

# Sherpa Van Scope of Work

20th January 2019

Document prepared by:

Jon Ewing, Creative Director, inframes.com ltd – 0118 9507617, jon@inframes.com

# Project Breakdown

# Admin Dashboard

A password-protected back office dashboard with the following sections in no particular order:

"Quick Search" (town/hotel name) form  
Notifications (new confirmed bookings and expiring hotel subscriptions)  
Pages (Content Management)  
Categories (Content Management)  
Prices  
Bookings/ Customers  
Routes  
Towns  
Hotels  
Statistics

# Users

ID, username (email address), password, role, status, created, last\_login

Some functions (eg. deleting a route) need to be limited to a handful of administrators only. And therefore I think the standard role should be somewhat less privileged - eg. "staff" or "moderator".

# Permissions

restrict access per user per function. Page action name is contained in the "permission" column. Define the permission name for each page action eg, permission would be "route" and you would have a tinyint value that defined whether user has permission to view, add, edit or delete.

We define the permission per action e.g. for permission "Routes" a given user can have 1 for view, 0 for delete, which will allow him/her to view but not to delete the route. Then in the code we will have a class/method that performs the "delete a route" action and the method will first check whether the user has the corresponding permission. This is similar to checking whether the user is logged in, in order to let them see a page on the admin dashboard.

TABLE **permissions**

- id (smallint, unsigned)  
- user\_id (int, unsigned)  
- permission - varchar 128,  
view, add, edit, del -tinyint unsigned

add foreign key for user\_id in "users" table

# User Admin

## Add User

Input email address, role (select list) and (temporary) password

Generate welcome email with instruction to choose a new password

# Route Admin

TABLE **routes** (pending changes)

`route\_id` int(11) NOT NULL AUTO\_INCREMENT,

`route\_name` varchar(45) NOT NULL,

`route\_code` varchar(5) NOT NULL,

PRIMARY KEY (`route\_id`)

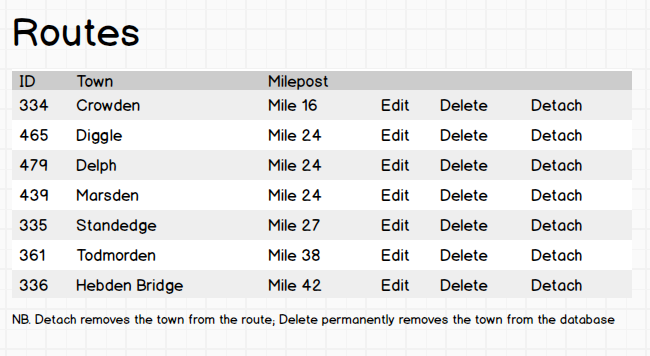
a system to add, edit and delete "routes" for walking holidays (eg. Coast to Coast), specifying the list of towns and villages along the route with the option to upload one or more images, description and other customer values (eg. season start and end dates)

delete process must first check if there are records in the hotel\_route\_link table and route\_town\_link table. Deleting a route is irrevocable, so this should only be permitted for administrator level.

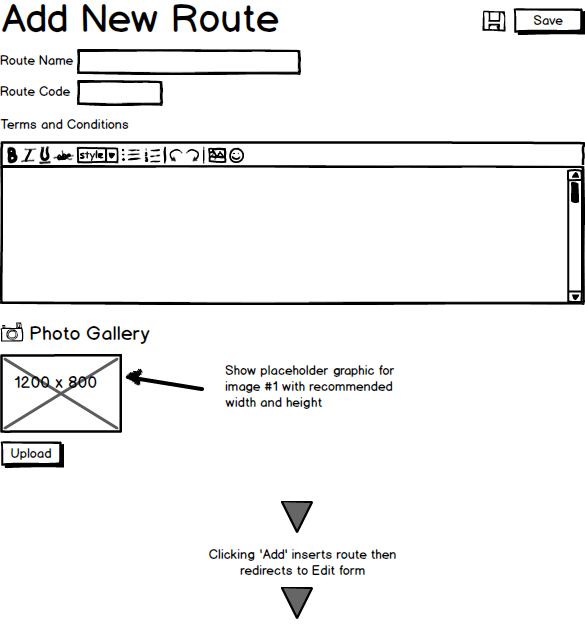
When adding a route, it will need to be "not live" while it is being prepared, so we need a column called "live" (with the value zero or 1 - default zero) in the routes table.

The front-end code will have to check that live=1 for the requested route as part of validating requests.

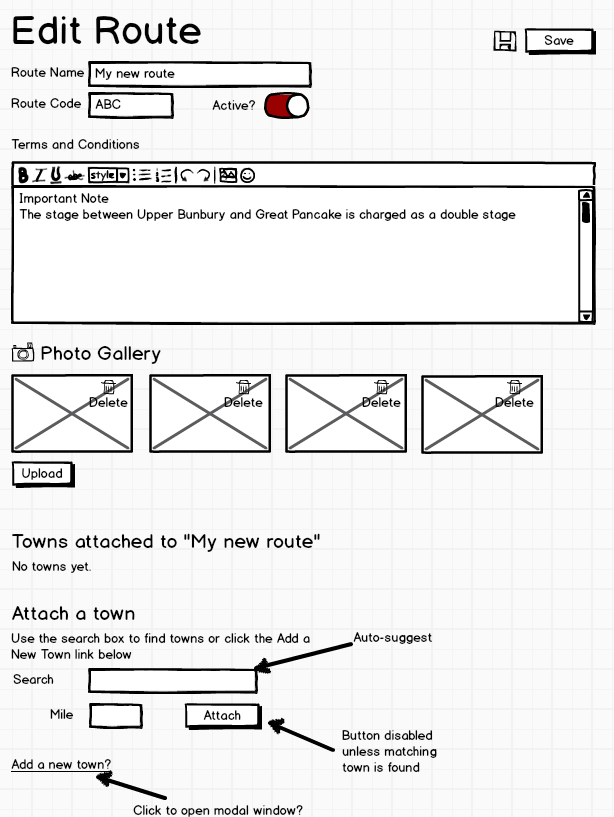
## List Routes

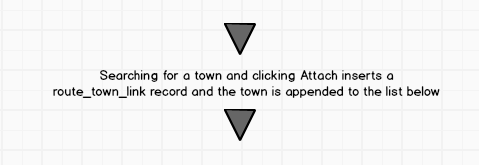


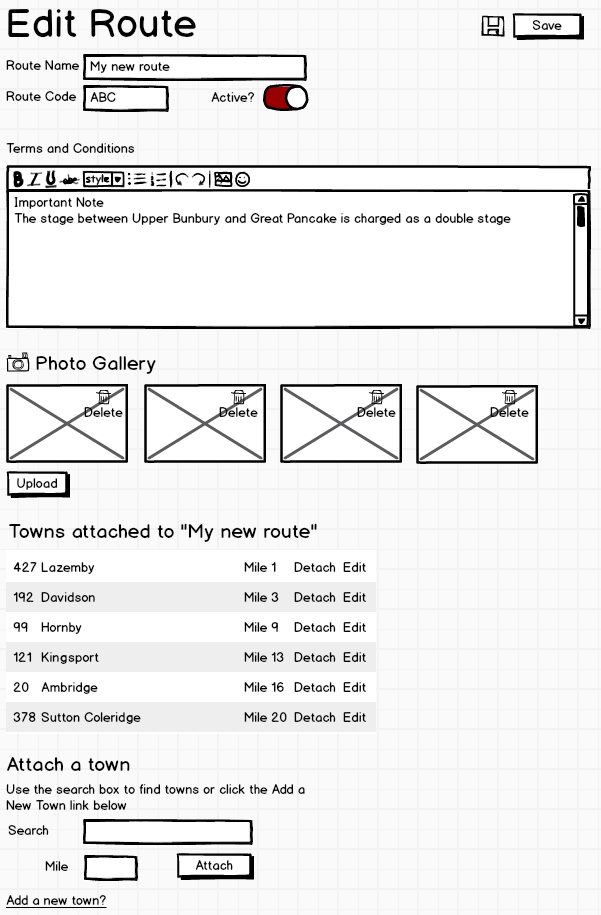
## Add Route



## Edit Route







# Town Admin

TABLE **towns** (pending changes)

`name` varchar(45) DEFAULT NULL,

`town\_id` int(11) NOT NULL AUTO\_INCREMENT

TABLE **route\_town\_link** (pending changes)

`route\_town\_link\_id` int(11) NOT NULL AUTO\_INCREMENT,

`route\_Id` int(11) NOT NULL,

`milepost` int(11) NOT NULL,

`description` text,

`town\_id` int(11) DEFAULT NULL,

PRIMARY KEY (`route\_town\_link\_id`)

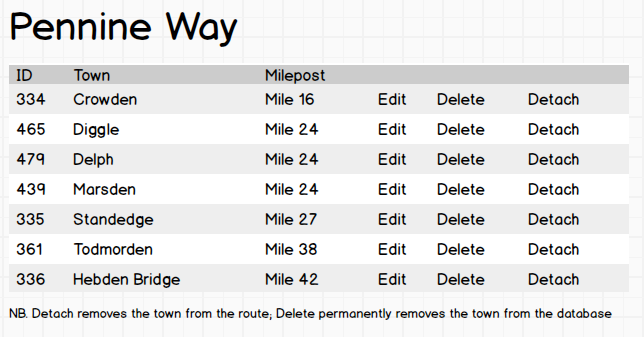
PRIMARY KEY (`town\_id`)

a system for adding, editing and deleting information about "towns", including the option to upload one or more images, description and other custom values, such as geolocation.

Town name and ID is consistent across all routes. However, a town description and milepost is different depending on the route.  
For example, town\_ID 14, Keld, is at mile 120 of the Pennine Way and mile 26 of the James Herriot Way. In many cases, the town description may be identical, but it might differ. For example, Keld has a different description for the James Herrot Way and Coast to Coast routes.

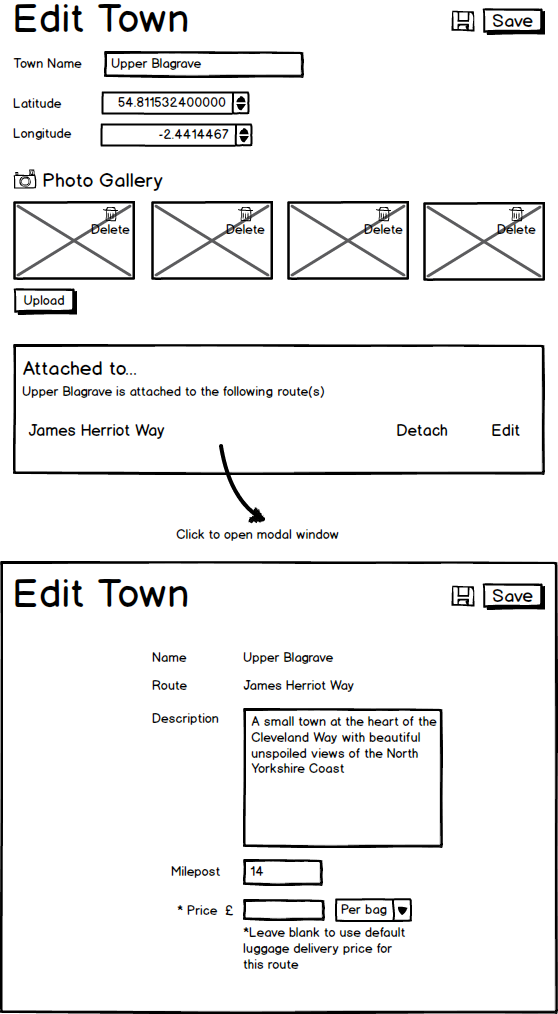
milepost and description are therefore stored in the route\_town\_link table, not the towns table.

## List Towns



Clicking the town name would take you to a lost of hotels in that town on that route.

## Add / Edit Towns



# Hotel Admin

TABLE **hotel\_route\_link** (pending changes)

`hotel\_route\_link\_id` int(11) NOT NULL AUTO\_INCREMENT,

`route\_id` int(11) DEFAULT NULL,

`date\_modified` datetime DEFAULT NULL,

`modified\_by` int(11) DEFAULT NULL,

`code` varchar(12) DEFAULT NULL,

`distfrompath` varchar(32) DEFAULT NULL,

`hotel\_id` int(11) DEFAULT NULL,

`town\_id` int(11) DEFAULT NULL,

PRIMARY KEY (`hotel\_route\_link\_id`)

TABLE **hotels** (pending changes)

`hotel\_id` int(11) NOT NULL AUTO\_INCREMENT,

`old\_id` int(11) DEFAULT NULL,

`code` varchar(10) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`route\_id` int(11) DEFAULT NULL,

`hotel\_name` varchar(100) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`description` text COLLATE utf8mb4\_unicode\_ci,

`location\_description` text COLLATE utf8mb4\_unicode\_ci,

`email` varchar(128) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`phone1` varchar(16) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`phone2` varchar(16) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`website` varchar(512) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`address1` varchar(128) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`address2` varchar(128) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`postArea` varchar(128) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`county` varchar(22) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`country` varchar(22) COLLATE utf8mb4\_unicode\_ci DEFAULT 'United Kingdom',

`postcode` varchar(9) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`fax` varchar(16) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`proprietors\_name` varchar(45) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`osref` varchar(16) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`expiry\_date` datetime DEFAULT NULL,

`date\_modified` datetime DEFAULT NULL,

`modified\_by` int(11) DEFAULT NULL,

`property\_type` varchar(45) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`live` int(11) DEFAULT NULL,

`picture` varchar(500) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`otherinfo` varchar(1000) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL,

`town\_id` int(11) DEFAULT NULL,

PRIMARY KEY (`hotel\_id`)

Valid property\_type list:

Not specified

De Luxe Hotel

Country House Hotel

Country Inn (Pub)

Budget Hotel

Guesthouse

B & B

Farmhouse B & B

Youth Hostel

Hostel

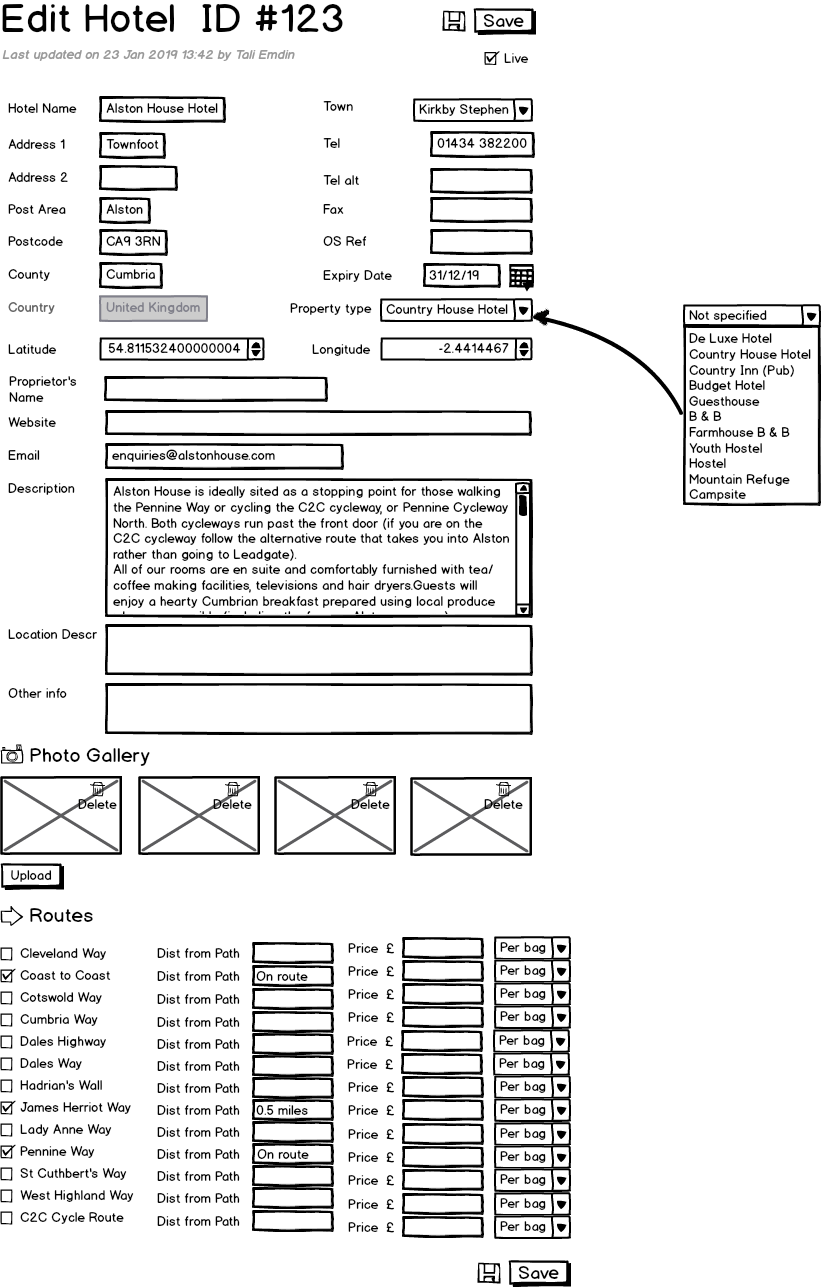
Mountain Refuge

Campsite

A system to add, edit and delete "hotels" (including guest houses, B&Bs and hostels), with the option to upload one or more images, description and other custom values such as email address, telephone number and website URL.

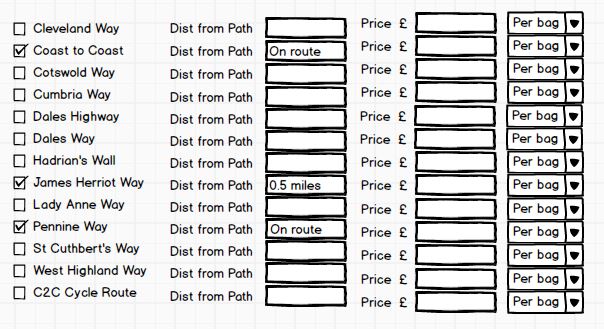
Hotels can be located by search or you will navigate to a list via the route and town. Hotels in a town that are not linked to the route will be flagged as such and there could also be a toggle switch to detach that hotel from that town on that route. On that page you will also be able to click on a hotel name and then edit the other details of that hotel. Intuitive and simple.

## Edit Hotel



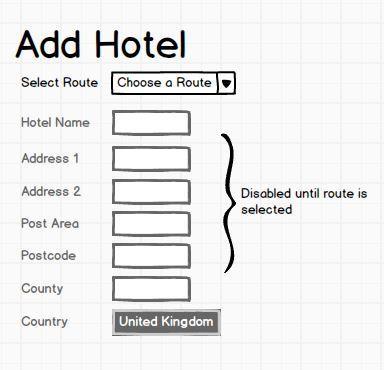
Some towns are on multiple walking routes. For example, Keld (town\_ID 14) is on the Pennine Way, James Herriot Way and Coast to Coast.

Luggage transfer price will be the default for the route/town unless specified

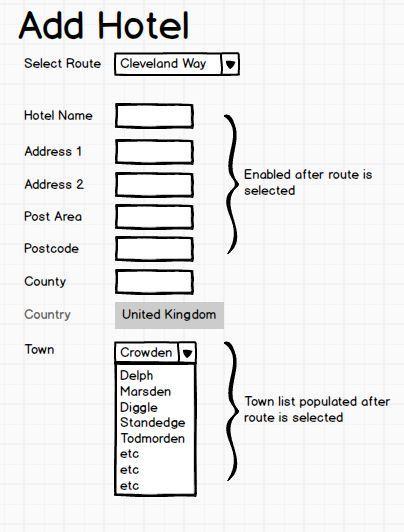


So, for each Hotel add/edit form, we need checkboxes to identify which of the routes is relevant to the hotel. Usually all hotels in every town on a route will be checked, but sometimes we might not want to show one (ie. in larger towns, the two routes that pass through might not be close to one another, so a hotel might be convenient for one route but not another).  
The relationship between routes and hotels is stored in the table hotel\_route\_link  
This query will return a list of live hotels for a given route/town (in this case, Crowden on the Pennine Way):  
select \* from hotel\_route\_link, hotels where hotel\_route\_link.hotel\_id = hotels.hotel\_id and hotel\_route\_link.town\_id = 334 and hotel\_route\_link.route\_id = 10 and live = 1 group by hotels.hotel\_id  
(You may change some of the field names to make them CamelCase)

When adding a new Hotel, it will be necessary first to choose a route, so that the towns select list can be populated.  
Note that a hotel can only be in one town, so (in addition to the address, which is not used for creating itineraries), there will be a select list in the add/edit form which includes every town on the route.



To be clear, that's not a button marked "United Kingdom" - it is a disabled text input!



distfrompath is different to milepost - this field is used to indicate whether a hotel is on the route or nearby. Some hotels are a mile or two away from the walk and the hotelier will offer a free pick-up service to collect walkers from the route. Some "hotels" are really just nearby farms with a converted barn or a couple of spare bedrooms, so they often offer this service. So the value of "distfrompath" might typically be something like "0.5 miles".

The "hotel\_route\_link" table requires a "live" field. Without it, if an administrator wants to make a hotel "not live", this would mean that if (s)he later on decides to make it "live", (s)he will need to re-enter the information we store in that table (ie. "dist\_from\_path" and any price that overrides the default price).

# Pricing

A back office pricing system to set hotel booking fees based on number of days and number of walkers

**TABLE** parties

`party\_id` int(11) NOT NULL AUTO\_INCREMENT,

`itinerary\_id` int(11) NOT NULL,

`user\_id` int(11) NOT NULL,

`first\_name` varchar(45) NOT NULL,

`last\_name` varchar(45) NOT NULL,

`nationality` varchar(45) DEFAULT NULL,

`DoB` date DEFAULT NULL,

`travel\_type` int(11) DEFAULT '1',

PRIMARY KEY (`party\_id`),

KEY `itinerary\_id\_idx` (`itinerary\_id`)

"travel\_type" will be 1 (walking) or 2 ("cycling") - this will be required for calculating prices as cyclists may be charged different fees. For example, Hadrian's Wall is charged at £8.50 per bag per move for walkers or a fixed fee of £45 per bag for cyclists for the whole route. Default will be 1 on all routes except C2C Cycle route which will always be 2.

For booking fees, the pricing for baggage transfers will have a default, based on the fee for a single bag, so that could be stored in the routes table. The routes table could also store the default minimum number of bags per party. Alternatively, instead of price per bag, there will be a fixed fee which will include a maximum number of transfers and a maximum number of bags.

For instance, the C2C Cycle Route http://www.sherpavan.com/baggage/traildivert.asp?trailmenu=C2C has a £100 charge per bag for up to five transfers.

Then we need to be able to override prices for selected baggage transfers in a table that connects the hotel with the route. We already have hotel\_route\_link, so that might be a good place to include it. In practice, the prices that need to be changed are usually for every hotel in a particular town, rather than for the hotels specifically. The reason for the surcharges is generally because the town takes additional time to reach, so usually all hotels in the town will be affected.

A three-tier hierarchy might be best. So you would have as default price per bag per transfer (routes table) and then an override price in the route\_town\_link table and then a third override price in the hotel\_route\_link table. So on the Pennine Way you would have default price £9.50 per bag per move but all hotels in Edale on the Pennine Way would be £17.50 per bag per move and if there was a particular hotel that was hard to reach (eg. down a long, muddy country lane), that hotel could have a small surcharge to make it £20.00. In 99.9% of cases the override price in hotel\_route\_link would never be used, but it would be a useful option.

Example exceptions: see Pennine Way http://www.sherpavan.com/baggage/traildivert.asp?trailmenu=PW

You could say that £9.50 was the default, but there will be different prices for all the hotels between Edale and Malham inclusive (the first 78 miles of the route) and all hotels in all towns between Alston (milepost 180) and Greenhead (milepost 200) and all hotels in all towns between Byrenss (milepost 238) and Kirk Yetholm (the end of the route - milepost 266)

The actual list of different prices on the Pennine Way is:

Edale to Malham: £17.50 per bag per move

Malham to Alston: £9.50 per bag per move

Alston to Greenhead: £28.00 for 1 – 4 bags per move

Greenhead to Byrness: £9.50 per bag per move

Byrenss to Kirk Yetholm: £20.00 per bag per move

### Luggage Transfer Pricing

Price per bag with a minimum number of bags per party

eg. Coast to Coast is £8.50 per bag, minimum 1 bag per person in the party. So for a party of two, the minimum daily charge will be £17.00.

Fixed price for a maximum number of luggage transfers

Double fee for transfers of more than 30 miles

eg. on the Coast to Coast Kirkby Stephen (milepost 83) to Reeth (milepost 106) would be the default £8.50 per bag but Kirkby Stephen to Richmond (milepost 117) would be £17.00 per bag

Fixed fee per bag for a maximum number of luggage transfers

eg. C2C Cycle Route is £100 per bag for up to five luggage transfers

The basic pricing setting will set the cost of a single baggage transfer on a route. But we need to create price packages with exceptions for specific transfers (eg. Great Langdale to Keswick on the Cumbria Way is charged double) and practical limitations such as minimum and maximum bag numbers per person and/or per party. This will allow us to create fixed-price packages (eg. Sea to Sea Package Deal) with a maximum number of transfers for a fixed-price package.

### How it Will Work

Every route will have a default price for transfers based on 1 bag and 1 transfer and for some routes this will suffice.

In addition, via the Dashboard we can create a price package:

We have 4 parts for price packages: route, number of bags, number of transfers and price or, for selected towns, 5 parts: route, town, number of bags, number of transfers and price.

For instance, on the C2C Cycle Route the package comprises (1) Route: C2C (2) No. Bags: 1, (3) No. Transfers: 5, (4) Price: £100.

The default price will be overridden by a package price if the itinerary meets the criteria for the package.

## Hotel Priorities

Hotel listings will have at least two priority levels, with paid listings appearing at the top of the list (where there is more than one priority listing for a town, top positions will be assigned randomly).

Initially all hotels will have the same priority, so we won't be using this in practice when the site launches. I suggest setting the default value in the table to 5 and then run an update query to assign 5 to all rows in the table. We will then have a priority to play around with in a future phase of development.

Expired hotel listings may change in appearance (not yet finalised).

Specified in the hotels table by priority (int) and expiry\_date.

# Bookings/Customers (Dashboard)

A back office system to view and edit completed and incomplete customer itineraries.

List of bookings, paginated, with options to sort by various columns + delete (with prompt) and view options. Default would be most recent booking first. "Confirmed" highlighted in green for itineraries that have been completed and paid for.



Click View to open **View Booking** page

Page showing all details of the customer's contact information and choices.



# Stats

Stats page showing a table of completed and incomplete itineraries per day with a graph across a specified date range.

Graph of page statistics according to a specified date range

System to log number of visits per page, per route and per town per day

Stats report of hotels sorted by number of email enquiries logged

Back office daily email stats report showing number of bookings, top page visits for the day and number of hotel email enquiries

# Data Import

One-time import all current town, route and hotel data ready for editing (currently stored in a single MS Access file)

(Already completed, but revisions to town data by Tali, Elliot and their team may need re-importing).

# Customer Account

Customer accounts - register, log in, reset password, edit profile, view future and past itineraries. See also [Users](#_Users).

# Town pages

A page for every town on every route with SEO-friendly URL, title tag and meta description.

URL in the following format: *https://www.sherpavan.com/coast-to-coast/st-bees*

[domain][route name][town name]

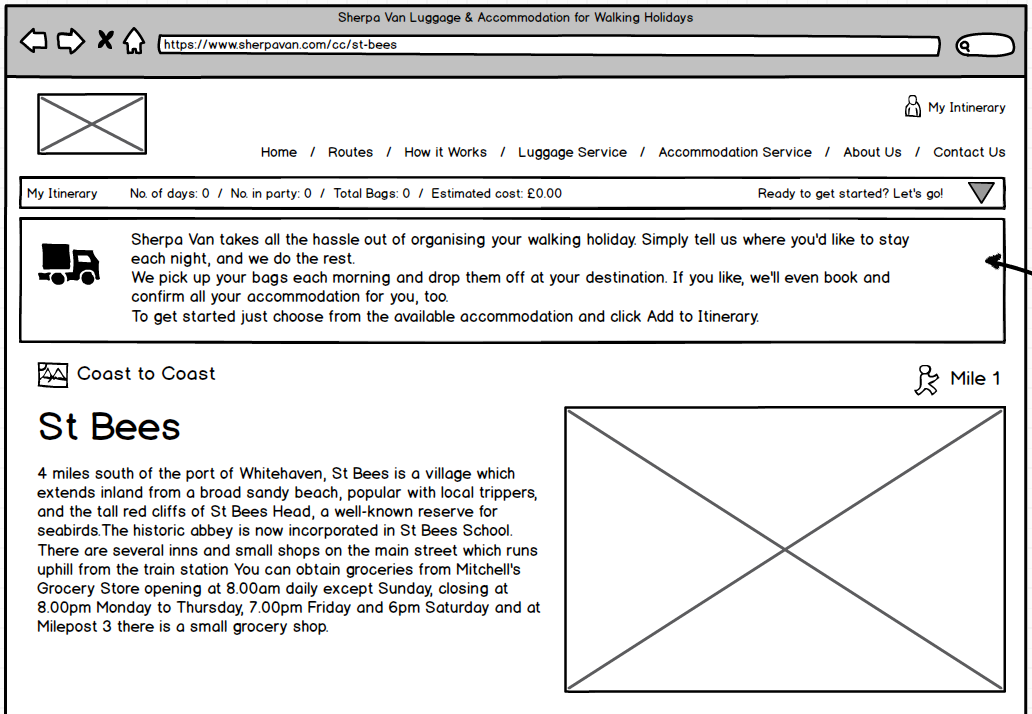
Town and route names with spaces to be replaced with spaces and other punctuation (eg. brackets, apostrophes) to be omitted,

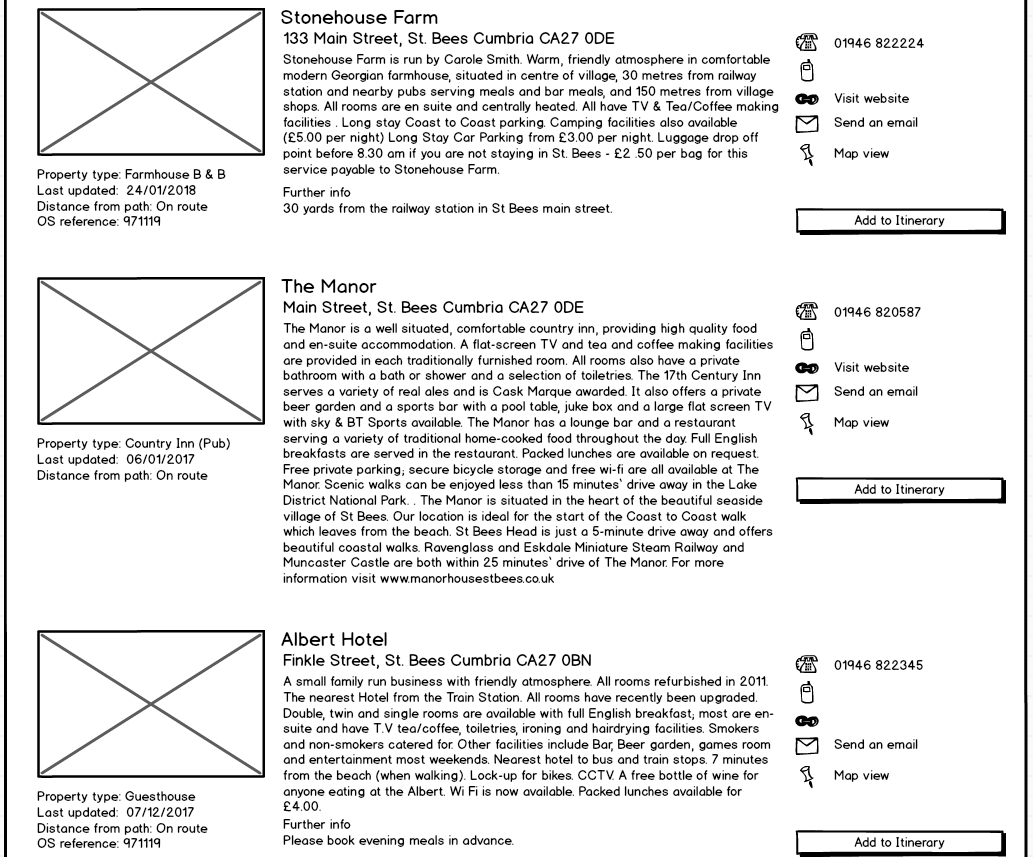
Town pages will contain a "Status Panel" at the top.

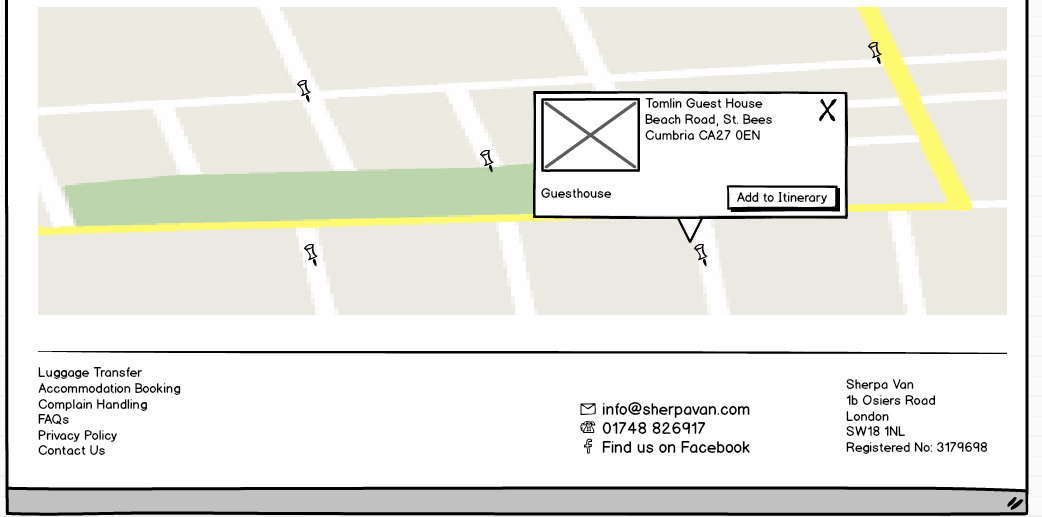
If you haven't started an itinerary, that panel contains a short explanation of how Sherpa Van works.

When you start an itinerary, the Status Panel contains a wizard that leads you through the steps required to create the itinerary. First you choose between (a) Luggage Transfers Only and (b) Luggage Transfers & Accommodation Booking. Second, you supply information relating to Luggage Transfers. Third, if you selected Accommodation Booking, you supply information about your room preferences. Fourth, you select the arrival and departure date. Fifth, you see a summary of your itinerary and select the next town you would like to stop at.

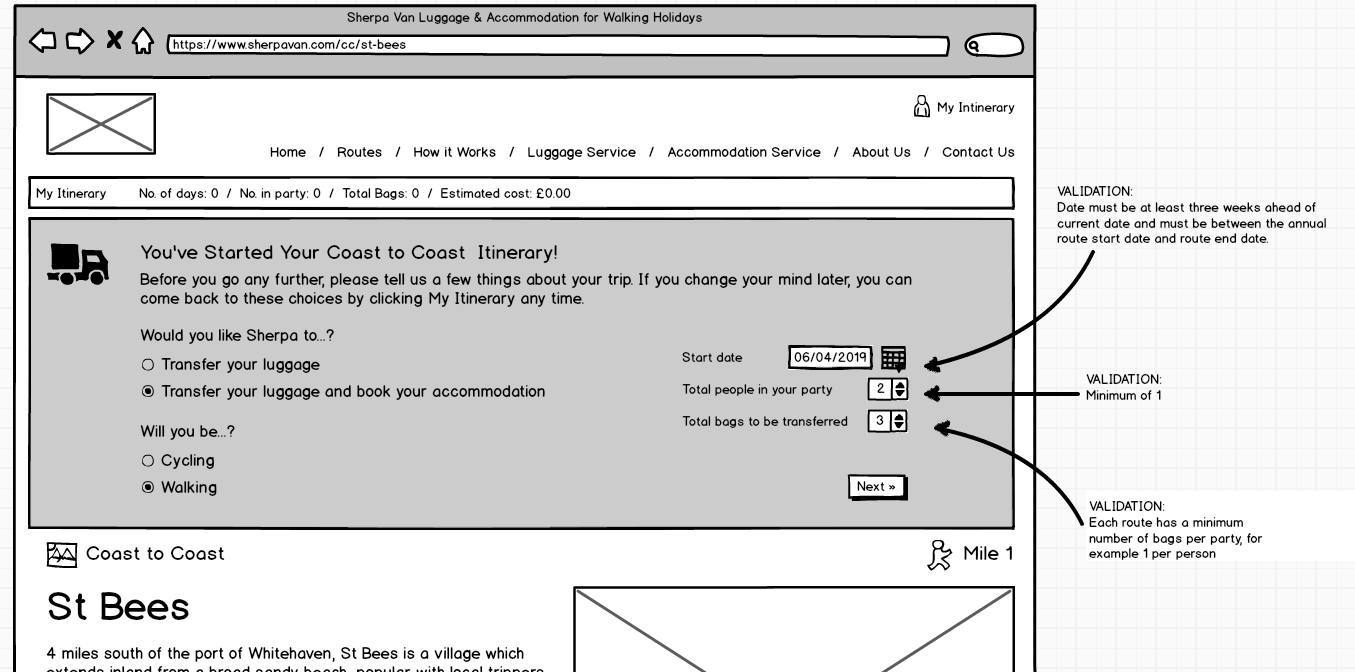
If you then select the next town and choose another hotel to add to your itinerary, you will skip straight to step four (arrival and departure dates) and so on as you build your itinerary. When you are ready to save your itinerary, you will be prompted to log in or register.



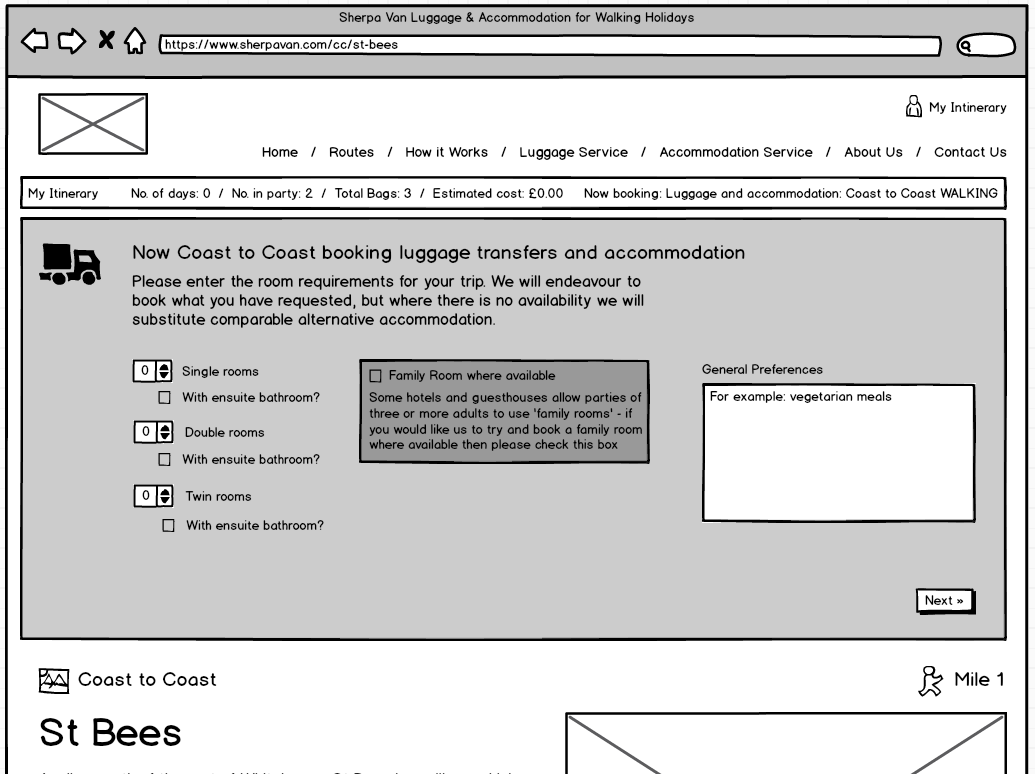




The following image shows what you might see when you click a link to start an itinerary, whether you require just luggage booking or both luggage and accommodation.



The following image shows the final questions required only for customers who require hotel accommodation



Hotel listings will have minimal data in order to reduce the admin cost of keeping the data up to date, but will include an image slideshow of multiple photos in some cases. There will also be a reCaptcha-protected enquiry form and a stat should be logged for each validated email enquiry. An itinerary will be validated based on logical restrictions, such as maximum distance per transfer, and practical restrictions, such as the season date range (eg. 24 March to 13 October) and direction (ie. to optionally allow walkers to take the reverse Hadrian's Wall route)

We will set up 301 redirects from all old URIs to the most relevant new ones

For example, from

*http://www.sherpavan.com/accomm\_booking/queryhotels.asp?trail=CC&maparea=&location=St.\*Bees*

to

*https://www.sherpavan.com/coast-to-coast/st-bees*

# Content Management

A simple, template-driven content management system for marketing and information pages (example from the current live site: <http://www.sherpavan.com/trails/hadrians-wall.asp>), home page, menus, header and footer and news posts. It will be easy to use for non-technical people

## On the Admin Dashboard

### Create/edit/delete new pages

A page will need these properties (off the top of my head): ID, title, meta\_description, template, unique\_filename, category/categories, live, date\_published, date\_modified, created\_by, content (where the content will include HTML/images). The content would be entered using something like Quill https://quilljs.com/

In order to be versatile, it will be necessary to have multiple blocks of content so that some templates can split content into different parts of the page.

A page will also need multiple images. I'd welcome advice on the best and most flexible way to achieve this.

TABLE **pages**

page\_id

pub\_date

page\_title

page\_excerpt

live

url

template

date\_created

created\_by

date\_modified

modified\_by

**TABLE** content

content\_id

page\_id

html\_content

priority

TABLE **images**

image\_id

page\_id

img\_src

priority

This would allow for a single page with multiple content blocks and multiple images. We would use the priority field in the content and images tables to correspond to variables in the template. So you have a template with <?=$html\_content1?> <?=$image1?> <?=$ html\_content2?> <?=$image2?> where $html\_content might also include inline images.

We also need to do something similar with sidebar/header/footer content. So we could maybe have a table called widgets that is the same as pages, where the HTML is in the content table and the pictures are in the images table and in the template it is just <?=$widget1?>.

I guess each template would need a function that uses the widget ID to grab content and then all of the other data for the page would relate directly to the page ID/url, so you could retrieve it with a single query. So that's basically just a couple of queries per page. 4 queries if you include the header and footer.

We would also need a categories table, plus a table linking categories to pages. Or you could have a categories field in the pages table with a comma-delimited list of categories.

### Create/edit categories

A category would have an ID, category\_name and unique\_filename

### Create/edit sidebars

Content blocks that can be shared between multiple templates, including headers and footers, allowing the user to change the content site-wide with a single edit.

Sidebar management would be much the same as page management - we would want to have a default footer and a default sidebar and then the PHP code would just have a single line of code to display "$footer". But we could then create a new sidebar block and give it a new name, "$CC\_footer" and then use that on the Coast to Coast page.

## On the website

Script to return a list of pages by date

Script to return a list of pages by category

### Pages for each of the following:

|  |  |
| --- | --- |
| **Route Summary**  Cleveland Way  Coast to Coast  Cotswold Way  Cumbria Way  Dales Highway  Dales Way  Hadrian's Wall  Herriot Way  Lady Anne Way  Pennine Way  C2C Cycle Route  St Cuthbert's Way  West Highland Way | **Editorial**  Routes  How it Works  Articles (news, editorial etc)  Luggage Service  Accommodation Service  About Us   * Press Cuttings * Employment Opportunities * Complaint Handling * Client Feedback * Privacy Policy   Contact  Competition |

Render pages created by admin users with a PHP template which will contain variables that can be replaced by text and images entered via the dashboard.

It will be simple to create new PHP page templates, either (a) without needing to update the database (possibly by filename or file contents) or (b) by a simple form to update database (ie. specifying a file name and template name). If we want a new page template specifically for content with the category "Competitions", we will be able to make a copy of an existing template and simply change the category name at the top of the file. Then I would edit the page in the admin dashboard to indicate that it should be displayed with the "Competitions" template.

Map View. Google Maps integration yet to be finalised. For example, for each town/village, we could plot the local hotels on a Google Map and for a customer's itinerary we would show all of the hotels they have chosen on a Google Map.

Possible approach: full-page Google Map showing hotels with the highlighted hotel in a side panel - like the Rightmove website: https://goo.gl/oG8pGH - suggested after the job was quoted, so need to investigate feasibility of doing this within budget

# Checkout

An ecommerce shopping basket system to display the customer's costs as they are accrued

A secure checkout process for customer payments, including email confirmations to customer and back office.

A back office list of hotels approaching their priority expiry date and a list of recently expired priority hotels

# Database Diagram

Work in progress, Jan 2019

