

VEhub Project Details

Aptean Ltd
Copyright © 2011-2025.

Contents

1 VEhub Project Details.....	1
1.1 Fleet efficiency and safety.....	1
1.2 Accessible technology.....	1
1.3 Easy access.....	1
1.4 Accident records.....	1
1.5 Assets located and logged.....	1
1.6 Reporting.....	1
1.7 Extra.....	1
1.8 Product Folder:.....	1
1.9 1. Development.....	1
1.10 1. Original Requirements.....	2
1.11 2. Product Updates.....	2
1.12 3. Source.....	2
1.13 Quality Control.....	2
1.14 Quality Control - 18072017.....	2
1.15 Web Tools.....	2
1.16 2. Management.....	2
1.17 3. Projects.....	2
1.18 5. Releasing.....	2
1.19 6. Documentation.....	2
1.20 APK - Android.....	2
1.21 APK - IOS.....	2
1.22 Source Code location.....	2
1.23 Git.....	3
1.24 Source/Version Control.....	3
1.25 Branches.....	3
1.26 Push Requests.....	3
1.27 Local vs Remote Repositories.....	3
1.28 The Source Code.....	3
1.29 Projects.....	3
1.30 VEHub.Admin.....	3
1.31 VEHub.Core.....	3
1.32 Calidus VEHub Device (DEV).....	3
1.33 Emailer.....	4
1.34 Emailer Service Installation.....	4
1.35 Location of installutil.exe.....	4
1.36 Location of Service.....	4
1.37 Poller.....	5
1.38 TomTomLink.....	5
1.39 VEHub.Service.....	5
1.40 VEHub.UI.....	5
1.41 Vehub PDF Printer Service.....	5
1.42 Dev Environment.....	5
1.43 Server Details.....	5
1.44 Database.....	5
1.45 Live and Test Environments.....	5
1.46 Live.....	5
1.47 Server Details.....	5
1.48 Database.....	6
1.49 Test.....	6
1.50 Server Details.....	6
1.51 Database.....	6
1.52 Test VeHub SMTP service is running using Telnet.....	6
1.53 Google Play Store Deployment.....	6
1.54 Update Version Numbers.....	6
1.55 Do a Release Build.....	6
1.56 Upload to Play Store.....	6
1.57 Checking for Errors and Faults.....	6
1.58 Vehub Custom Vehicle Checks Flag.....	6
1.59 Vehub Custom Vehicle Checks Flag.....	7
1.60 VeHub Inaccessible Vehicle Checks Issue.....	7
1.61 IMPORTANT:.....	7

1 VEhub Project Details

About

1.1 Fleet efficiency and safety

Maintaining the correct safety information across your fleet just got simpler and faster for you and your drivers. In the process of doing away with most of the paper, OBS Logistics and our strategic partner TomTom reinvented the process of keeping your fleet safe and in compliance with Driver & Vehicle Standards Agency (DVSA - previously VOSA) requirements. We provide a timely capability for recording accident information, as well as knowing where your key transport resources are.

1.2 Accessible technology

CALIDUS VEhub runs as a straightforward App on Android devices, and specifically for TomTom followers, will deploy on all 827x series devices. This gives you an all-in-one driver-orientated dashboard device that acts as both Sat Nav and driver information centre.

1.3 Easy access

All your driver has to do is log on and, through a few easy steps, they can capture all the safety checking information the law requires before starting their journey. Photo capture and defect reporting make the process more consistent and accurate than paper, not to mention faster as an end to end process.

1.4 Accident records

If the worst happens and a driver is involved in an accident, information is best captured as soon as possible.

VEhub has a full accident logging capability, including accident scene and vehicle damage photo capture.

The format driven recording helps to ensure that the details that are required are fully captured. The accident report can then be sent back to the transport department for processing rather than waiting for the driver to return to base.

1.5 Assets located and logged

Add a low cost barcode scanning App to your VEhub and it becomes a rapid data capture tool. All you have to do is attach barcode labels to your fleet assets and your staff can rapidly scan them with VEhub logging the precise location and time, making tracking fast, accurate and above all easy.

1.6 Reporting

CALIDUS VEhub has a full suite of reports so that you can rapidly assess the condition of the fleet and take proactive action to keep your vehicles operating in optimum condition.

1.7 Extra

The Vehub App communicates with a WCF Service to fetch data. The Vehub App is built using the Cordova framework.

The App is available on the google play store

1.8 Product Folder:

\\spekefs2012\Projects\Product\VEHUB

1.9 1. Development

This contains the original requirements, updates and the source code for the current VeHub



1.10 1. Original Requirements

This contains documents laying out the original requirements for VeHub

1.11 2. Product Updates

This contains documents explaining product updates since the original requirements

1.12 3. Source

This contains source code backups and a place to put temporary code to be shared

1.13 Quality Control

This contains QA documents that have been put together when testing VeHub

1.14 Quality Control - 18072017

This contains QA documents that have been put together when testing VeHub

1.15 Web Tools

This contains tools obtained from the web

1.16 2. Management

Project Management related documentation such as handover documentation

1.17 3. Projects

Project Management related documentation such as project requirements

1.18 5. Releasing

Documentation on how to release certain parts of the project

1.19 6. Documentation

This contains any number of documents related to VeHub

1.20 APK - Android

This contains the Android Builds of VeHub

1.21 APK - IOS

This contains the IOS Builds of VeHub

1.22 Source Code location

All of the source code is housed in our web development server's Azure source control and development management system. The home page for the Vision is located here:

Visual Studio 2017(Warning: Cordova project will fail to load in Visual Studios 2019) you can view the solution locally by navigating here and cloning: se your Aptean credentials when prompted.

There are two repos but this is the one that should be used, ?VeHub?



1.23 Git

1.24 Source/Version Control

Track progress of files by creating a series of snapshots and branches which you can move back and forth between.

Why

1. Distribute file changes overtime
2. Prevent data loss/damage
3. Manage complex project structures Created by guy who made Linux to manage Linux, The most popular source control software in the world with lots of documentation and support, and lots of integration with other applications (SourceTree, Heroku, GitHub).

Mercurial - a popular alternative to Git

1.25 Branches

Allow you to create a series of commits on a separate timeline without the need to taint your master branch. Be sure to do an initial commit before creating branches or you risk the master being set as one of the branches.

Pull/Fetch Requests git pull is a Git command used to update the local version of a repository from a remote repository

1.26 Push Requests

The git push command is used to upload local repository content to a remote repository. Pushing is how you transfer commits from your local repository to a remote repository

1.27 Local vs Remote Repositories

Git will allow you to have both Local and Remote versions of your repositories, Local repositories are stored on your local computer, changes made to a local repository won't be reflected in the associated Remote repository until a push request is sent.

Remote Repositories are available to everyone with access and changes to the remote repository can be seen at any time, and with a pull/fetch request a local repository can be created/updated with the latest changes.

1.28 The Source Code

The source code is a .NET Framework 4.5 Visual Studio solution.→ This started out as a paid, externally developed application which was brought in house and developed further.

The solution contains 2 websites, 1 Cordova project and 5 C# projects

1.29 Projects

1.30 VEHub.Admin

1.31 VEHub.Core

The Core Project is used to contain all of the core functionality for the application, from business logic, to helper classes, this allows for greater separation and can be used to create a cross platform application.

1.32 Calidus VEHub Device (DEV)

The android device project, a Cordova application using JavaScript to handle the page layout and and logic.



1.33 Emailer

The Vehub emailer is a windows services that checks the Email table in the database and sends emails out using the details created by the system Alerts.cs.

The emails are sent via SMTP.

If an email should fail to send then email is flagged as unsent and the error is logged.

The Emailer Is Running as a service and is located here on the server

The Settings for the Emailer service are stored in the Emailer.exe.config file:

```
<Emailer.Properties.Settings> <!--Emailer Service Error Log location--> <setting name="LogPath" serializeAs="String">
<value>C:\Services\Emailer\log\</value> </setting> <!--SMTP From Name--> <setting name="SMTP_FromName"
serializeAs="String"> <value>VEHub</value> </setting> <!--SMTP From Address--> <setting
name="SMTP_FromAddress" serializeAs="String"> <value>vehub@obs-logistics.com</value> </setting> <!--SMTP
Server--> <setting name="SMTP_Server" serializeAs="String"> <value>10.43.0.20</value> </setting> <!--SMTP PORT
Number--> <setting name="SMTP_Port" serializeAs="String"> <value>25</value> </setting> <!--SMTP Requires
Authentication Flag--> <setting name="SMTP_RequiresAuth" serializeAs="String"> <value>True</value> </setting>
<!--SMTP UserName--> <setting name="SMTP_User" serializeAs="String"> <value>vehub</value> </setting> <!--SMTP
PASSWORD--> <setting name="SMTP_Pass" serializeAs="String"> <value>{smtpPassword}</value> </setting> <!--SSL
Required for SMTP--> <setting name="SMTP_Ssl" serializeAs="String"> <value>False</value> </setting> <!--How Often
it checks for email--> <setting name="CheckFrequency" serializeAs="String"> <value>60000</value> </setting>
<!--Connection String--> <setting name="DB_String" serializeAs="String"> <value>{connectionString}</value> </setting>
<setting name="UseChilkat" serializeAs="String"> <value>False</value> </setting> <!--PDF Attachments location -->
<setting name="PdfAttachments" serializeAs="String"> <value>C:\inetpub\wwwroot\VEHub.Service\pdf</value>
</setting> </Emailer.Properties.Settings>
```

The Emailer Services uses the configuration to set up and send emails.

1.34 Emailer Service Installation

The services can be installed using a batch file, e.g.: @echo off

1.35 Location of installutil.exe

```
cd C:\Windows\?\Framework\v4.0.30319
```

1.36 Location of Service

```
sc create Emailer binpath="C:\Services\New Emailer{ServiceName}.exe" pause
```

```
if ERRORLEVEL 1 goto error
```

```
exit pause
```

```
error
```

```
echo There was a problem pause and uninstalled using a Powershell script: @echo off cd
C:\Windows\Microsoft.NET\Framework\v4.0.30319
```

```
sc stop {ServiceName} sc delete {ServiceName}
```

```
pause
```

```
if ERRORLEVEL 1 goto error exit pause
```

```
error
```

```
echo There was a problem pause
```



1.37 Poller

The Poller service works similar to the emailer service, it checks the system for vehicles that are overdue a vehicle check then creates an email alert to be sent out.

The OverDue Poller is Running as a service and is located here on the server:

1.38 TomTomLink

Used to connect and interact with the TomTom elements of the application.

1.39 VEHub.Service

The VeHub Web Services that the device applications use to Fetch/Create/Update/Delete data through

1.40 VEHub.UI

This is the PC site for handling of data/account information that the devices are used to record and update.

1.41 Vehub PDF Printer Service

Vehub uses the PDF Printer service to generate pdf reports for gift aid.

The PDF Printer service takes in html generated by vehub and returns a pdf document.

Vehub points to the pdf service via an API endpoint.

This endpoint is stored in the config file of vehub.ui and vehub.service.

<add key="PdfWebApi" value="<http://localhost:3001/%22/>> The Vehub Web Portal has an endpoint that uses the razor engine to generate strongly typed html pages. The html is generated and passed on to the PdfWebApi to generate the pdf.

1.42 Dev Environment

The system is available for use at OBS

1.43 Server Details

Server name: SHARED-WEB-R OS: Windows 2012 Standard IP: 10.43.6.50 RDC User: SHARED-WEB-R <Own User>
RDC Password: <Own Password> IIS Version 8.0.9200.16384

1.44 Database

Server name: SHARED-SQLSVR-R OS: Windows 2012 Standard IP: 10.43.6.51 RDC User: SHARED-SQLSVR-R<Own User> RDC Password: <Own Password> SQL server: Microsoft SQL Server 2012 - 11.0.5058.0 (X64) SA: sa / sa_epod13- Epod user: epod / ePOD1234

1.45 Live and Test Environments

1.46 Live

1.47 Server Details

Server name: SHARED-WEB-R OS: Windows 2012 Standard IP: 172.183.45.50 RDC User: SHARED-WEB-R <Own User> RDC Password: <Own Password> IIS Version 8.0.9200.16384



1.48 Database

Server name: SHARED-SQLSVR-R OS: Windows 2012 Standard IP: 172.184.45.200 RDC User: SHARED-SQLSVR-R<Own User> RDC Password: <Own Password> SQL server: Microsoft SQL Server 2012 - 11.0.5058.0 (X64) SA: sa / caL1duS# Epod user: epod / ePOD1234

1.49 Test

1.50 Server Details

Server name: SHARED-WEB-R OS: Windows 2012 Standard IP: 172.183.45.55 RDC User: SHARED-WEB-R <Own User> RDC Password: <Own Password> IIS Version 8.0.9200.16384

1.51 Database

Server name: SHARED-SQLSVR-R OS: Windows 2012 Standard IP: 172.184.45.205 RDC User: SHARED-SQLSVR-R<Own User> RDC Password: <Own Password> SQL server: Microsoft SQL Server 2012 - 11.0.5058.0 (X64) SA: sa / sa_epod13- Epod user: epod / ePOD1234

1.52 Test VeHub SMTP service is running using Telnet

Open a command prompt as an administrator and enter the following line by line: telnet 10.43.0.20 25 helo ehlo MAIL RCPT DATA This is a email sent via Telnet .

QUIT This assumes the SMTP service is still 10.43.0.20 on port 25, obviously this may change at some point. The recipient email address should be changed to one that the tester has access to.

1.53 Google Play Store Deployment

1.54 Update Version Numbers

Using Visual Studio Open the config.xml file in the solution. In the Common Tab Update the version number:

For Android builds, update the version number in the android tab:

1.55 Do a Release Build

Deploy the app by selecting Deploy in the context menu. Visual Studio will build an APK for the app and output this to the bin directory.

1.56 Upload to Play Store

The Google account that the app is registered to is: ? Google Account: ? Password: THaAw Used for google services, like analytics, playstore , captcha etc? anything google that?s related to dev work:? Google Account: ? Password: _Liverp Play Store upload link: ? ? T ? ?

1.57 Checking for Errors and Faults

System Log Table

```
SELECT * FROM SystemLog
```

The Vehub emailer is a windows services that checks the Email table in the database and sends emails out using the details created by the system Alerts.cs.

The emails are sent via SMTP. If an email should fail to send then email is flagged as unsent and the error is logged.

1.58 Vehub Custom Vehicle Checks Flag



1.59 Vehub Custom Vehicle Checks Flag

VEhub Customers that had an account prior to the new registration will need to have a flag set in order to activate the custom fields functionality.

The flag enables custom fields for the device(app) and the web-portal.

When the flag is set users will be able set custom-fields and drivers will be able to use these fields on the device.

To get started you will need the Customer Code to be able to run the query.

With the customer code, Query the customers table to get the customer Id: `SELECT * FROM Customer WHERE CustomerCode ='Demo'`

Once you have the Customer Id , use it to run the stored procedure `uspCustomerLegacySettingSave` :

To Test this, login to the web portal using the account and check the settings tab. What used to be Vehicle Check Items is now Custom Vehicle Check Items :

With the flag now set, the customer can create custom vehicle check items.

1.60 VeHub Inaccessible Vehicle Checks Issue

When trying to access vehicle checks the following error message is displayed: SYSTEM ERROR

Internal Server Error [500].

It is believed that this is caused when Custom Vehicle Check Templates are introduced after a customer upgrade: these don't have Vehicle Check Items. Therefore if old "vehicle check" indicators are set, for instance by the use of an older version of the App, that doesn't support the newer checks, after the upgrade causing the checking of "vehicle check" items which can no longer be accessed.

The proposed solution is to run a script that will set all the old indicators back to 0 for the new "vehicle check templates". However if there is still a customer device operating the old "vehicle checks" App then the problem will occur again.

1. "Vehicle check" reset script

1.61 IMPORTANT:

? This process needs to be tested before being implemented on a live customer? Replace the `Template:CustomerId` placeholder in the script

```
UPDATE VehicleCheck SET
CkLeaks = 0, CkBattery = 0, CkTyres = 0, CkSpray = 0, CkSteering = 0, CkLoad = 0,
  CkMirrors = 0, CkLights = 0, CkReflector = 0, CkIndicators = 0, CkWipers = 0,
  CkWashers = 0, CkHorn = 0, CkExhaust = 0, CkBrakeLines = 0, CkCoupling = 0,
  CkElectrics = 0, CkBrakesABS = 0, CkBodywork = 0, CkMarkers = 0, CkGlass = 0
WHERE TemplateId NOT IN
(SELECT VehicleCheckTemplateId FROM dbo.VehicleCheckItem WITH (NOLOCK)
WHERE CustomerId = Template:CustomerId GROUP BY VehicleCheckTemplateId)
```

