Hitch a Ride

On-line Carpooling Service

Final Report for CS39440 Major Project

Author: Alexander Roan (alr16@aber.ac.uk)
Supervisor: Fred Labrosse (ffl@aber.ac.uk)

4th April 2014
Version: 1.0 (Draft)

Abstract

Hitch a Ride is an on-line carpooling service which enables people to share there journeys across the country with the aim of saving money and fuel.

The service encourages drivers to post the details of prospective journeys in the hope that hitchers will request to join them on their travels in exchange of a shared fuel price.

Something doo doo blah blah blah....

Contents

1 Introduction **4**
- 1.1 Overview . . . 4
- 1.2 Example Use Case . . . 4
- 1.3 Existing Services . . . 5
- 1.4 How This Compares . . . 5

2 Objectives **5**
- 2.1 Original Goals . . . 5
- 2.2 Requirements . . . 5
  - 2.2.1 Interface Requirements . . . 5
  - 2.2.2 Functional Requirements . . . 5
  - 2.2.3 Database Requirements . . . 5
  - 2.2.4 Software Requirements . . . 5
  - 2.2.5 Performance Requirements . . . 5

3 Development Process **5**
- 3.1 Overview . . . 5
- 3.2 Methodology . . . 5
- 3.3 Planning . . . 5
- 3.4 Prototypes . . . 5
- 3.5 Research . . . 5

4 Design **5**
- 4.1 Overview . . . 5
- 4.2 Technologies . . . 5
  - 4.2.1 PHP . . . 5
  - 4.2.2 JQuery . . . 5
  - 4.2.3 JSON . . . 5
  - 4.2.4 PSQL Database . . . 5
  - 4.2.5 Github . . . 5
- 4.3 Service Infrastructure . . . 5
  - 4.3.1 University Hosting . . . 5
  - 4.3.2 LAMP Server . . . 5
- 4.4 Service Design . . . 6
  - 4.4.1 Overview . . . 6
  - 4.4.2 Method . . . 6
  - 4.4.3 Structure of Database API . . . 6
- 4.5 Website Design . . . 6

4.5.1 Overview 6
4.5.2 PHP 6
4.5.3 Bootstrap 6
4.5.4 JavaScript and JQuery Library 6

5 Implementation **6**
5.1 Overview 6
5.2 PHP 6
5.3 Google Directions API 6
5.4 Database 6
5.5 Database API 6
5.6 Website 6

6 Testing **7**
6.1 Overview 7
6.2 Database API Testing 7
6.2.1 Overview? 7
6.2.2 Unit Tests 7
6.2.3 Functional Tests 7
6.3 Website Testing 7
6.3.1 Overview 7
6.3.2 Functional Tests 7

7 Evaluation **7**
7.1 Original Goals 7
7.2 Accomplishments 7
7.3 Future Improvements 7
7.4 Future Development 7
7.5 Design Choices 7
7.6 Approach 7

8 Bibliography **7**

1 Introduction

1.1 Overview

1.2 Example Use Case

When first opening the home page of the site, the user will be confronted with a splash page, with the option of either registering, or logging in using existing details. Either action done successfully will log the user in.

Once logged in, the user is taken to their home page, which displays predicted journeys that the site thinks the user might be interested in using their preferences. From this page, they can access all features within the site, which include:

- 'My Activity' - Information about upcoming rides and hitches related to the user. The ability to accept or decline hitch requests for their journeys.
- 'Messages' - Messages to and from other users.
- 'My Profile' - Personal details about the user, available for editing.
- 'Find a Journey' - A search page used to search for journeys from one location to another.
- 'Post Journey' - The page used to post new journeys that the user will be partaking in.

When a user posts a journey from location A to location B using the 'Post Journey' page, it will display a map route which needs to be accepted by the user. Once this is accepted, the journey will be entered into the database, where it will be returned in other users searches if the parameters match the details of the journey.

If another user requests to hitch a ride on the journey, the driver will be prompted in the 'My Activity' page, and will have the option of accepting or declining the hitch request.

The hitch request could be from location A to location B, but it could also be from location A to a new location C. Similarly it could be from location C to location B, or even location C to location D. It is up to the driver to accept or decline the hitch request, depending on how it changes the route that the driver takes.

If the driver accepts this new hitcher, the route will be altered if necessary to include the new pickup / drop off points and saved into the database with one of the spare spaces now filled.

1.3 Existing Services

1.4 How This Compares

2 Objectives

2.1 Original Goals

2.2 Requirements

2.2.1 Interface Requirements

2.2.2 Functional Requirements

2.2.3 Database Requirements

2.2.4 Software Requirements

2.2.5 Performance Requirements

3 Development Process

3.1 Overview

3.2 Methodology

3.3 Planning

3.4 Prototypes

3.5 Research

4 Design

4.1 Overview

4.2 Technologies

4.2.1 PHP

4.2.2 JQuery

4.2.3 JSON

4.2.4 PSQL Database

4.2.5 Github

4.3 Service Infrastructure

4.3.1 University Hosting

4.3.2 LAMP Server

(PSQL instead of MySQL)

4.4 Service Design

4.4.1 Overview

4.4.2 Method

4.4.3 Structure of Database API

-Controllers -Models

4.5 Website Design

4.5.1 Overview

4.5.2 PHP

4.5.3 Bootstrap

4.5.4 JavaScript and JQuery Library

Google maps embedding

5 Implementation

5.1 Overview

5.2 PHP

5.3 Google Directions API

-talk about long motorway journeys and how steps can be very far apart

5.4 Database

5.5 Database API

-procedural v OO

5.6 Website

-communication with API.

6 Testing

6.1 Overview

6.2 Database API Testing

6.2.1 Overview?

6.2.2 Unit Tests

6.2.3 Functional Tests

6.3 Website Testing

6.3.1 Overview

6.3.2 Functional Tests

7 Evaluation

7.1 Original Goals

7.2 Accomplishments

7.3 Future Improvements

7.4 Future Development

7.5 Design Choices

7.6 Approach

8 Bibliography