

wendell job

Versie: 0.7 (Concept)

Technisch Ontwerp V.0.7

Contents

[Introduction 2](#_Toc408778356)

[Revisions 3](#_Toc408778357)

[Scope delineation Inside the scope 4](#_Toc408778358)

[Outside the scope 4](#_Toc408778359)

[Project specifications 5](#_Toc408778360)

[Devices 5](#_Toc408778361)

[Placement 5](#_Toc408778362)

[Network design 6](#_Toc408778363)

[Raspberry 7](#_Toc408778364)

[Captive portal 8](#_Toc408778365)

[Tomcat installation: 9](#_Toc408778366)

[HTML page 10](#_Toc408778367)

[Ip Tables The following commands were used for the changes of the ip tables: 14](#_Toc408778368)

[Tomcat (7) default page 15](#_Toc408778369)

[Servlets 16](#_Toc408778370)

[DHCP & DNS 18](#_Toc408778371)

[Sources 19](#_Toc408778372)

# Introduction

In this document you will find all technical information regarding the installation of how to make a captive portal.  
  
All the devices will be placed inside a Corendon-owned Boeing 737-800.  
  
The following information will be discussed in this document:

* Raspberry pi specification and installation
* Device quality
* Access point location/placement
* SSID name
* Servlet and servlet code
* IP tables
* Tomcat installation
* Html page layout and code

## Revisions

22-10-2014

* Added revisions table
* Changed database from physical objects to the configuration of the tomcat side
* Information added to device specifications table
* Subject “captive portal” added
* *To-do: change database from physical to digital (tomcat specifications)*

19-11-2014

* Subject protocols removed: *no content*
* Subject Database removed: *no content*

16-12-2014

* Expending information
* Pictures updated
* Combining the old TD from Jasper with the new TD

## Scope delineation Inside the scope

* Corendon have a Captive Portal so the users can login.
* Debian configuration and installation
* WiFi configuration
* Switch configuration
* Captive Portal configuration
* Firewall configuration
* Access point configuration

## Outside the scope

* Wiring
* Installing an internet backbone into the airplane
* setting up a database to store the passengers information

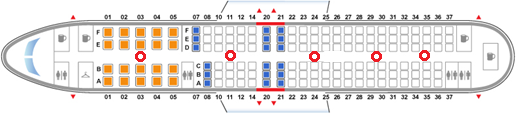
## Project specifications

### Devices

|  |  |
| --- | --- |
| Model | Raspberry pi B |
| Quantity | 5 |
| OS | Raspbian Wheezy |
| Storage | Minimal 4GB |
| Processor | 700MHZ ARM-processor |
| Hdmi port | 1x Hdmi 1080p |
| Ram | 512Mb |
| Usb ports | 2x USB 2.0 |

|  |  |
| --- | --- |
| Wi-Fi receiver/transmitter | Edimax EW-7811Un Wireless-N |
| Wi-Fi receiver/transmitter speed | 150Mbps |
| Wireless | 802.11b/g/n standard |
|  | 64/128-bits WEP, WPA, WPA2 encryption and WPS |

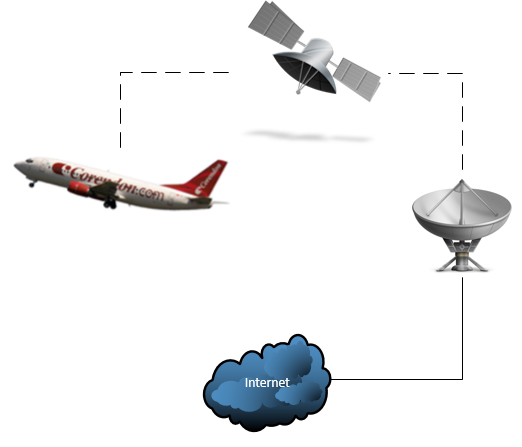
### Placement



The devices shall be spread throughout the plane as to make sure that every location within the plane has a good connection for the entire flight. The wireless access point will be called ‘Corendon’ and will start off with a captive portal.

|  |  |
| --- | --- |
| Wireless connection name (SSID) | Corendon |
| Amount of connection points through plane | 5 |

## Network design

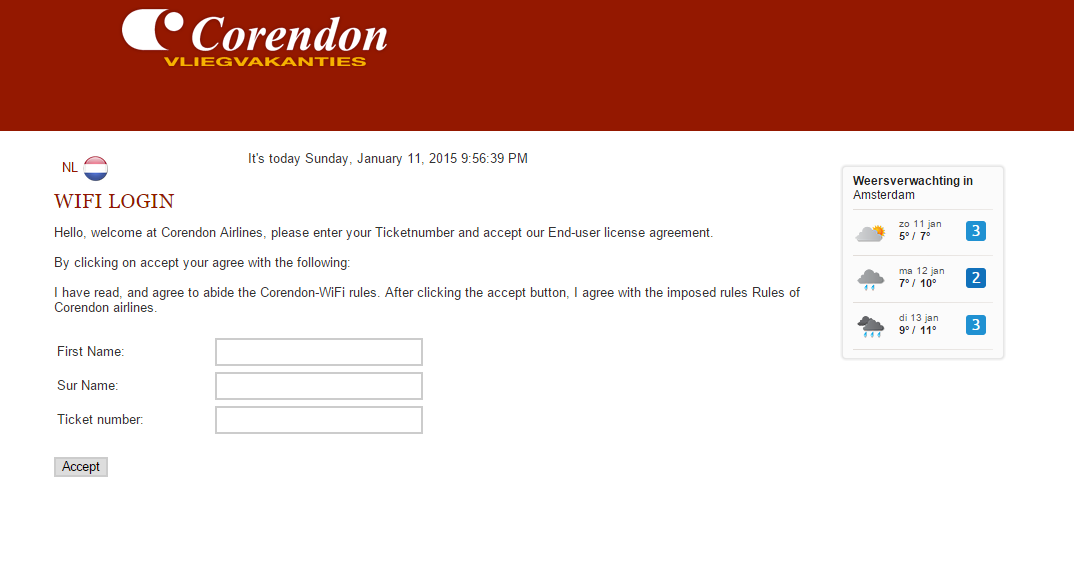
  
In the picture above, you will see the network drawing of this project.  
  
Now follows a short explanation of the network:  
  
Inside the airplane (Boeing 737-800) we’ll place four Raspberry pi’s.  
We have chosen for four Raspberry pi’s, so that all the users have perfect signal strength.   
These Raspberry Pi’s will broadcast a Wireless signal (SSID: Corendon), of which the user can connect to. The user needs to fill in his boarding number and surname to get access to the internet. If this information is not correct the user gets an error, which will tell the user to try again.

Raspberry  
  
We use the following OS for the Raspbarry Pi:  
  
Raspbian  
Debian Wheezy  
Version: September 2014  
Release date: 2014-09-09  
Default login: pi / raspberry  
Kernel version: 3.12  
  
The ISO file is available at the following link:  
<http://www.raspberrypi.org/downloads/>

When the ISO file is downloaded, you need to place it on a SD card.  
You can use the following program to accomplish this.  
  
-Win32 DiskImager  
  
Win32 DiskImager is available at the following link:  
<http://sourceforge.net/projects/win32diskimager/>  
(There are also other programs that you can use to put the ISO file onto the SD card)  
  
When the ISO file is placed on the SD card you need to follow the following tasks:  
  
-Disconnect the SD card from your laptop  
-Put the SD card into your Raspberry Pi  
-Connect your Raspberry Pi to your television (use a Hdmi cable)  
-Plug the power supply into a power outlet  
  
You will see that the Raspberry PI will boor into a configuration page.  
  
Make sure that “SSH connection” is enabled.  
  
Finish the configuration.  
  
When the Raspberry is rebooted, you can use Putty to connect remotely to your Raspberry Pi.  
(Use your ip address, which you can find with the following command: “ifconfig”)  
  
When you are connected to your Raspberry pi, you want to enter the commands below.  
These commands will get your software up to date.

|  |
| --- |
|  |
| apt-get update |

|  |
| --- |
|  |
| apt-get upgrade |

Captive portal  
  
In the picture below you will see an example of the captive portal.   
  
This is what the user sees when the connection with the access point is established.   
The URL in the picture is set to ‘192.168.42.1:8080’, which is the default gateway.  
  
As you can see in the picture above is that no matter which URL the users enters, he will be forwarded to the captive portal.  
  
The IP tables make an exception when the user clicks on the “Submit” button.  
This allows the user to reach other websites.

Tomcat installation:

To install Tomcat server seven, type in the command below:

|  |
| --- |
|  |
| *sudo apt-get install tomcat7* |

We also want an example, for the default page.

|  |
| --- |
|  |
| *sudo apt-get install tomcat7-docs tomcat7-admin tomcat7-examples* |

Of course we also want an interface for our server:

|  |
| --- |
|  |
| *sudo apt-get install default-jdk* |
| *sudo apt-get install ant git* |

Now you want to add an user too your Tomcat server, you can use the command below:

|  |
| --- |
|  |
| *sudo nano /etc/tomcat7/tomcat-users.xml* |

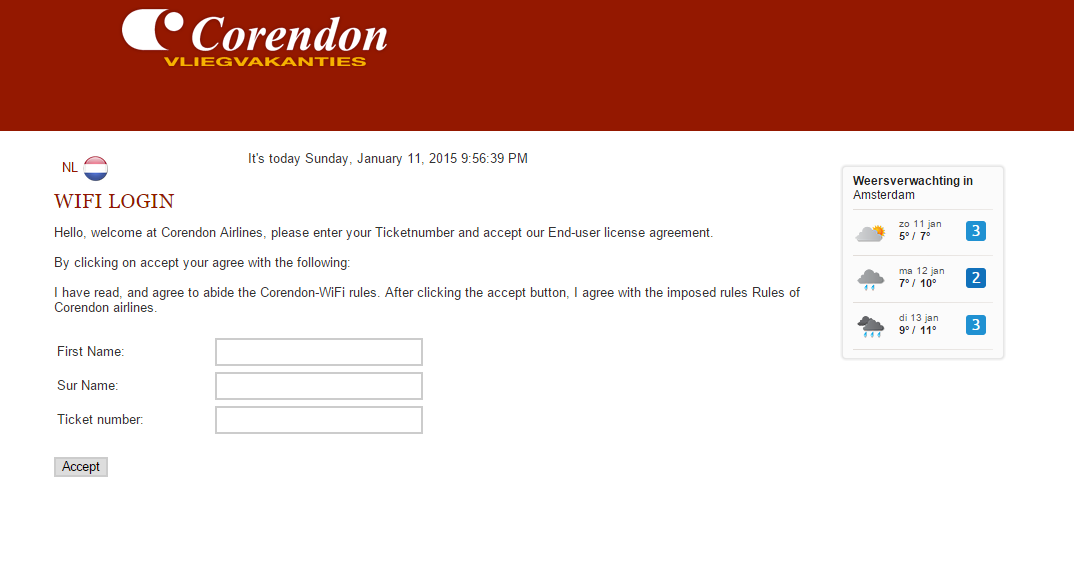
*If everything above has gone correctly, you want to restart your tomcat server.  
You can do this with the line below:*

|  |
| --- |
|  |
| *service tomcat7 restart* |

If everything went correctly; you will now have a working Tomcat server.

## 

## HTML page



The following points have been added to the HTML page:

|  |
| --- |
|  |
| Date and time |
| English & Dutch interface |
| First Name field |
| Surname field |
| Ticket number field |
| Login button |

The following items will be added later on:

|  |
| --- |
|  |
| Weather |
| Photo slide |

The following code is used for the HTML page:

|  |
| --- |
|  |
| <!DOCTYPE html>  <html>  <title>Welkom bij Corendon!</title>  <link href="./css/styles.css" rel="stylesheet" type="text/css">  <div class="page-content">  <div class="col1">  &nbsp; <a href="indexnl.html"> <span id="HistorySavedCount">NL  <img src="nl.png" alt="" height="27" width="27">  </span></a>  </div>  <script type="text/javascript">  tday = new Array("It's today Sunday", "It's today Monday",  "It's today Tuesday", "It's today Wednesday",  "It's today Thursday", "It's today Friday",  "It's today Saturday");  tmonth = new Array("January", "February", "March", "April", "May",  "June", "July", "August", "September", "October", "November", "December");  function GetClock() {  var d = new Date();  var nday = d.getDay(), nmonth = d.getMonth(), ndate = d.getDate(), nyear = d  .getYear(), nhour = d.getHours(), nmin = d.getMinutes(), nsec = d  .getSeconds(), ap;  if (nhour == 0) {  ap = " AM";  nhour = 12;  } else if (nhour < 12) {  ap = " AM";  } else if (nhour == 12) {  ap = " PM";  } else if (nhour > 12) {  ap = " PM";  nhour -= 12;  }  if (nyear < 1000)  nyear += 1900;  if (nmin <= 9)  nmin = "0" + nmin;  if (nsec <= 9)  nsec = "0" + nsec;  document.getElementById('clockbox').innerHTML = "" + tday[nday]  + ", " + tmonth[nmonth] + " " + ndate + ", " + nyear + " "  + nhour + ":" + nmin + ":" + nsec + ap + "";  }  window.onload = function() {  GetClock();  setInterval(GetClock, 1000);  }  </script>  <div id="clockbox"></div>  <div class="col4">  <h1>WiFi Login</h1>  <p>Hello, welcome at Corendon Airlines, please enter your Ticketnumber and accept our End-user license agreement.</p>  <p>By clicking on accept your agree with the following:</p>  I have read, and agree to abide the Corendon-WiFi rules. After clicking the accept button, I agree with the imposed rules Rules of Corendon airlines.  <form action="Servlet" name="loginform" id="loginform">  <form method="post" id="loginform">  <table>  <tbody>  <tr>  <td class="label">First Name:</td>  <td><input id="FirstName" name="FirstName" type="text" value=""></td>  </tr>  <tr>  <td class="label">Sur Name:</td>  <td><input id="SurName" name="SurName" type="text" value="">  </td>  </tr>  <tr>  <td class="label">Ticket number:</td>  <td><input id="TickerNr" name="TickerNr" type="text" value=""></td>  </tr>  </tr>  </tbody>  </table>  </form>  <form action="Servlet">  <div>  <input type="submit" value="Accept">  </div>  </form>  </form>  </div>  <div  style="font-family: Arial; background-color: #fbfbfb; border: 1px solid #e7e7e7; width: 160px; height: 191px; -moz-box-shadow: 0 0 2px 1px #e7e7e7; -webkit-box-shadow: 0 0 2px 1px #e7e7e7; box-shadow: 0 0 2px 1px #e7e7e7; overflow: hidden; -webkit-border-radius: 4px; -moz-border-radius: 4px; border-radius: 4px;">  <div style="width: 160px; height: 270px;">  <div style="margin: 7px 10px;">  <div  style="color: #222222; font-family: Arial; font-size: 12px; font-weight: bold; margin: 0px 0px 7px 0px; line-height: 14px;">  Weersverwachting in<br />  <span style="font-weight: normal;">Amsterdam</span>  </div>  <iframe id="widget-frame"  src="http://www.weeronline.nl/Go/ExternalWidgetsNew/ThreeDaysCity?gid=4058223&sizeType=2&temperatureScale=Celsius&defaultSettings=False"  width="140" height="142" frameborder="0" scrolling="no"  style="border: none;" allowtransparency="true"></iframe>  <a  href="http://www.weeronline.nl/Europa/Nederland/Amsterdam/4058223"  style="background: url(http://www.weeronline.nl/Shared/Images/list\_icon\_blue\_trans.png) no-repeat scroll left 1px transparent; color: #0160b2; font-family: Arial; font-size: 12px; font-weight: normal; padding-left: 14px; margin: 7px 0px 5px 0px; line-height: 12px; outline: none; text-decoration: none; display: inline-block;"  target="\_blank"  style="border: none;background-color: transparent;box-shadow: none;" /></a>  </div>  </div>  </div>  </div>  </div>  <div class="page-header">  <div>  <div class="logo">  <a href="http://www.corendon.nl/" title="Corendon Vliegvakanties"></a>  </div>  <IMG SRC="Corendon\_Logo.png" ALT="Corendon" ALIGN=LEFT>  <div class="mycorendon">  <link href="themes/8/js-image-slider.css" rel="stylesheet"  type="text/css" /> |

## Ip Tables The following commands were used for the changes of the ip tables:

|  |
| --- |
|  |
| sudo iptables –A PREROUTING –t nat –p tcp –dport 80 –j REDIRECT --to-port 8080 |

The command above redirects all the traffic that is normally on port 80 to port 8080, which is the port our tomcat server uses.  
(Port 80 is used for http session)

|  |
| --- |
| IP table |
| sudo iptables –A PREROUTING –t nat –p tcp –dport 443 –j REDIRECT --to-port 8080  sudo iptables –A PREROUTING –t nat –p tcp –dport 80 –j REDIRECT --to-port 8080  iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED  iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT  iptables -t mangle -N internet  iptables -t mangle -A PREROUTING -i wlan0 -p tcp -m tcp --dport 80 -j internet  iptables -t mangle -A internet -j MARK --set-mark 99  iptables -t nat -A PREROUTING -i wlan0 -p tcp -m mark --mark 99 -m tcp --dport 80 -j DNAT --to-destination '172.0.0.1:8080  rmtrack <IP>  **With this command you give the specific mac-address permission to connect with the internet:**  sudo iptables -t mangle -I internet 1 -m mac --mac-source c0:ee:fb:27:9d:b3 -j RETURN  **With this command you deny the specific mac address to connect with the internet**  sudo iptables -D internet -t mangle -m mac --mac-source 64:5a:04:aa:66:a8 -j RETURN |

You must enter the command below to save your ip tables.

|  |
| --- |
|  |
| sudo iptables-save > /etc/iptables.ipv4.nat |

You can show your running ip tables with the following command:

|  |
| --- |
|  |
| Iptables –L |

## Tomcat (7) default page

To change the default tomcat page you need to do the follow:  
  
Change the default tomcat page with the following command:

|  |
| --- |
|  |
| nano /var/lib/tomcat7/webapps/ROOT/index.html |

In this file you will add the following code:

|  |
| --- |
|  |
| <script language=”javascript”> window.location.href = “/SampleServlet/formpage.html”  </script> |

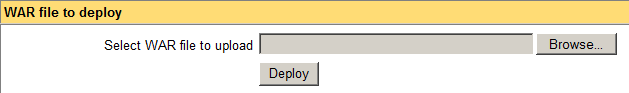
You must place this code between the <head> and the </head>.  
Now when you save the file, the tomcat default page will be redirected to formage.html.

## Servlets

We make use of servlets to process the user input. We need this to check if the user has agreed to the “End-user license agreement” (EULA).  
  
We programmed the code in the following program:  
- Eclipse Luna  
  
You can find the software on the following website: <https://eclipse.org/>

Code:

|  |
| --- |
|  |
| package ProjectTest1;  import java.io.BufferedReader;  import java.io.IOException;  import java.io.InputStreamReader;  import java.io.LineNumberReader;  import java.net.MalformedURLException;  import org.apache.http.HttpResponse;  import org.apache.http.client.methods.HttpPost;  import org.apache.http.entity.StringEntity;  import org.apache.http.impl.client.DefaultHttpClient;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  import java.io.PrintWriter;  import java.lang.ProcessBuilder.Redirect;  @WebServlet("/Servlet")  public class Servlet extends HttpServlet {  private static final long serialVersionUID = 1L;  /\*\*  \* @see HttpServlet#HttpServlet()  \*/  public Servlet() {  super();  }  protected void doPost(HttpServletRequest request,  HttpServletResponse response) throws ServletException, IOException {  String originalURL = request.getScheme()  + "://"  + request.getServerName()  + ("http".equals(request.getScheme())  && request.getServerPort() == 80  || "https".equals(request.getScheme())  && request.getServerPort() == 443 ? "" : ":"  + request.getServerPort())  + request.getRequestURI()  + (request.getQueryString() != null ? "?"  + request.getQueryString() : "");  String username = request.getParameter("username");  String password = request.getParameter("password");  String ticketnumber = request.getParameter("ticketnumber");  // Van hier, %555%  String testString = "1234";  if (testString.equals(ticketnumber)) {  try { |

When you’re finished with the code, you want to Export the project to a “.War” file.  
We need to export this to a “.war” file so we can upload it into our Tomcat server.  
  
See the example below:  
  
  
Now that we have the servlet uploaded in our Tomcat server, it’s time to change our default page to our servlet.

## DHCP & DNS

|  |  |
| --- | --- |
|  |  |
| 192.168.42.2 t/m 192.168.42.190 | For all passengers one. |

## Sources

*raspberry*<http://www.kiwi-electronics.nl/raspberry-pi-512mb>[*http://elinux.org/RPI-Wireless-Hotspot*](http://elinux.org/RPI-Wireless-Hotspot)

*wifi*<http://www.hardwarewebwinkel.nl/netwerk/netwerk-adapters/netwerk-adapters-wireless/wireless-adapters-usb/edimax-ew-7811un-wireless-n-nano-adapter-150mbps-usb.html?utm_source=tweakers&utm_medium=cpc&utm_content=textlink&utm_campaign=pricecompare>

*acces point*[*http://www.staples.nl/wifi-access-point-wap300n/cbs/5887403.html?price=incvat&cm\_mmc=SEM\_PLA-\_-google-\_-feedgoogle&gclid=CjwKEAjwqamhBRDeyKKuuYztxwQSJAA1luvGocuV0Wz6\_k4zuXQUgM9fWKpYrX2m7tl\_por06jpothoCwdjw\_wcB*](http://www.staples.nl/wifi-access-point-wap300n/cbs/5887403.html?price=incvat&cm_mmc=SEM_PLA-_-google-_-feedgoogle&gclid=CjwKEAjwqamhBRDeyKKuuYztxwQSJAA1luvGocuV0Wz6_k4zuXQUgM9fWKpYrX2m7tl_por06jpothoCwdjw_wcB)

*Switch*<https://www.google.nl/shopping/product/11932336231812621196/specs?q=beheerder+switch&espv=2&biw=1448&bih=785&bav=on.2,or.r_cp.r_qf.&bvm=bv.76477589,d.ZWU&ion=1&tch=1&ech=1&psi=otcrVLjPAc7YPMbxgdgI.1412159394693.5&sa=X&ei=B9grVNPvGcvtO9r3gMAC&ved=0CFUQ6iQ>