**PAGE 1**

The specifications for my application are fairly simple…

* The ability for a user to post a ride from A to B (and possibly to C, D etc) of which they have spare seats
* The ability to search for rides for users who would like to hitch a ride
* The ability to request to hitch a ride
* To accept/decline the hitcher

Other services such as Carpooling.com do provide this simple service from A to B, but that is as advanced as it goes. The are no partial pick ups on the way unless specifically detailed by the driver when they register a journey.

I propose a more complex way of looking at the problem. For example…

**PAGE 2**

* If a user is driving from Aberystwyth to Cardiff and has a spare seat, they would post a journey to the website such as this. My database stores the journey data.
* If another user searched for a ride between Aberystwyth and Cardiff, this ride would show up on current available services as well as mine. However, if a user searched for a ride between Builth Wells and Cardiff (or similarly: Aberystwyth to Builth Wells) on current services it would not show up. On my service, because it is along the route, it would also show up as a partial journey…

**PAGE 3**

* If the user wants to be picked up in Builth Well and driven to Cardiff, they would request a hitch to this journey and it is up to the driver to decide whether they wish to pick them up.
* Similarly, if a user search for rides between Builth Wells and Aberdare, this journey would also show up…

**PAGE 4**

* Because Aberdare is close to the route, the search would provide the hitcher with the option of requesting a hitch on this journey.

**PAGE 5**

* Person hold personal data about a user. This includes their hometown, frequent destinations, etc for the system to suggest journeys to them.
* Journey table hold the basic data about a posted journey. Origin, destination, driver, date and time, spaces avaiable, etc
* Journey Step table hold the more detailed steps of the journey. It stores the geographical locations of each of the steps it takes to drive from origin to destination of the related journey. This is what allows partial journey search. This data is retrieved from the google directions API.
* Hitch Request Table stores data which relates to one user wishing to hitch one of the journeys in the journey table. It is subject to the journey drivers decision to accept or decline the hitch request. The request may include waypoints, which means that the journey may have shown up searching as a partial journey (as explained earlier). In this case, is the driver accepts the hitch request, the Journey Steps may be subject to change with the new Waypoint.
* Other tables are present such as messaging, temporary step table for altering the journey step and a few others but main project outcomes involve these tables

**PAGE 6**

* A server side API built by myself in PHP manages the database and the retrieval of information from APIs etc when data is posted to it via HTTP POST from the website.
* The database stores all user, journey and hitch data.
* Geolocating of place names and direction data is pulled by the php api when new journey data is being inserted or current journey data is altered
* When the website is complete, it will use a range of javascript and HTML techniques to display users information.

**PAGE 7**

* PHP is powerful language but not known to be a particularly fast one.
* Using an execution engine such as HipHop to increase speed of PHP.
* Shortcuts like using journey steps from existing similar journeys by other users instead of requesting more data from API.
* Suggesting juorneys to users when they log in using search algorithms which use their data and preferences to predict if they may want to hitch a ride somewhere