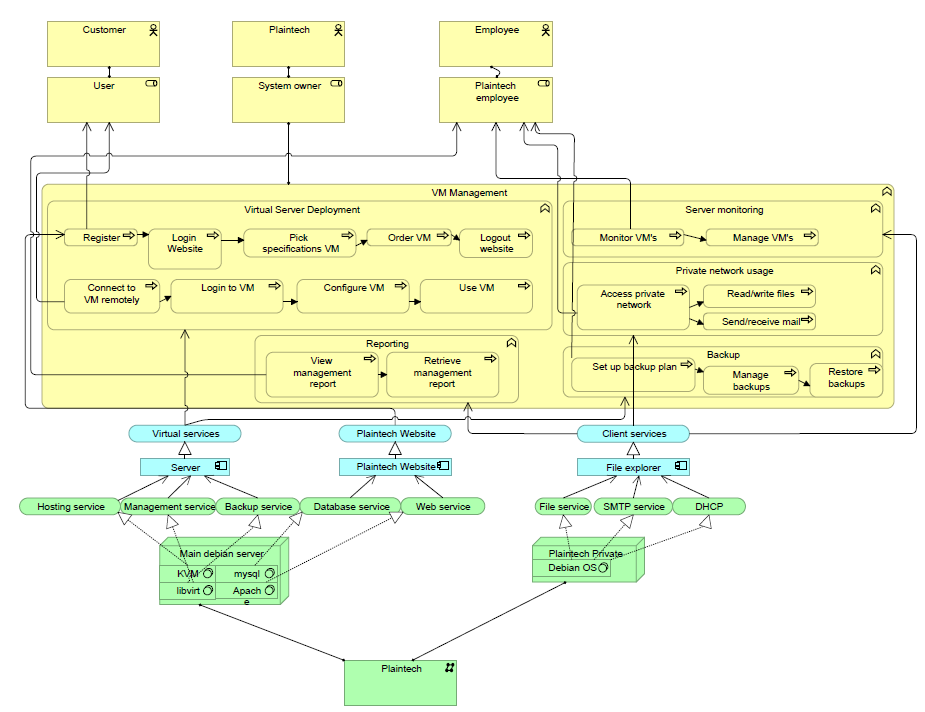
## 2.2 Technical Design Overview

### *Archimate model*

We have designed an Archimate model for the business, application and technology layers.  
This way we can provide a clear understanding of how our technical layer is communicating with the website, how the user rights are set up and how the servers are being used as.

The main debian server is running the following services:

* **KVM (Kernel-based virtual machine)** Is a full virtualization solution for Linux on x86 hardware containing virtualization extensions
* **Libvrt** -  Is an open source API, daemon and management tool for managing platform virtualization. (In other words managing KVM).
* **Mysql -**Is a database management system, an structured collection of data.
* **Apache2 -** Is a freely available Web server that is distributed under an "open source" license.

The Plaintech private server is being used a file, dhcp and SMTP (mail) server.

Yellow – Business layer  
Blue – Application layer  
Green – Technical layer

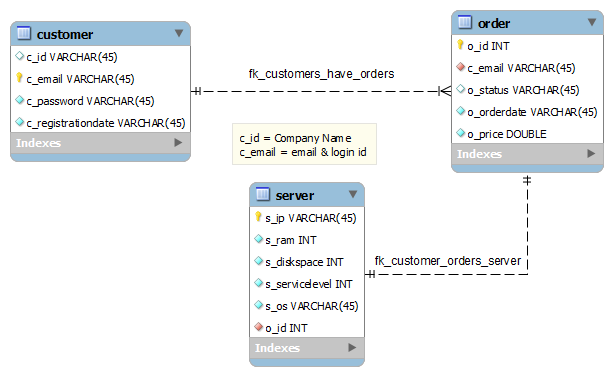
### *Database*

The database ERD has a simple structure of 3 tables, each containing logical colums.

* Customer
* Order
* Server

In the following ERD we can clearly see which table has what coloms.  
We can also see the logical relationships between the coloms.  
  
A customer goes on the site and registers for an account. If the customer decides to purchase a VM, the customer is prompted to fill in their customer data such as pay method and SLA (Service Level Agreement) to complete the order.

After the customer has successfully filled in the required fields of the order, the order goes to the main debian server were the order is being processed by our technical staff.



### Security

To keep the server and all the information on it secure, several security measures are required.   
Hackers or other people that are trying to access the server have lots of ways to brute force (attack) the server.

We selected a few that we used on the server, however there are far more and different ways to secure a server.

We used the following security strategy’s/packages:

* **Fail2ban** – Is a software package that is created to protect a server against malicious attacks like brute force attacks.
* **Disabling the root account** – Is for disabling the root user and allowing another user to assume the root users permissions. This adds another layer of security because an additional username and password must now be entered before gaining the root user privileges.
* **SSL (Secure Sockets Layer)** - Enhances online security by encrypting the communication between web server and browser.
* **Hashed passwords** - Is one of the most basic security considerations that must be made when designing any application that accepts passwords from users.

### Backup

The back-up system of the server is a requirement for 2 of the 3 SLAs (Service Level Agreement) we have with Plaintech UK.

* Light SLA – No backup
* Medium SLA – Contains backup
* Heavy SLA – Contains backup

We are running backups through a script which is described in detail in our system documentation.