جامعة الأخوين
AL AKHAWAYN UNIVERSITY

School of Science and Engineering

YALLA

Carpooling App

EGR 4402
Capstone Design

[Final Report]

LIMOUNI OUADIE

Supervised by: Dr. FALAH BOUCHAIB

Fall 2014

Acknowledgement

I would like to express my deepest appreciation to all those who provided me the possibility to complete this capstone project in general and this report in particular. A very special gratitude I give to the capstone coordinator, Dr. Yassine Salih Alj, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of Dr. Bouchaib Falah, my supervisor, who constantly guided me throughout the semester, provided me with valuable information, and helped me have a clear vision of what the project should look like so as to eventually achieve it.

I am also highly indebted to Al Akhawayn University with all its faculty members and staff who made these past four years an enriching and extraordinary learning journey.

Last but not least, I would like to thank my parents, brother, and my friends without whom I would not be here today. They supported me in every possible way and I would forever be grateful to them.

Table of Contents

Abstract

Today we have come to identify so many issues and problems related to the growth of population all around the world. One of these relate to transportation as people need to commute from place to place constantly. Employees, for instance, need to shuttle from their work place to their home, and students need to go back and forth to school etc… A lot of times, people would have a spare spot in their car and could give a ride to someone heading to the same direction. However, the problem that exists today can be defined as the lack of coordination and communication between these two parties.

Carpooling also known as ride-sharing or car-sharing solves the above issue. This is done by sharing car journeys so that more than one person travels in the car leading to a considerable reduction of each person's transportation costs but also offering numerous benefits that touch upon all aspects of the society from environmental to financial ones. [1]

"Yalla", the title of my project, which have been chosen so that it relates to the Moroccan dialectic word for "Let's go", is a web service offering the possibility for anyone to either post a ride or request a spot in an already existing one. As a consequence, this creates a stress-free environment for people to travel without worrying about calling a taxi or being late for an appointment etc…

Key words: Web service, PHP, MySQL, HTML/CSS, JavaScript, Ajax, ride, driving, riding, origin, destination, localization.

I. Introduction

Morocco like many other countries suffer from public transportation issues and also from heavy traffic, especially in large cities such as Casablanca and Rabat. In this matter, the idea of creating a platform for both drivers and passengers to post or find rides at the simple click of a button or tap of a finger came to me as a way to ease the traffic and solve lots of transportation problems. Because after all more people in a car means less cars on the roads which eventually leads to safer roads and a healthy environment.

During a specific period, two or more people going from the same origin to the same destination can make arrangements to share a car journey. “Yalla” not only offers its user a friendly web platform to make this idea practical but also takes it a step further to make it socially driven so that people can make new connections and friendships.

.

II. Objectives of the Project

The project, which is open to all users and particularly to Moroccan ones who have access to the internet through their computer, aims to provide them a simple and fun way to get where they need to go. The concept of "Yalla" combines simplicity, fun and utility all at once.

III. STEEPLE Analysis

Social and ethical aspects among various others are omnipresent in this project as the main idea behind "Yalla" as a carpooling application is to bring people together through real healthy social connections provided that the website will be using selected features such as profiles. Moreover, drivers and passengers which constitute the pillars of this app will be ethically contributing in countless areas both directly and indirectly, from saving money to making new friends all the way through taking care of the environment.

The intended results of using the website are:

1) Reduction in the number of vehicles on the roads,

2) Reduction in expenses of gas,

3) Reduction in energy consumption, CO2 emissions and pollution in general,

4) Provision of social connections.

Putting these words in a STEEPLE form clears the vision I have of the project and the aspects it touches upon:

Social – lifestyle changes, social mobility, social connections

Technology – development of a new web service

Economic – cost saving, economic growth

Environment – preservation of natural resources, reduction of carbon foot print

Political – Competition with means of public transportation

Legal – insurance policies, risk management

Ethical – client confidentiality, safety of passengers

IV. Description of the Website's features

The following is a description of what the website will look like broadly and the features that will be made available:

- After the web page is loaded, the user is automatically assigned the status of "guest". With this status, the user can only view the different pages of the website. One can search for rides only and browse the about page made available to the public.
- When trying to access the 'Post a ride' page, the 'Rides' page, or the 'Profile' page, the user will be notified that a Login is required.
- In the navigation bar of the website there is a link to the "Registration form" page if the user is a new one and has never been on the website before. Else, the user can choose to Login in the same page by entering valid credentials consisting of an Email Address and a Password.

- Once the user is logged in, which can be clearly seen in the navigation bar, he/she can now post a ride in addition to searching for existing ones on the home page.
- The "rides" page is composed of two parts and helps the user manage his/her rides:
    - *Driving*: in this part of the page, the user has access to the upcoming rides that he/she posted
        - The table contains the following columns:
            - Departure date and time
            - Origin
            - Destination
            - Number of riders
            - Action to be taken: The user can manage the requests by either declining, approving, or simply ignoring a particular request.
                - ✓ Upon approval of a request, the counter for the remaining seats in the car is decremented and the user who has made the request is automatically notified that it has been approved.
    - *Riding*: this part of the page gives an overview on the upcoming rides for which a user has made a request.
        - A table contains the following columns:
            - Departure date and time
            - Origin
            - Destination

- Request status: the request can be either pending (no action has been taken by the author of the ride), approved, or declined.

- The "search for a ride" page which is at the same time the homepage contains a search mechanism as follows:
    - From: the origin
    - To: the destination
    - Date: a date picker
    - Smoking allowed: a check box if clicked on specifies that the user filters only rides in which smoking is allowed.

- In the same page ("search for a ride") if the search is successful a list of found trips is displayed to the users with information about the author of the post:
    - Driver's name
    - Price
    - Ride length
    - Seats remaining
    - Message from the driver
- The "Post a ride" page allows the user to fill in a form and submit it so as to post a ride, it contains fields as follows:
    - From: the origin
    - To: the destination
    - Departure: date and time
    - Trip length: the length is hours and/or minutes is calculated automatically by the system depending on the origin and destination

- Price: the price for a single seat
- Seats in the car: how many seats are available
- Smoking allowed: a check bock if clicked on specifies that smoking is allowed in the car
- Message: a custom message from the user to his/her potential ride sharers

V. Requirements Engineering

Requirements engineering can be defined as the process of establishing the services that are required from a system and the constraints under which it is developed and operates. It is concerned with the whole life cycle of the system, from the elicitation, specification, analysis, all the way to the implementation or development of this latter.

1. Levels of Requirements

1.1. User requirements

- "Yalla" carpooling web service should provide a platform for its users in which they can interact in a way to fulfill the ride sharing objective.

1.2. System requirements

- This web service should provide a search mechanism for rides and eventually make a request, a way to post new rides, and another to manage posted rides and keep track of requested ones.
    - The system fails if it doesn't meet one of these criteria. For instance, returning a "No rides found" error even though there are rides at that specific date, origin, and destination. Otherwise, the system is perfectly operational.

2. Types of Requirements

2.1. Functional requirements

- The web service shall enable users to register to the website by entering information such as their first name, last name, email address, password, gender, and driver's license number which is optional.
- The users shall be able to login to the system using their email address and passwords as credentials.

- The web service shall enable users to perform the following actions:
    - Search a ride and view it: depending on inputs that include the origin, destination, date, and whether smoking is allowed or not.
    - Post a ride: by entering information that describe the ride namely the origin, destination, date and time, price, smoking allowed criterion,, available seats in the car, and a customizable message.

- Users shall be able to logout of the system whenever they want.
- The system shall assign a profile page to the users automatically after they register.

❖ Other features that can be added to the system if the time allows for implementation are:

- The system shall provide a basic messaging system where users post and reply to other users.
- The system shall provide a way for users to be friends with other users based on a request/respond logic
- The system shall provide a notification system to keep users informed of a new notice in their personal profile.

2.2. Non-Functional requirements

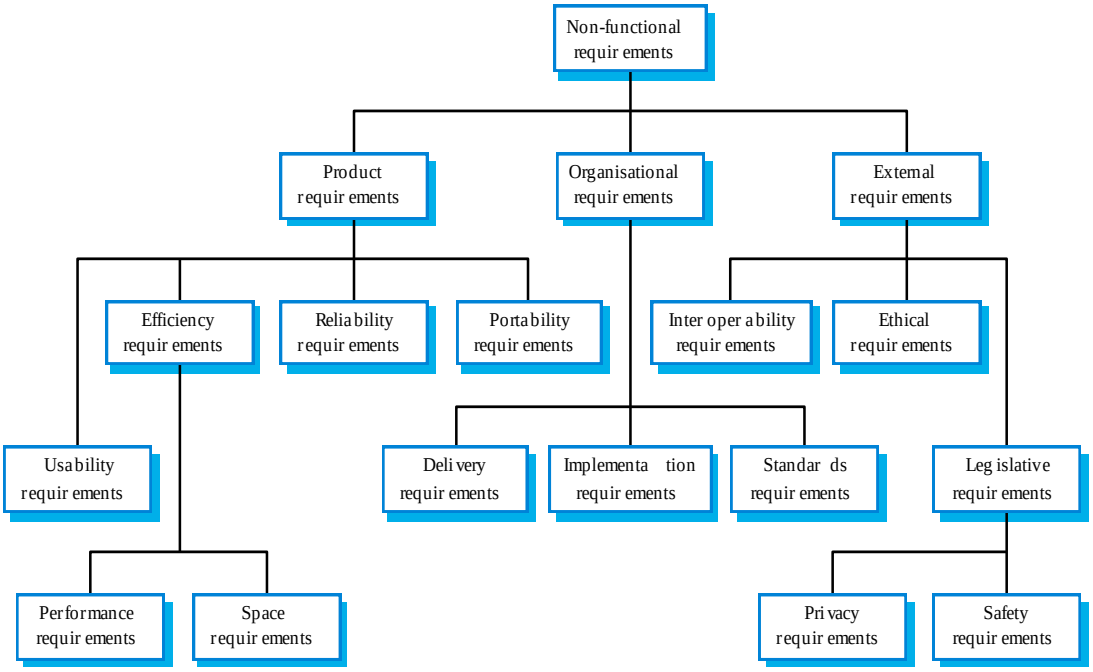


Figure 1: Non-functional requirements types

2.2.1. Product requirements

- The web application shall be efficient, reliable and most all perform all its operations in a correct and relatively fast manner.
- User friendliness of the interface shall be taken into consideration.
- The website shall work on all browsers.
- The user interface shall be implemented in HTML, styled with CSS, and hides all the backdoor functionalities that PHP, Ajax, and MySQL offer.

2.2.2. Organizational requirements

- Data shall be stored and retrieved from a local database.
- The project shall be developed in a period of three months.

- The project shall be delivered on time as a Capstone project within an academic scope.

2.2.3. External requirements

- The website shall not disclose personal information about its users apart from basic information.
- The website shall not constitute any harm to its users.

  P.S: Please refer to the STEEPLE ANALYSIS for legislative and ethical requirements.

VI. Development Process

The development process of a web service is of no difference from other software. It is essential to follow a model that will cover all aspects of the life of the software. The model I chose to deploy and I see most efficient is the classic life cycle model or waterfall model. By choosing this approach I will guarantee a well-designed system that meets all the requirements agreed upon in the first stage. [2]

1. Development Methodology

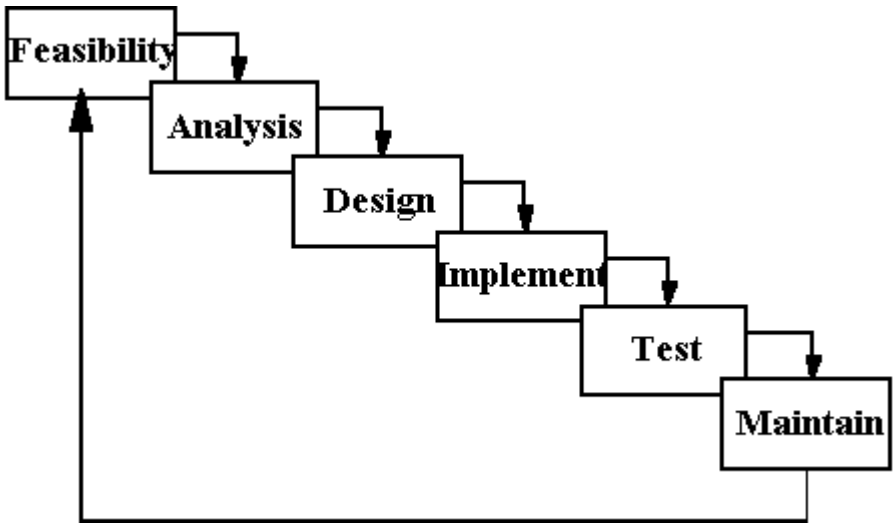


Fig2: Classic waterfall model

1.1. Feasibility

From a pragmatic point of view and at an early stage of the project, I should make sure that the system to be deployed is attainable and will be in the real world. “Yalla” targets mainly Moroccan users but can also be seen as a service for any user in general.

The fact that “Yalla” uses features such as profiles that makes it possible for them to interact, it can be seen as socially driven which makes it unique and relatively different from existing similar web services.

The next step after requirements specification and a discussion of the project’s feasibility is the analysis and design described below.

1.2. Analysis

The goal of system analysis is to determine where the problem is in an attempt to fix the system. This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

1.3. Design

This step is crucial as it is the ultimate basis of the project as a whole where the overall structure is to be defined. Low-level (How modules inside the website should relate to each other), interface (How the website should look like) and data (What data will be needed) designs should be addressed which will facilitate the implementation of the system that will basically be a translation of these latter into code.

1.3.1. UML Diagrams

1.3.1.1. Use case Diagram

A Use case diagram refers to what the system does from the perspective of an external viewer. The stress here is on ***what*** the system does rather than how it does it. In brief, a use case is a summary of situations for a particular job.

Below is the general use case diagram of "Yalla" Carpooling system:

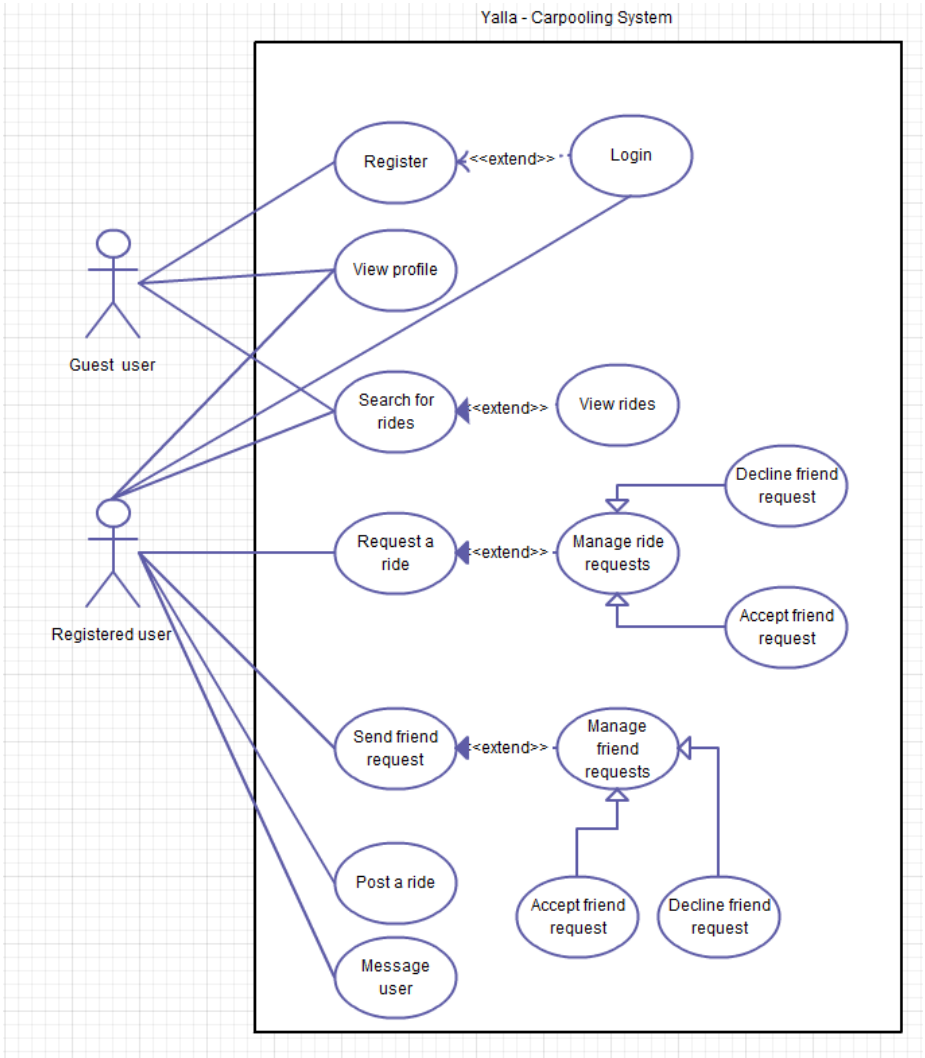


Figure 3: General Use Case Diagram [4]

1.3.1.2. Flow of Events of some Use Cases:

- Register:

Use Case Name ID	**Register** **1**
Actor	**Any user**
Description	**Any visitor of the website can register to become member of the website.**
Pre	**User is new to website.** **User has chosen to follow the link for registering.**
Main Flow	**Step 1.** User presses the "Register" button. **Step 2.** User fills in all the fields of the registration form **Step 3.** System makes sure the fields are filled in appropriately. **Step 4.** System stores the user information in the local database.
Alternative Course	**Step 3.** System notifies user if a field is missing. **Step 4.** User fills in the missing fields. **Step 5.** System makes sure the fields are filled appropriately. **Step 6.** If some fields are missing again, repeat step 3.
Post	**System creates an account for the user.** **User can now use his Email address and Password to Login.**

- Post a ride:

Use Case Name ID	**Post/create a ride** **2**
Actor	**Registered user**
Description	**If the user is registred on "Yalla", then posting or creating a customized ride is possible. This ride will be viewed by other users.**
Pre	**User is registered.** **User has logged into the system.**
Main Flow	**Step 1.** user presses the "post a ride" button **Step 2.** user fills in the needed fields with the appropriate information **Step 3.** System makes sure all the fields are filled correctly. **Step 4.** User press submit button. **Step 5.** System stores the information in the database and creates a ride.
Alternative Course	**Step 3.** System notifies user if a field is missing. **Step 4.** User fills in the missing fields. **Step 5.** System makes sure the fields are filled appropriately. **Step 6.** If some fields are missing again, repeat step 3.
Post	**System creates a ride for the registered user.** **Other users (Registered users or guest users) can search for this ride and view it.**

1.3.1.3. Entity Relationship Diagram

The entity-relationship diagram (ERD) is a graphical representation of entities and their relationships to each other. These relationships are typically used in regard to the organization of data within a certain database.

A relationship describes how the data is shared between entities.

Below is the ERD for “Yalla” carpooling system which makes use of three types of relationships:

1- Many mandatory to many optional: this relationship is present between the entity ‘user’ and the entity ‘friends’ and basically means that users can have friends and these latter are in return friends with other users.

2- One mandatory to many optional: this relationship is present between the entity ‘user’ and the three entities ‘notifications’, ‘ride_request’, and ‘ride’. This means that a user can post zero or more rides, can request zero or more rides, and can get zero or more notifications depending on the situation.

3- One mandatory to many mandatory: this relationship is present between the entity ‘ride’ and the entity ‘place’. This means that a ride is composed of one or more places. In this case a ride is composed of exactly two places, namely an origin and a destination.

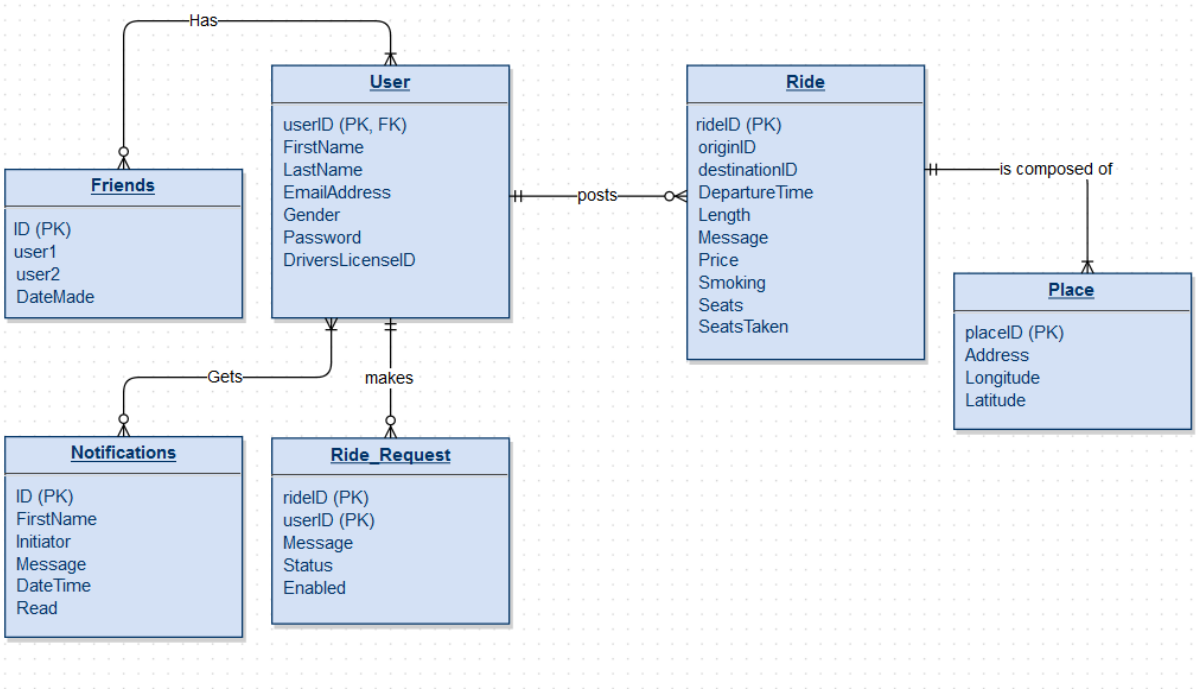


Figure 4: Entity Relationship Diagram [4]

1.3.1.4. Data dictionary

The following is the data dictionary of some of the most important tables that compose the database as generated by PHPmyAdmin:

Table**: Place**

Column	Type	Null	Default
id	int(10)	No	
address	varchar(128)	Yes	*NULL*
lat	float	No	
lon	float	No	

Table: **Ride**

Column	Type	Null	Default
id	int(10)	No	
price	int(11)	No	
driver_id	int(10)	No	
spots	tinyint(4)	No	
spots_taken	tinyint(4)	No	
length	varchar(128)	No	
message	varchar(4096)	Yes	*NULL*
smoking	binary(1)	No	
departure_time	int(11)	No	
origin_id	int(11)	No	
destination_id	int(11)	No	

Table: **User**

Column	Type	Null	Default
id	int(10)	No	
first_name	varchar(64)	No	
last_name	varchar(64)	No	
email_address	varchar(128)	No	
drivers_license_id	varchar(64)	Yes	*NULL*
gender	binary(1)	No	
password	varchar(64)	No	

1.4. Implementation

1.4.1. Technology Enablers

There are different approaches to choose from when trying to develop a web application, either go through a commercial software package or a free one. Nothing stops one from building a website in a CMS (Content Management System) using free ready-to-use templates, but the approach I opted for is to develop the website locally on my machine for various purposes, namely testing ones that I will cover on the following part on this report.

Being currently a user of Windows, I opted for one of the WAMP stacks that are available free for use.

A WAMP stack is basically a set of open source applications that are combined with Miscrosoft Windows and are used in web server environments. The WAMP stack provides four key elements of a web server all being local: an operating system, a web server, a database, and a web scripting languages, plus add-ons if necessary. Combining usage of all of these elements is called a server stack, and in this case:

- Windows is the Operating System
- Apache is the HTTP server (web server)

- MySQL is the Relational Database Management System (RDBMS)
- PHP is the scripting language

Other Programming languages I am using are:

- HTML5 (Hyper Text Markup Language) as a markup language for web pages creation
- CSS (Cascading Style Sheets) to format and style the web pages
- JavaScript is the dynamic scripting language
- Ajax (Asynchronous JavaScript+XML) is a group of technologies mainly used in my project to exchange data asynchronously between browser and server to avoid full page reloads.

➢ *Explaining different roles that PHP, MySQL, HTML, CSS, and JavaScript play in web development:*

Fig 5: Relationships between User, browser, and server

Softaculous AMPPS as a Platform for web development

I chose to download and install AMPPS which offers the WAMP solutions described above. It has a very nicely shaped and friendly graphical user interface with a panel offering numerous functions that can be accessed directly from the browser. [3]

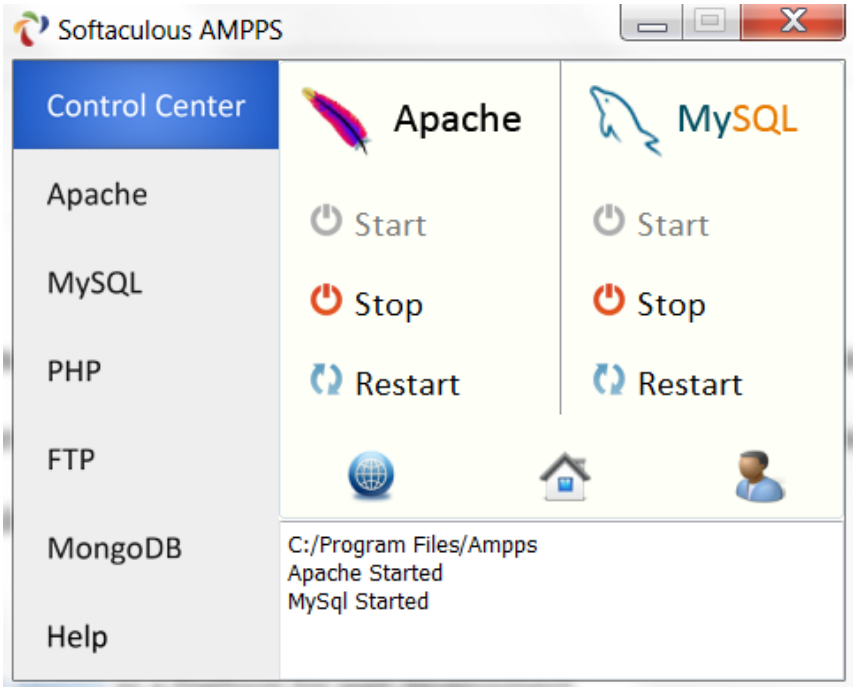


Fig 6: AMPPS control panel

a. **Database/Tables creation using MySQL:**

```
// Ride Table Definition
$ride_table_definition = RIDE_TABLE."
(
   id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   price INT NOT NULL,
   driver_id INT UNSIGNED NOT NULL,
   spots TINYINT NOT NULL,
   spots_taken TINYINT NOT NULL,
   length VARCHAR(128) NOT NULL,
   message VARCHAR(4096),
   smoking BINARY(1) NOT NULL,
   departure_time INT NOT NULL,
   origin_id INT NOT NULL,
   destination_id INT NOT NULL,
   CONSTRAINT chk_spots CHECK (spots_taken <= spots))";
```

```
// Place Table Definition
$place_table_definition = PLACE_TABLE."
(
   id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   address VARCHAR(128),
   lat FLOAT NOT NULL,
   lon FLOAT NOT NULL
)";


// Ride Request Table Definition
$ride_request_table_definition = RIDE_REQUEST_TABLE."
(
   ride_id INT UNSIGNED NOT NULL,
   user_id INT UNSIGNED NOT NULL,
   PRIMARY KEY (trip_id, user_id),
   message VARCHAR(4096),
   status TINYINT DEFAULT 0,
   CONSTRAINT chk_status CHECK (-1 <= status AND status <= 1)
)";

// User Table Definition
$user_table_definition = USER_TABLE."
(
   id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
   first_name VARCHAR(64) NOT NULL,
   last_name VARCHAR(64) NOT NULL,
   email_address VARCHAR(128) NOT NULL UNIQUE,
   drivers_license_id VARCHAR(64),
   gender BINARY(1) NOT NULL,
   password VARCHAR(64) NOT NULL
)";
```

b. Establishing the database connexion:

```php
$appname = 'Yalla';
$dbhost = 'localhost';
$dbname = 'yalladb';
$dbuser = 'root';
$dbpass = 'mysql';

$connection = mysql_connect($dbhost, $dbuser, $dbpass) or die(mysql_error());
mysql_select_db($dbname, $connection) or die(mysql_error());
```

Fig 7: Screenshot of the code demonstrating the connection to the Database

c. Modularization of code:

I have opted to write the code of the application in a modularized way that will enable me to debug and find errors easily. By doing this, the code is kept in separate files and can be called when needed. This type of implementation is very efficient, clean, and easily maintainable.

We can see below how different files are separated inside different folders. CSS files are kept apart, JS files as well, PHP files that execute specific tasks are isolated, and the main PHP files are in the root folder.

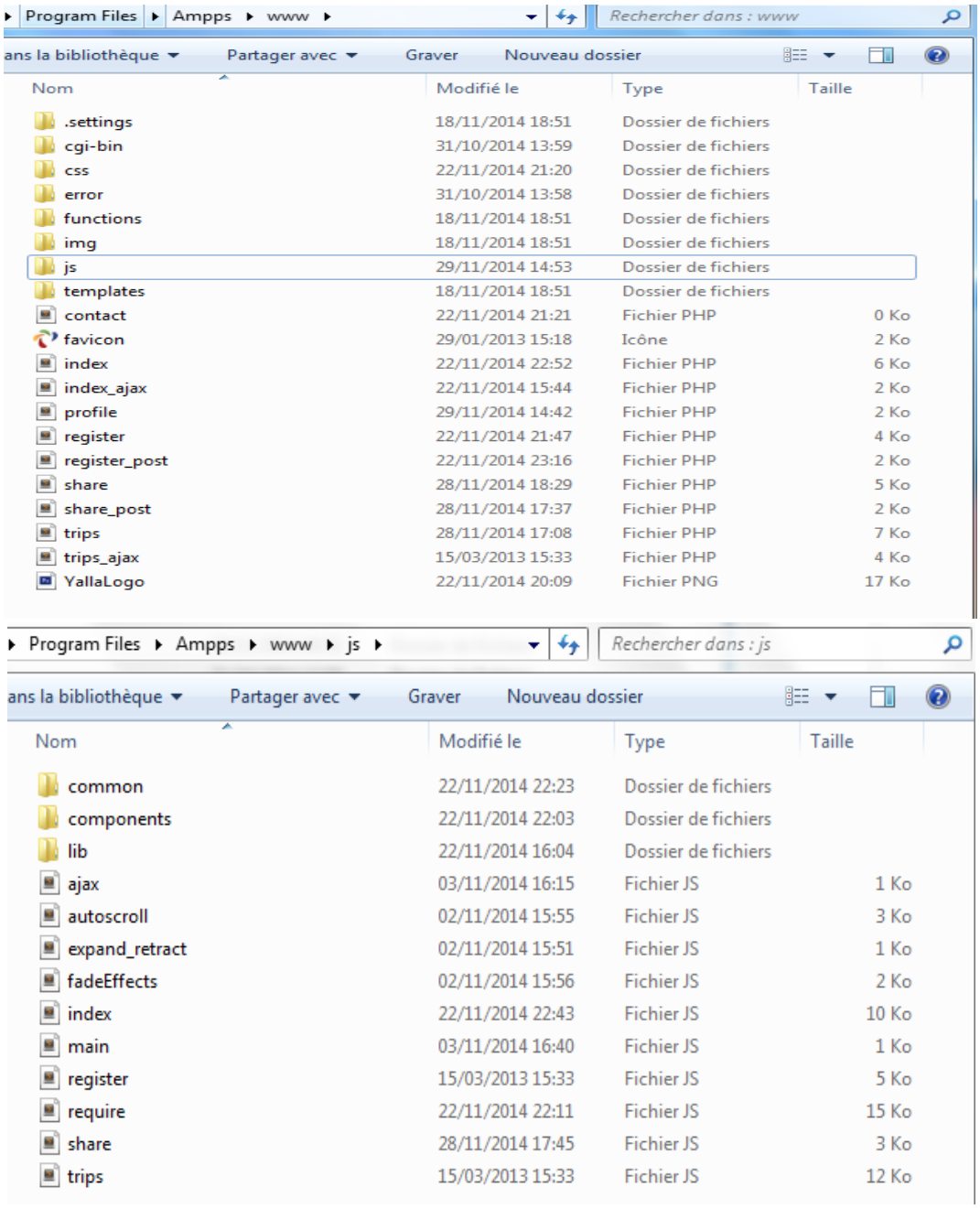


Fig 8: Modularization of the code

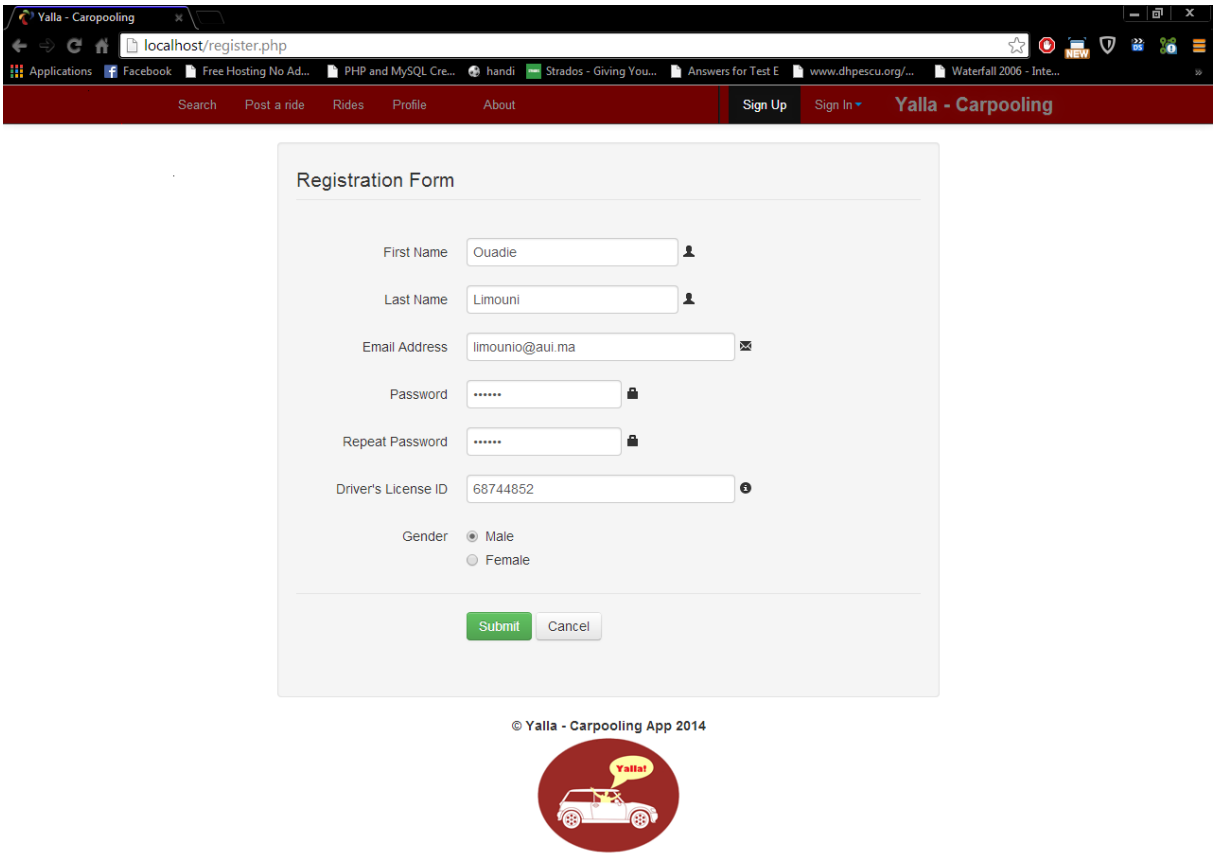
1.1. Testing

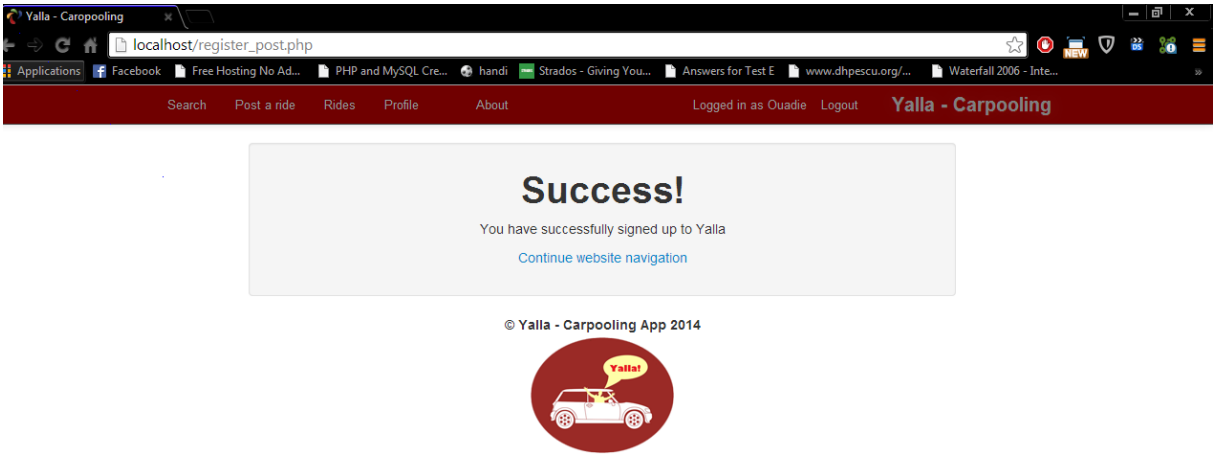
Testing of different components of the website is done simultaneously with the implementation. Every time a new component, function, of table in the database etc… is added, I make sure to check that the website is still running and functional.

This type of testing done on a regular basis provides early detection of errors that give insight on the actions to be taken so that we can move to developing other modules.

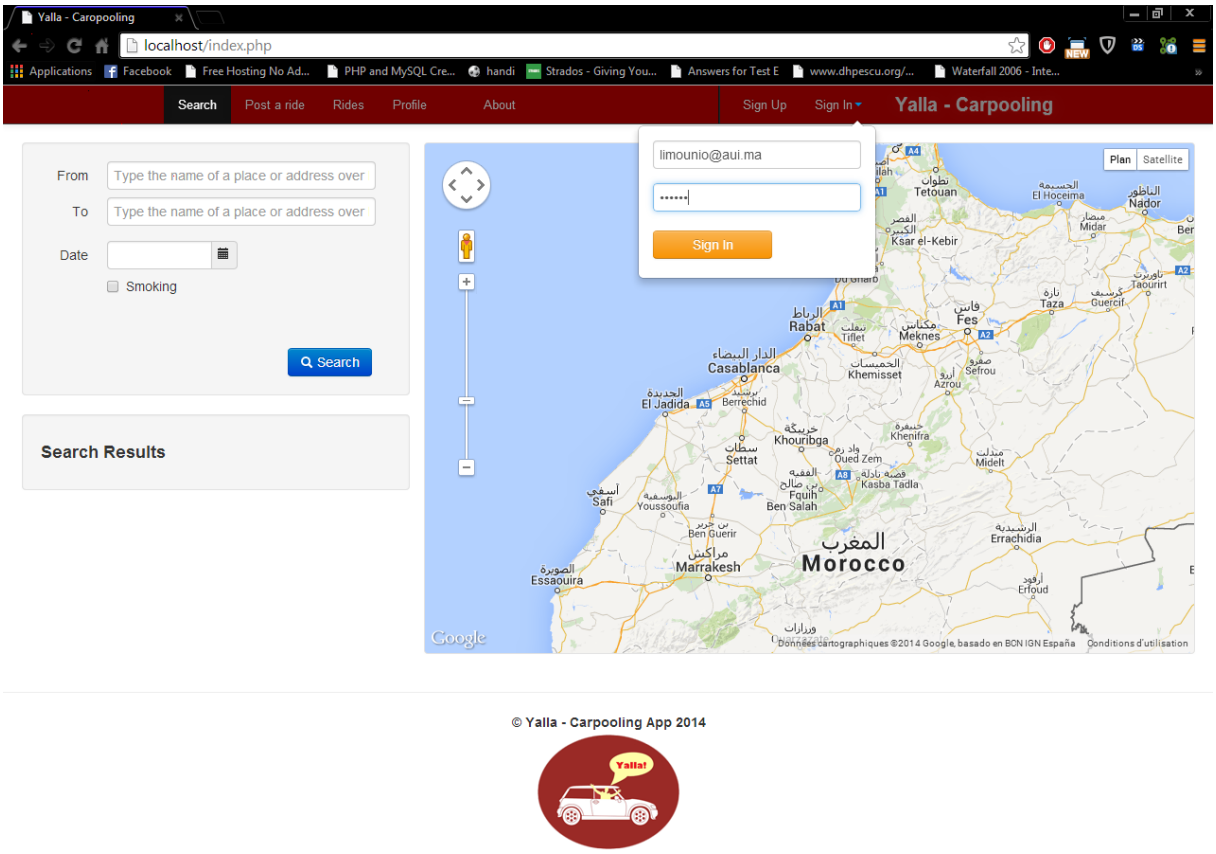
VII. Screenshots of the website's different pages:

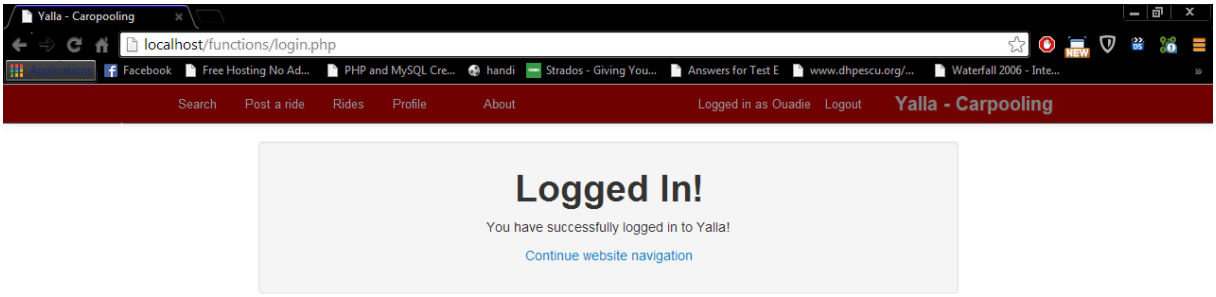
- Registration form page (register.php);



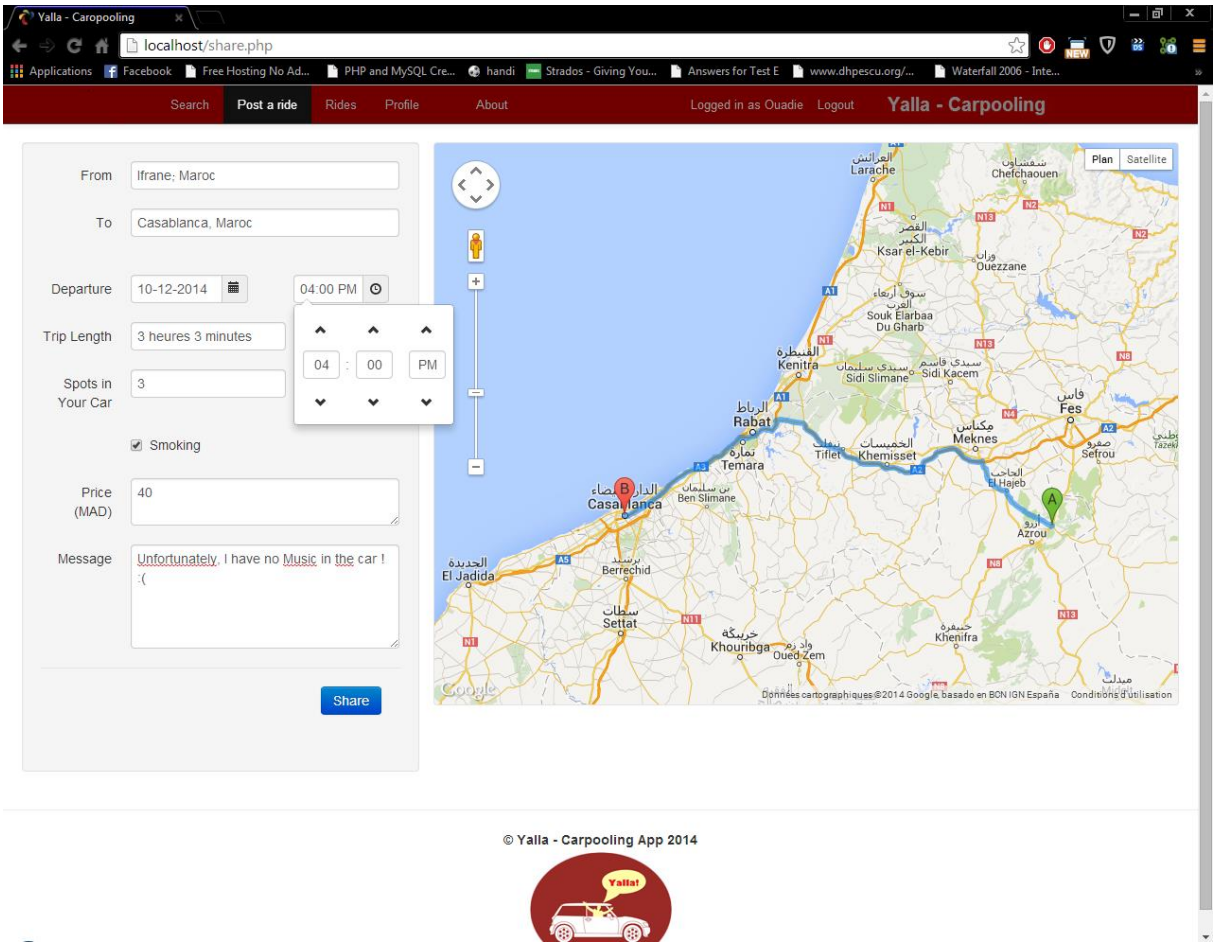

Yalla - Caropooling
localhost/register_post.php
Applications
Facebook
Free Hosting No Ad...
PHP and MySQL Cre...
handi
Strados - Giving You...
Answers for Test E
www.dhpescu.org/...
Waterfall 2006 - Inte...
Search
Post a ride
Rides
Profile
About
Logged in as Ouadie
Logout
Yalla - Carpooling
Success!
You have successfully signed up to Yalla
Continue website navigation
© Yalla - Carpooling App 2014
Yalla!

- Sign in (index.php)

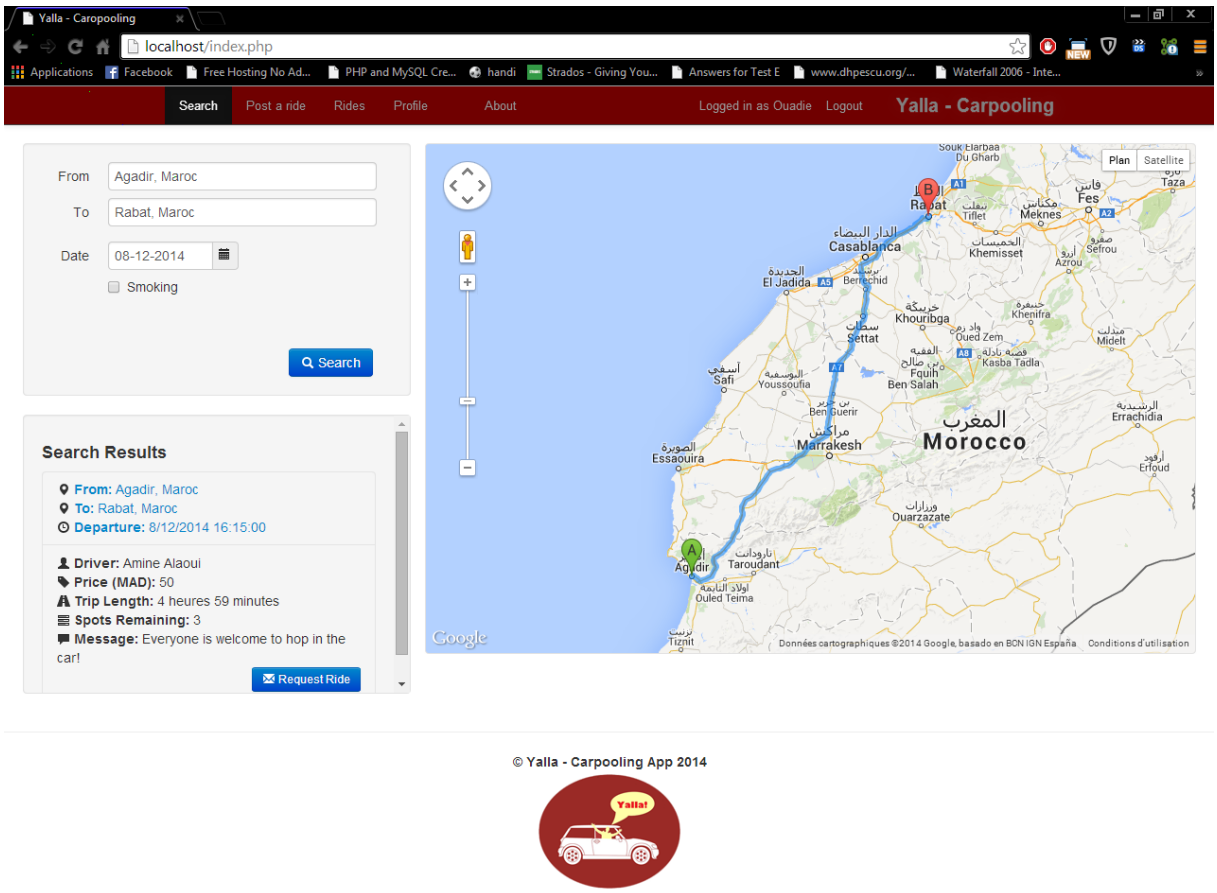




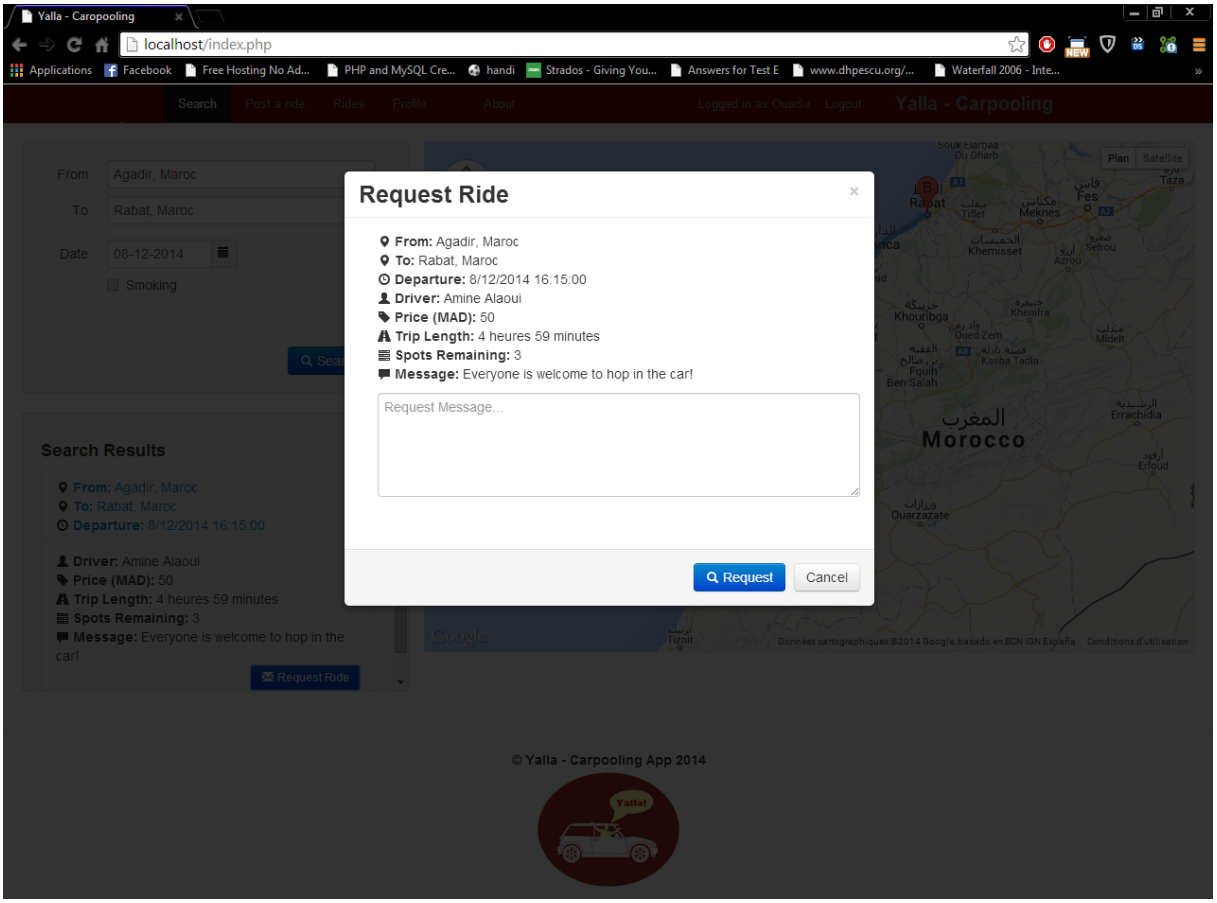
- Post a ride


Yalla - Caropooling
localhost/share.php
Search
Post a ride
Rides
Profile
About
Logged in as Ouadie
Logout
Yalla - Carpooling
From
Ifrane, Maroc
To
Casablanca, Maroc
Departure
10-12-2014
04:00 PM
Trip Length
3 heures 3 minutes
Spots in Your Car
3
Smoking
Price (MAD)
40
Message
Unfortunately, I have no Music in the car ! :(
Share
Plan
Satellite
© Yalla - Carpooling App 2014

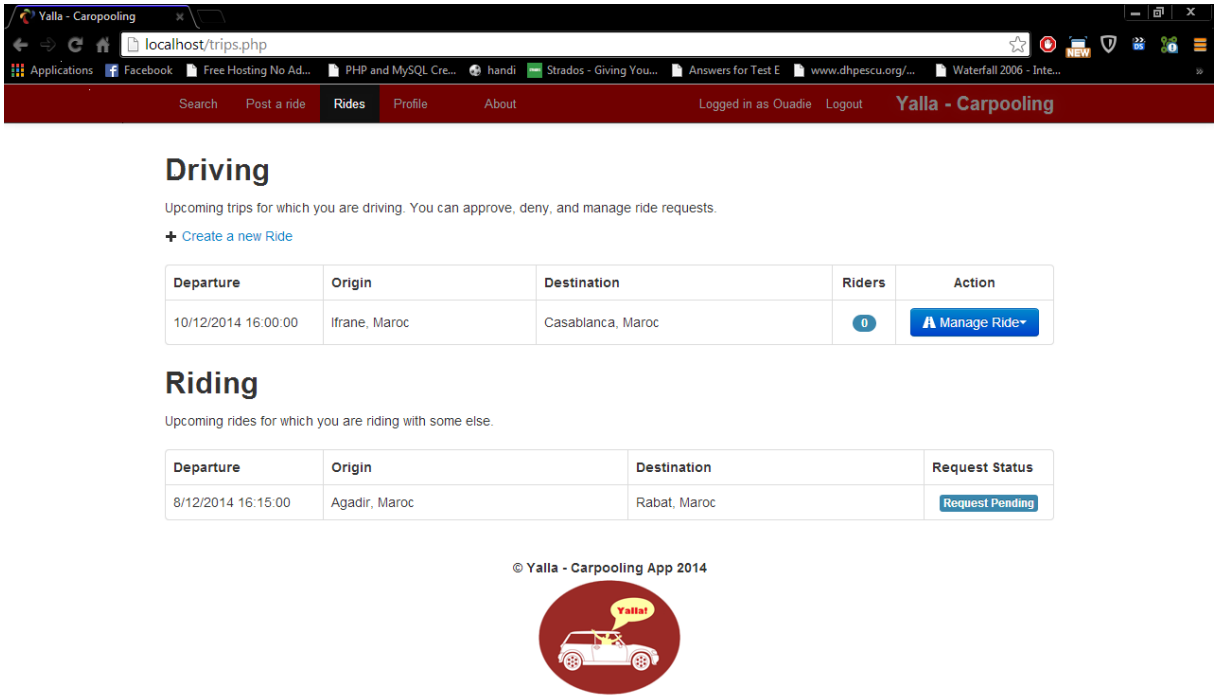
- Search for a ride

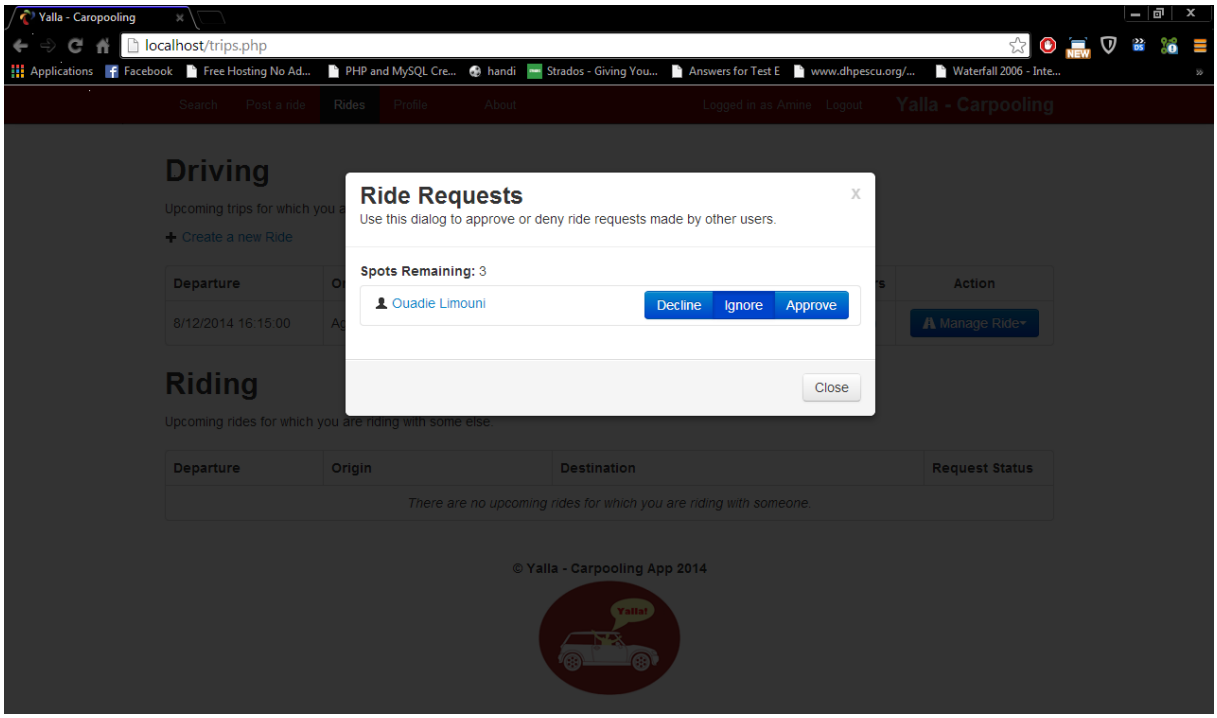


- Request a ride



- Manage rides





Usage of Bootstrap and GitHub:

Bootstrap is a front-end framework for building responsive websites. Whether it is application frameworks, blogs, or other CMS applications, Bootstrap can be a good fit, as it can be as vanilla as you like. Its combination of HTML, CSS, and JavaScript make it easy to build robust sites without adding a lot of code. With a default grid system, layouts come together with ease, and the styling of buttons, navbars, and tables make basic markup look great from the get-go. A dozen or so JavaScript plugins catapult you into adding interactive elements to your site. [6]

Bootstrap can be downloaded for free and comes with various JS and CSS files that can be pasted with one's code. Classes defined in such files can be called directly and used.

GitHub is a Git depository and web-based hosting service, which offers all of the source code management (SCM) functionality of Git as well as adding its own features. GitHub provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project. [7]

Working with both these tools as part of code reuse made the task of coding and implementing the project less difficult.

Below is a snapshot of a basic template using Bootstrap that can be modified and edited as one's wishes to:

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Fig 9: Bootstrap template

Usage of Google Directions API:

The Google Directions API is a service that calculates directions between locations using an HTTP request. Search is done to find directions for several modes of transportation, including transit, driving, walking or cycling. In “Yalla’s” case, the transportation mode is ‘Driving’, since cars are the means used in Carpooling. Directions may specify origins, destinations and waypoints either as text strings (e.g. "Casablanca, Morocco") or as latitude/longitude coordinates.

"This service is generally designed for calculating directions for static (known in advance) addresses for placement of application content on a map; this service is **not** designed to respond in real time to user input, for example. For dynamic directions calculations (for example, within a user interface element), the use of JavaScript is primordial." [5]

Shown below is a screenshot of the function that calculates the route based on input that is we get from the user:

```
MapRoute.prototype.calculateRoute = function() {
  var from, request, to,
    _this = this;
  from = this.from.val().trim();
  to = this.to.val().trim();
  if (!from || !to) {
    return;
  }
  request = {
    origin: from,
    destination: to,
    travelMode: google.maps.TravelMode.DRIVING,
    region: 'Morocco'
  };
  return this.directionsService.route(request, function(result, status) {
    if (status === google.maps.DirectionsStatus.OK) {
      _this.directionsDisplay.setDirections(result);
      _this.result = result;
      _this.removeError();
      return _this.updateRoute();
    } else {
      _this.result = null;
      return _this.setError('No routes found.');
    }
  });
};
```

Fig 10: Snapshot showing how a route between two locations is calculated

VIII. Future work

The least that can be said about this project is that it is far from being complete. This project was done as a Capstone Project during a three months period of time. The fact that I worked on this project alone and had taken other classes in parallel restricted the effort and time I put on the implementation of the project.

There are a lot of improvements that can be added to this project that will eventually make it more complete and richer when it comes to features and also content. The vision I have for this project to make it compete with existing “Carpooling” web services is for it to be socially driven.

Nowadays, one of the major areas in technology is Social media. This latter comes side aside with big fields such as Big data, cloud computing, etc... Having a website that offers at the same time a unique service such as Carpooling and making is a social platform where users can view each other’s’ profiles, request friendship, and talk to each other, will allow the user to fully experience the service.

- Some of the features that can be added to the project:
    - Migrate towards a mobile oriented application:
        - This will allow to dynamically locate users that are nearby over a certain radius and display their posted rides for the user to check them out.
        - Users can identify themselves by sharing their location and either requesting or posting a ride on the go.
    - Add a live feed that displays rides once they are posted to allow all the community to view them.
    - Add a secure online payment method allowing users to pay via the website, so that the author of the ride reserves spots for them.
        - Build a rating system where riders can rate the driver after completion of the carpool, and vice versa. Based on this idea, the user can gain points upon which he can win prizes such as gas coupons etc…

IX. Conclusion

Working on this capstone project helped me gain a lot of experience concerning web development. Being the first time I had to deal with languages such as JavaScript, Ajax, and having little knowledge about HTML and CSS, I had to work hard to adapt to this new field, but most of all learn everything by myself through online tutorials and programming forums.

The idea that I had initially of this project was abstract and blurry which created numerous problems and difficulties for me, but as I advanced in the software process, my vision of things began to clear and I started believing in my abilities to wrap up the project and deliver it on time.

The difficulties that I faced were mainly related to the implementation phase as I relied sometimes on software reuse from open source frameworks such Bootstrap and code snippets from GitHub. Thus, putting this code along with mine created some inconsistencies that I had to debug as I progressed in the project.

All things considered, I must say that this project gave me the opportunity to open up to a new field that I never thought of entering before, and mostly helped me decide on what area to specialize in.

X. References

[1] "Green Living Tips." Green Living Tips RSS. Web. 20 Sept. 2014. <http://www.greenlivingtips.com/articles/car-pooling-for-the-planet.html>.

[2] "Category Archives: SDLC." Java World. Web. 24 Nov. 2014. <https://anshuchoudhury.wordpress.com/category/sdlc/>.

[3] "WAMP, MAMP and LAMP Stack : AMPPS Tour." *WAMP, MAMP and LAMP Stack : AMPPS Tour*. Web. 8 Oct. 2014. <http://ampps.com/tour>.

[4] "About Creately , How It All Began and Meet the Team." Online Diagram Software to Draw Flowcharts, UML & More. Web. 24 Nov. 2014. <http://creately.com/about-us>.

[5] "The Google Directions API." *Google Developers*. Google, 16 May 2010. Web. 24 Nov. 2014. <https://developers.google.com/maps/documentation/directions/#Introduction>.

[6] Spurlock, Jake. Bootstrap. S.l.: O'Reilly Media, 2013. Print.

[7] GitHub. "GitHub Help." - User Documentation. GitHub, 6 Apr. 2008. Web. 11 Oct. 2014. <https://help.github.com/>.

Appendix A

LIMOUNI Ouadie
CSC
YALLA – CARPOOLING APP
FALAH B.
FALL 2014

The main purpose of this project is to produce a system that shall be used as a social ridesharing community. This latter will allow users to rapidly find drivers or passengers, depending on the need, who are travelling towards the same destination or along the same route. The system in question is a website hosted locally.

An efficient way to approach this project is by breaking it into small manageable tasks that will add up to constitute our system's lifecycle. Starting by a feasibility study that will allow us to evaluate the potential of the application, then moving to the elicitation part that will result in a software requirements specification to be analysed. The next phases will consist of the design followed by the implementation and testing of the application. This last part is important to ensure that the application actually fulfils the requirements it was designed for and meets quality expectations. In this project, I am going to take the simple software lifecycle a step further so as to bring together both subsystems to function correctly as part of the integration phase. With the full support and supervision of Dr. FALAH, I are sure that I will get all the help and coordination needed among us.

Timeline of the project:

-*September 8: First meeting with supervisor and discussion on the project idea.*

-*September 12: Specifications of the capstone project.*

-*September 19: Feasibility Study and Analysis.*

-*September 22 – October 20: Weekly diaries and Work on the Requirements and design.*

-*October 20: Interim Report.*

-*October 27: November 24: Weekly diaries and Implementation phase*

-*November 24: Final Report*