|  |
| --- |
|  |
| Technical design |
| *Fasten your seatbelts* |
|  |

ITopiaLogo

*Datum: 08-12-2014*

*Klas: IN-102 team 3  
Versie: 0.5*

*Naam stud: Jasper Nota*

*Naam stud: Jony Zeitoun*

*Naam stud: Sjors de Haan*

*Naam stud: Talitha Wingelaar*

*Naam stud: Ward v/d Meulen*

Contents

[Introduction 3](#_Toc405732186)

[Revisions 4](#_Toc405732187)

[Project specifications 5](#_Toc405732188)

[Devices 5](#_Toc405732189)

[Placement 5](#_Toc405732190)

[Network design 6](#_Toc405732191)

[Raspberry 7](#_Toc405732192)

[Captive portal 8](#_Toc405732193)

[Ip Tables 10](#_Toc405732194)

[Ip table range 11](#_Toc405732195)

[DHCP & DNS 12](#_Toc405732196)

[Database 13](#_Toc405732197)

[HTML page 14](#_Toc405732198)

[Layout and code 14](#_Toc405732199)

[HTML & Servlet 15](#_Toc405732200)

[Servlets 16](#_Toc405732201)

[Tomcat (7) default page 17](#_Toc405732202)

# Introduction

In this document you will find all technical information regarding the installation of a wireless network inside a Corendon-owned Boeing 737-800. The following information will be discussed in this document:

* Raspberry pi specification
* Device quantity
* Access point location/placement
* SSID name
* HTML code for the portal page and servlet
* IP tables

In short, this document will contain information about all technical matters regarding the project and will be edited as the project progresses as to keep the reader up to date with every revision.

## 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*

28-11-2014

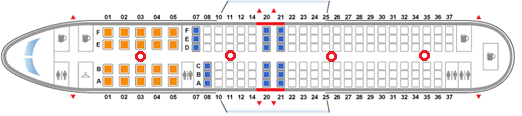
* Subject “HTML page” added
* Subject “HTML & Servlet” added
* Subject “Captive portal” edited (pictures have been updated to follow the new design)
* Subject “IP tables” edited (information regarding the choice of numbers added)

## Project specifications

### Devices

|  |  |
| --- | --- |
| Model | Raspberry pi B/B+ |
| Quantity | 4 B’s |
| OS | Raspbian Wheezy |
| Storage | Minimal 4GB |
| Wi-Fi receiver/transmitter | TP-LINK WL823N |
| Wi-Fi receiver/transmitter speed | 300Mbps |

### Placement

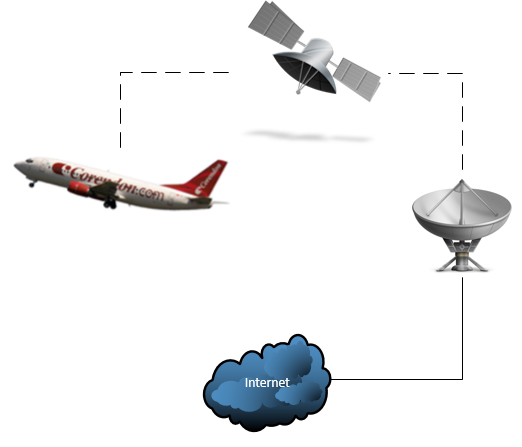


The devices shall be spread throughout the plane as to make sure that every location within the plane has a good connection for the entire flight. The wireless access point will be called ‘Corendon airlines’ and will start off with a captive portal.

|  |  |
| --- | --- |
| Wireless connection name (SSID) | Corendon Air |
| Amount of connection points through plane | 4 |

## 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 Air), of which the user can connect too.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.

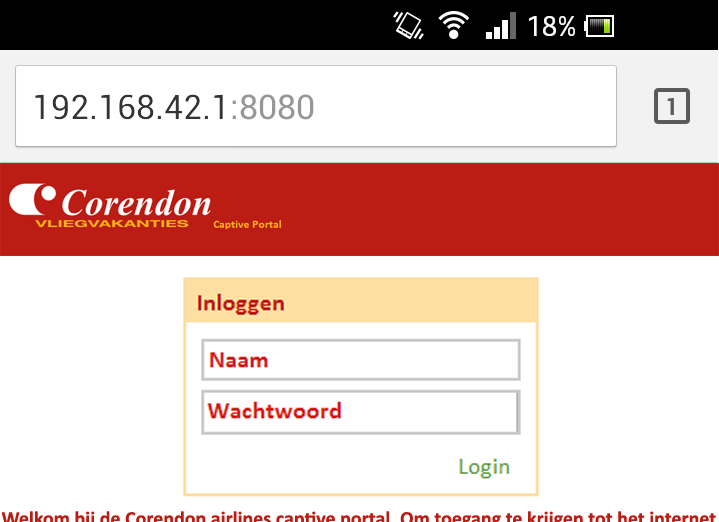
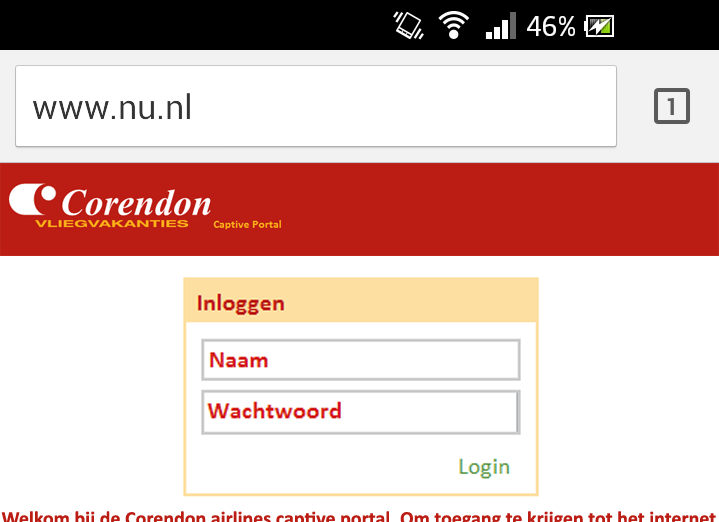


Raspberry  
  
We use the following OS for the Raspberry Pi:  
  
Raspbian  
Debian Wheezy  
  
  
When the ISO file is downloaded, you need to place it on a SD card.  
You can use the following program to accomplish this.  
  
-Win32 DiskImager  
  
You will see that the Raspberry PI will boot into 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 currently in development*)  
The URL in the picture is set to ‘192.168.42.1:8080’, which is the default gateway.  
  
This is what the user sees when the connection with the access point is established.   
  
(*This is currently in development*)  
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.

## 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)

|  |
| --- |
|  |
| sudo iptables –A PREROUTING –t nat –p tcp –dport 443 –j REDIRECT --to-port 8080 |

The command above redirects all the traffic that is normally on port 443 to port 8080.  
(Port 443 is used for https sessions)  
  
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 |

### Ip table range

As described in the chapter “DHCP & DNS” we’ve chosen for an iprange of 192.168.42.2 to 192.168.42.190. This specific amount has been chosen due to the fact that the planes (Boeing 737-800’s) that Corendon will be having the Wi-Fi systems placed in can carry 188 passengers and cabin crew. This is to ensure that every passenger will have 1 ip which can be assigned to him/her.

If a passenger would like to connect 2 devices to the internet, he’ll be notified by the captive portal that the log in code and password he’s trying to use will already be in use by another system (the first system that person has logged in on and is still logged in on) and that he or she will have to shut down the original logged in system first before logging in on the other system.

## DHCP & DNS

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

## Database

We first need to Install Mysql server before we can even make the Database:

|  |
| --- |
|  |
| apt-get install mysql-server |

To run “Mysql” as root, we will use the command below:

|  |
| --- |
|  |
| mysql -u root |

The first thing we want to do is to make our Database:

|  |
| --- |
|  |
| create database Corendon; |

To make chances onto our Database, we will use the following command:

|  |
| --- |
|  |
| use Corendon; |

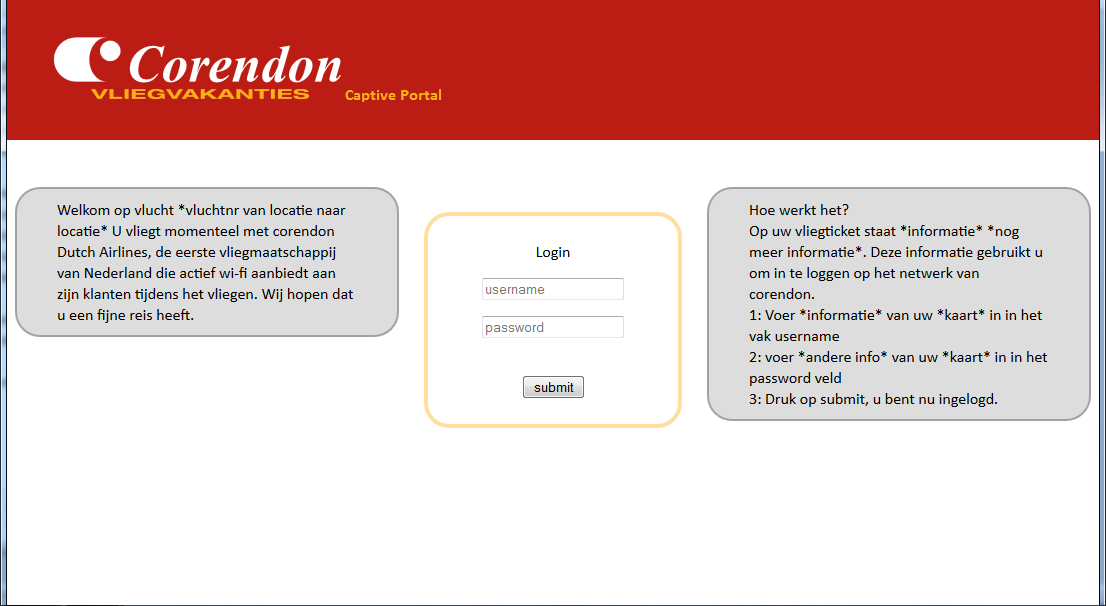
Inside our Database we want to make a table with information inside:

|  |
| --- |
|  |
| create table Passengers ( |
| lastname char(30), |
| firstname char(20), |
| primary key(boardingnumber) ) |

Note: each field is a different command!  
  
To show your tables inside your database:

|  |
| --- |
|  |
| show tables; |

## HTML page



This is the front end of the captive portal page. This page uses the servlet to block the person from entering the internet until they log in using the information printed on the boarding pass.

### Layout and code

A simple layout is being used as to make sure that the user doesn’t have to put in too much effort to log in on the network. Because of this the front end has been designed with just a log in box and two textboxes of information to keep things easy for the user and the log in process.

|  |
| --- |
| Code |
| <!DOCTYPE html>  <html>  <head>  <style>  body {  background-image: url("expansionheader.png");  background-repeat: repeat-x;  font-family:Calibri;  }  form {  background-color: white;  width: 200px;  padding: 25px;  border: 4px solid #FEDFA2;  border-radius: 25px;  margin: 25px;  overflow: hidden;  }  #first {  float: left;  border: 2px solid #a1a1a1;  padding: 10px 40px;  background: #dddddd;  width: 300px;  border-radius: 25px;  overflow: hidden;  }  #second {  float: right;  border: 2px solid #a1a1a1;  padding: 10px 40px;  background: #dddddd;  width: 300px;  border-radius: 25px;  overflow: hidden;  }  </style>  </head>  <body>  <center><imgsrc="header.png"></center>  <br><br>  <div id="first">  Welkom op vlucht \*vluchtnr van locatie naar locatie\*    U vliegt momenteel met corendon Dutch Airlines,  de eerste vliegmaatschappij van Nederland die  actief wi-fi aanbiedt aan zijn klanten tijdens het vliegen.  Wij hopen dat u een fijne reis heeft.  </div>    <div id="second">  Hoe werkt het?  <br>  Op uw vliegticket staat \*informatie\* \*nog meer informatie\*.  Deze informatie gebruikt u om in te loggen op het netwerk van corendon.  <br>  1: Voer \*informatie\* van uw \*kaart\* in in het vak username  <br>  2: voer \*andere info\* van uw \*kaart\* in in het password veld  <br>  3: Druk op submit, u bent nu ingelogd.  </div>    <center>  <form>    Login<br>  <p><input type="text" placeholder="username"></p>  <p><input type="password" placeholder="password"></p>  <br>  <input type="submit" value="submit">    </form>  </center>    </body>  </html> |

### HTML & Servlet

While not present in the previous code, the form will link the html and the servlet as to make sure that there’s an actual functionality on the page. When the user inserts his info in the username and password fields he’ll be redirected to the servlet page to indicate whether he gets access.

## Servlets

We make use of servlets to process the user input.  
We programmed the code in the following program:  
- Eclipse Luna

Code:

|  |
| --- |
|  |
| package servletPackage;  import java.io.IOException;  import java.io.PrintWriter;  import javax.servlet.ServletException;  import javax.servlet.annotation.WebServlet;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class Servlet  \*/  @WebServlet(description = "This is a sample servlet.", urlPatterns = { "/Servlet" })  public class Servlet extends HttpServlet implements javax.servlet.Servlet {  private static final long serialVersionUID = 1L;  /\*\*  \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  // TODO Auto-generated method stub  System.out.println("Test");  String searchTerm = request.getParameter("searchTerm");  String SearchWord = request.getParameter("SearchWord");  PrintWriter writer = response.getWriter();  writer.println("Welcome to the Corendon airlines network."  + "You are flying with a Boeing 737-800. "  + "We wish you a pleasant flight. "  + "Thank you for flying with us. "  + "Your Username is: " + searchTerm  + "Your Password is: " + SearchWord);  }  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  \*/  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  // TODO Auto-generated method stub  }  } |

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.

## 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> |

Now when you save the file, the tomcat default page will be redirected to formage.html.