General Trip management  
  
Trip management is kernel features of a carpool system. Generally, the drivers’ operations on a trip include selecting passengers who would like to travel to the destination in addition to creating, modifying and canceling a trip entry. Passengers, on the other side, may join and quit a trip as well as inquire specific trip information of the driver through online chatting system or email.   
  
Create a trip

This function allows the driver to create a trip thread in which he or she will look for passengers to share the ride. A few mandatory fields, such as trip name, start date, one or multiple destinations, price, maximum number of passengers, car model etc., must be filled out on the “creating a trip” form. Once the driver submits the form, a trip thread will be created in the database and displayed on the website with public access.   
  
Modify a trip

This function allows the driver to add or modify the details of a trip he created. Email notification, which highlights the modified information and the reason to make changes, will be sent to all passengers if anyone has joined the trip. If no passengers have joined the trip yet, no email notification will be sent. All trip details can be changed by the driver.  
  
Cancel a trip

This function allows the driver to cancel a trip with a suitable reason in which an automatic email notification will be sent out to passengers who have joined the trip. The trip information will be permanently removed from the database and disappeared from the website. This function may result in bad rating against the driver if passengers are not satisfied with the reason to cancel the trip. Details will be illustrated in the rating system feature.  
  
Select passengers to join the trip

This function allows the driver to select passengers who would like to travel to the destination from a candidate list, possibly based on candidates’ reputation. The candidate list is ordered by descending reputation with the exception that candidates at the top of the list should be people from driver’s friend list. Besides reputation, driver can view candidates’ public personal information as well. Different email notifications will be separately sent to the passengers who have been picked and unsuccessful candidates if the maximum number of passengers is reached. The driver is free to alter the selection at any time. However, again, the driver may receive bad rating if he or she changes the selection too often.  
  
Join a trip

This function allows the passenger to apply to join a trip. Because carpool is a kind of bidirectional selection, passengers who are interested in a trip have to apply to join. Passengers need to indicate the payment method (i.e. PayPal or Cash) during the application in addition to some personal information. An email confirmation will be sent out to passengers who submit the application. After that, passengers will be pushed into the candidate list so that driver may select them from there. Both successful and unsuccessful applicants will receive the email notification eventually.   
  
Quit a trip

This function allows the passenger to quit the trip with a suitable reason in which an automatic email notification will be sent out to the driver. However, passengers are not allowed to quit a trip one day before the start date of the trip. Similar to canceling a trip, this function may possibly result in bad rating against the passenger if the driver is not satisfied with the final situation.   
  
Inquire additional trip information

This function allows the passenger to inquire additional trip information of the driver. There are multiple ways that can realize this point:

1. Website internal chatting: Under the circumstance that both driver and passenger are visiting the carpool system, they could use the website internal real-time chatting.  
2. Embedded Google talk (**potential security issue**): Driver may choose to attach his or her Google talk gadget on the trip thread so that passengers can chat with him when he is online.  
3. Email  
4. Phone

## List my trips

This function allows the user to view all the past and upcoming trips that he or she has been involved or will be involved in the near future. For each trip, some useful information, such as money saved, will be displayed in addition to all the basic information. Driver and passengers have the option to send a request to be friends so that they will receive the notification if their friends create any trips in the future. They will also be requested to rate each other in this panel once the trip is over. Details will be explained in the rating system and social networking sections.

# Feature Enhancements

## Cross-browser compatibility

This pertains to an extra feature of our system as a whole. Cross‐browser compatibility can often be found an issue with the use of CSS. Extra code and method calls will be used to bypass this otherwise inefficient viewing issue. We plan to allow the user to interact on one browser (i.e. Firefox) with another user on a different browser (i.e. Internet Explorer) as well. It seems only fair that we do this if we support viewing of our system at all on multiple browsers.

## Newsfeed

This pertains to an extra feature of our main page. The expandable widget, displayed after login, will have a list of notifications based on the actions of their friends, such as creating new trips, uploading a new picture, etc. It will also include any number of alerts about the changes of their upcoming trips. Traffic news will be displayed at the bottom as well.

## Real-time Gas Price

This pertains to an extra feature of our main page. Instead of using admin updated gas price, which is considered as a constant number, we add a real-time widget to retrieve the gas information from a service provider.

## Google Map

This pertains to an extra feature in order to visualize the trip information, like the distance, route, destination etc.

## Instant Messenger

This pertains to an extra feature of our website. Users can easily communicate with the trip driver via the internal chatting system to discuss the trip details. It’s an easy, fast, convenient and efficient way to make an inquiry of the driver.

## Notification System

This pertains to an extra feature to achieve some functional goals. An important use case of it is when user finishes submit the sign-up form, he or she will receive a confirmation email which requests to activate the account. In addition, any changes of upcoming trips will be notified through email. Some use cases will also be illustrated in other sections.

===================================================================

From Europa:

**Feature Enhancements**

Route Map

A route map will display on the trip information page using Google Map, based on the home, transfer stations and destination of the trip. Estimated total distance and travelling time will also display along with the route map to help interested users find an optimal choice.

Notification System

A user’s home page, displayed after login, will have a list of notifications for updates on their trips. These notifications will be sent automatically to relevant users when the trip information is modified or a trip is cancelled. It may also include notifications for confirmed friend requests sent from you. A user can control the types of notifications they want to receive/discard through \*\*\*Account Setting\*\*\* page.

Cross-browser Compatibility

This pertains to an extra feature of our system as a whole. Cross-browser compatibility can often be found an issue with the use of CSS. Extra code and method calls will be used to bypass this otherwise inefficient viewing issue. The team plan to allow user to interact on one browser with another user on a different browser as well. It seems only fair that we do this if we support viewing of our system at all on multiple browsers.

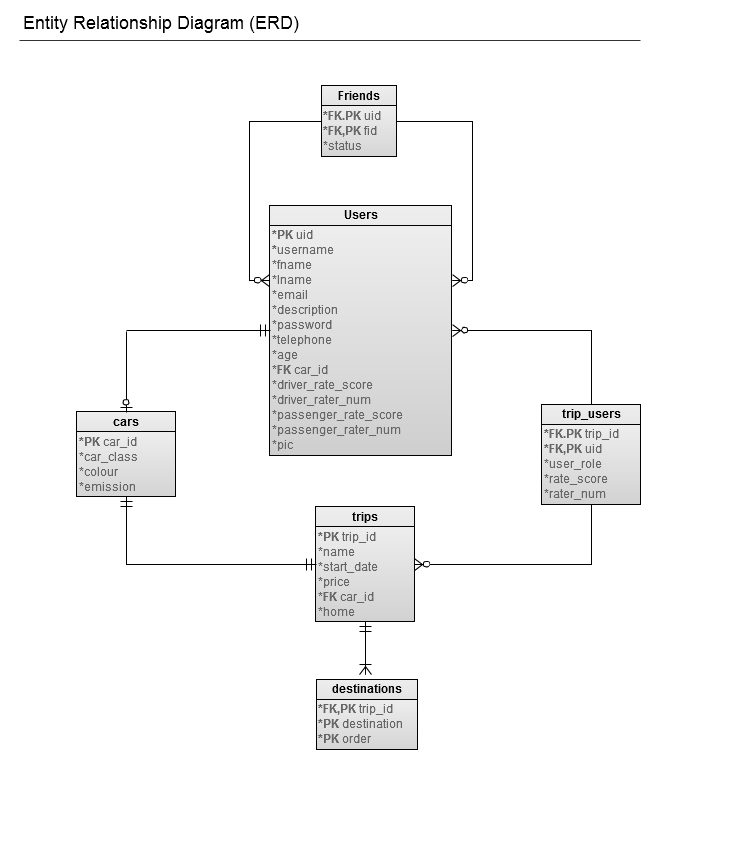
Instant Message

A Chat with Me button can be accessed on the user’s profile page if the user allows our connection to his/her Google Talk. Users can contact with each other and exchange trip information whenever they want as long as they both have Google Talk account.

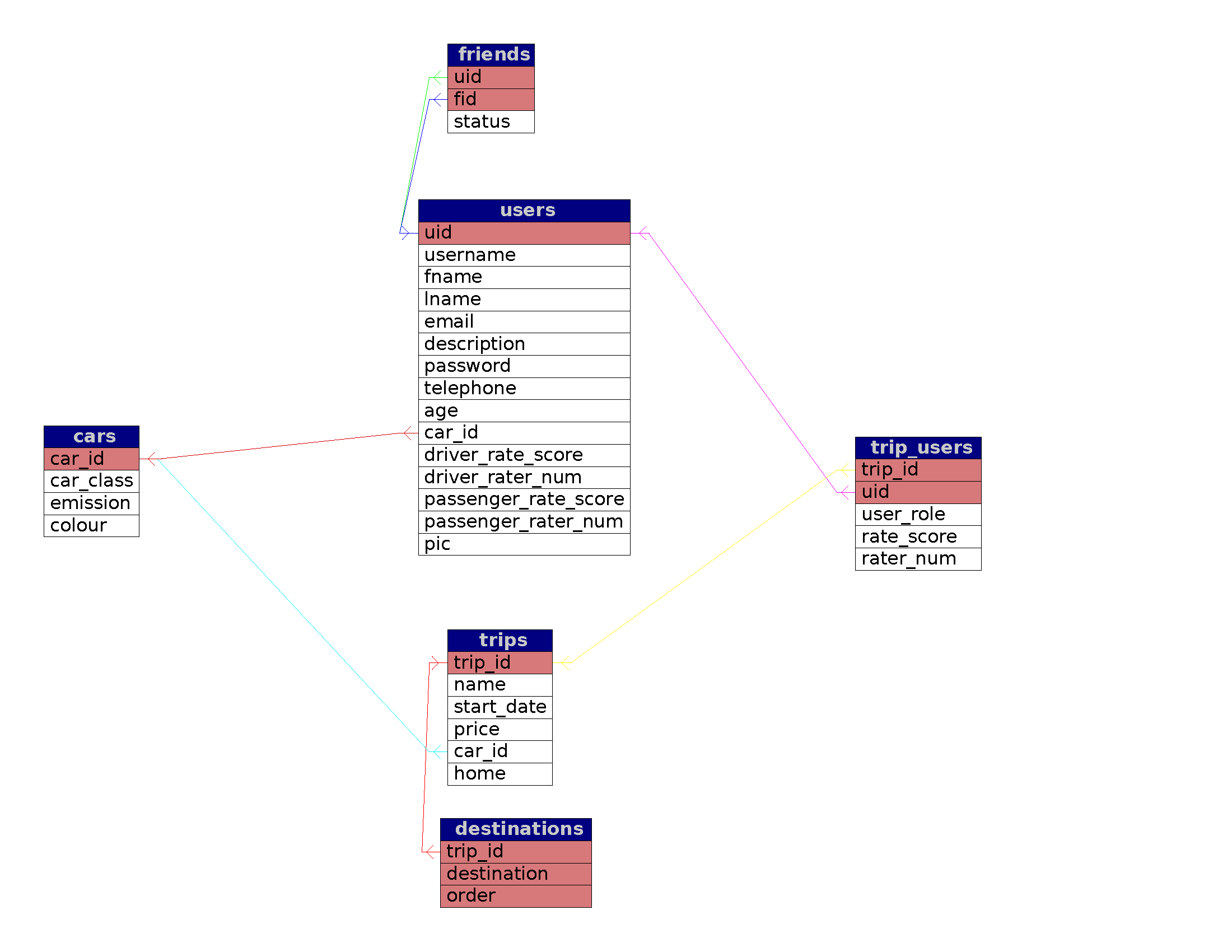
# Information Representation

The Database Management System (DBMS) that will be used is MySQL as it is among the most widely used DBMSs as well as the most popular open source DBMS because of its ease of use, fast performance and reliability. We also use Phpmyadmin as our frontend tool to handle the administration of MySQL for convenience.

## ER Diagram



## Data Schema



Cars

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| car\_id | int(11) | No |  | auto\_increment |  |
| car\_class | varchar(16) | No |  |  |  |
| emission | int(11) | No |  |  |  |
| colour | varchar(16) | No |  |  |  |

Destinations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| trip\_id | int(11) | No |  |  | trips.trip\_id |
| destination | varchar(128) | No |  |  |  |
| order | int(11) | No |  |  |  |

Friends

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| uid | int(11) | No |  |  | users.uid |
| fid | int(11) | No |  |  | users.uid |
| status | int(11) | No | 0 |  |  |

Trip\_users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| trip\_id | int(11) | No |  |  | trips.trip\_id |
| uid | int(11) | No |  |  | users.uid |
| user\_role | varchar(16) | No |  |  |  |
| rate\_score | float | No | 0 |  |  |
| rater\_num | int(11) | No | 0 |  |  |

Trips

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| trip\_id | int(11) | No |  | auto\_increment |  |
| name | varchar(64) | No |  |  |  |
| start\_date | date | No |  |  |  |
| price | int(11) | Yes | NULL |  |  |
| car\_id | int(11) | No |  |  | cars.car\_id |
| home | varchar(128) | No |  |  |  |

Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Default | Extra | Links to |
| uid | int(11) | No |  | auto\_increment |  |
| username | varchar(32) | No |  |  |  |
| fname | varchar(16) | No |  |  |  |
| lname | varchar(16) | No |  |  |  |
| email | varchar(64) | No |  |  |  |
| description | text | Yes | NULL |  |  |
| password | int(150) | No |  |  |  |
| telephone | varchar(16) | Yes | NULL |  |  |
| age | smallint(6) | Yes | NULL |  |  |
| car\_id | int(11) | Yes | NULL |  | cars.car\_id |
| driver\_rate\_score | float | No | 0 |  |  |
| driver\_rater\_num | int(11) | No | 0 |  |  |
| passenger\_rate\_score | float | No | 0 |  |  |
| passenger\_rater\_num | int(11) | No | 0 |  |  |
| pic | varchar(255) | No |  |  |  |

# Interaction with environment

In the case of interaction with browsers, our carpool system aims to be compatible with the browsers we think to be of use in the mainstream public, including Mozilla Firefox 4.0, Google Chrome 12, Internet Explorer 6/8/9, and Opera 11. We’ve considered supporting mobile browsers, but currently we are focused more on internet capabilities. As three mainstream mobile operating systems, iOS, android and windows phone 7 have very excellent support on website browsing, we believe it will not be a problem to run our carpool system on smartphone or tablet.

Catering to the notion that we aim to have a marketable product by the end of this project, our service will properly operate on Windows, Ubuntu and Mac OS, the three leading operating systems in the market right now.

As two main enhancements in our carpool system, Instant Messenger and Notification System will be developed based on an open source product PHP Free Chat and PHP Extension PEAR-Mail. The PEAR-Mail package will enable us to use gmail account to send out the notification email as SMTP port 25 is commonly blocked by the ISP which prevents us from building our own email server.

As for our choice of web server, the Apache HTTP Server is the most popular web server in current use and because we are familiar in its use, the choice was clear. Apache Version 2.2.17 is the one running on our server and we plan to use it in tandem with PHP scripts. In addition, for security purpose, we will enable the SSL channel to protect the connection, because account information may be processed in our system.

# High Level System Architecture

**System Architecture**

At the high level, the system will follow the standard model‐view‐controller architecture in the manner described below. In addition, the technology/language selected for each implementation is provided.

**Model ‐ (Database ‐ MySQL)**

Our database will be the backend of the system whose sole responsibility is that of data storage and retrieval. MySQL will provide us the required database management system, and will run as a server, allowing multi‐user access.

**View ‐ (Webpage ‐ HTML, PHP, CSS)**

In essence, the user interface of our project. The webpage will be responsible for laying out information in a visually pleasing manner, and providing information to the controller and model as required. A subset of the provided technologies will be used.

**Controller ‐ (Editor/Webpage – AJAX, Ext-JS)**

The controller works as a liaison between view and model aspects of the system. It takes required inputs, performing whatever manipulations are necessary and stores that information as required. Ext-JS is a JavaScript library for building interactive web applications, including a set of GUI-based form controls for use within the web applications. The team will use it as an extension of AJAX to provide the functionality required from the webpage.