ΘΕΜΑ: «Σχεδιασμός και Ανάπτυξη εφαρμογής σε κινητά για συνεπιβατισμό»

Πτυχιακή Εργασία

Καζαντζής Μελέτιος ΑΜ 1193

Κυρτίδης Γεώργιος ΑΜ 1218

Παπασταμάτης Λεωνίδας ΑΜ 1260

Επιβλέπων: Δρ. Χριστοδούλου Σωτήριος

Φεβρουάριος 2016

Ευχαριστίες

Η πτυχιακή εργασία αποτελεί διαδικασία χρονοβόρα και επίπονη, διαδικασία που απαιτεί αφοσίωση, επιμέλεια και οργάνωση μα συνάμα επισφραγίζει και ολοκληρώνει τον κύκλο των ακαδημαϊκών σπουδών. Το αποτέλεσμα αυτής δικαιώνει τους συντελεστές, οι οποίοι είναι πλέον έτοιμοι να σταδιοδρομήσουν στο αντικείμενο που επέλεξαν και καταρτίστηκαν.

Για την ολοκλήρωση του πονήματος, πέραν της υλικοτεχνικής υποδομής, απαιτήθηκαν ψυχικά εφόδια για τους γράφοντες, τα οποία πλούσια παρείχαν τα άτομα που πλαισιώνουν την καθημερινότητα των ανθρώπων. Το εγχείρημα είναι πρωτόγνωρο για τον φοιτητή ο οποίος καλείται να τερματίσει το στάδιο του προπτυχιακού και γεμάτος πια από γνώση και δεξιότητες, συμβουλές και παραδείγματα, να ανοίξει τα φτερά του στον ουρανό του επαγγελματισμού. Πως θα ήταν αυτό όμως δυνατό χωρίς συμπαράσταση, χωρίς ενθάρρυνση και κίνητρο; Γι αυτό, η πρώτη σελίδα του έργου αυτού, είναι αφιερωμένη στους αρωγούς της προσπάθειας.

Το συγκεκριμένο έργο είναι προϊόν συνεργασίας, κοινής προσπάθειας και θέλησης, ένας ομαδικός στόχος που αρμονικά επετεύχθη. Πέραν όμως της συγγραφικής ομάδας, πρέπει να αναφερθούμε στα πρόσωπα που ακούσια ή εκούσια συνέβαλαν στο αποτέλεσμα. Αρχικά θα θέλαμε να ευχαριστήσουμε τον εισηγητή του θέματος, καθηγητή του ΤΕΙ Δυτικής Ελλάδος, κύριο Χριστοδούλου Σωτήριο, του οποίου οι κατευθυντήριες γραμμές αποτέλεσαν οδηγό και γνώμονα για την ολοκλήρωση της πτυχιακής εργασίας αλλά και όλους τους διδάσκοντες του τμήματός, οι οποίοι συνέβαλαν στην επιστημονική μας ολοκλήρωση.

Ιδιαίτερη μνεία οφείλουμε στην οικογένεια, τους ανθρώπους που πλαισιώνουν κάθε βήμα της ζωής μας και με την αμέριστη συμπαράστασή τους εμψυχώνουν και υλοποιούν τα όνειρά μας. Άλλωστε το οικογενειακό περιβάλλον είναι η εγγύηση για κάθε επιτυχία και ευημερία.

Τέλος ένα μεγάλο ευχαριστώ στο ευρύτερο φιλικό περιβάλλον στο οποίο κάθε άνθρωπος ζυμώνεται και βελτιώνεται και με τη σειρά του αποτελεί πυρήνα κοινωνικοποίησης και έμπνευσης.

Πρόλογος

Η πρακτική του carpooling (στα ελληνικά ο συνεπιβατισμός) αποτελεί μία διαδεδομένη διαδικασία ταξιδιού κατά την οποία ο έχων μέσο μεταφοράς προσφέρει τη δυνατότητα σε άλλους ανθρώπους να συνταξιδέψουν μαζί του ή αντίστροφα, οι ταξιδιώτες που δεν διαθέτουν μέσο, να αναζητήσουν τέτοιου είδους προσφορές. Η φιλοσοφία της κίνησης συνοψίζεται στη μείωση των εξόδων μετακίνησης, της οποίας το συνολικό κόστος επωμίζονται ισομερώς οι συνεπιβάτες. Αυτή η πρακτική ενισχύει την ανθρώπινη επικοινωνία και συλλογικότητα και φυσικά ελαφραίνει τον ταξιδιώτη από τα όλο και αυξανόμενα ποσά της μετακίνησης. Ειδικά αν αναλογιστούμε την οικονομική κρίση που μαστίζει τις περισσότερες χώρες του κόσμου, μια τέτοια πρακτική αναμένεται να γνωρίσει τεράστια απήχηση.

Στα πλαίσια των γνώσεων μας, αναπτύξαμε μια τέτοιου είδους εφαρμογή για κινητά τηλέφωνα που χρησιμοποιούν λειτουργικό σύστημα android (αλλά και την αντίστοιχη ιστοσελίδα) που δίνουν τη δυνατότητα στο χρήστη να συμπληρώσει τα πεδία ημέρας αναχώρησης και επιστροφής, τόπου αναχώρησης και άφιξης, διαδρομής, διαθέσιμων θέσεων προκειμένου να προβλέψει τη μεγαλύτερη δυνατή εξυπηρέτηση του επιβατικού κοινού. Η διαδικασία αναζήτησης είναι απλή και ευχάριστη και διευκολύνει τους ταξιδιώτες που είτε παρέχουν μέσο μετακίνησης είτε το αναζητούν.

Η εφαρμογή και ο ιστότοπος που αναπτύξαμε προϋποθέτουν δύο μέσα που γνωρίζουν ταχεία αφομοίωση στις μέρες μας. Τόσο τα κινητά τηλέφωνα όσο και το ίντερνετ αποτελούν πια σημεία της καθημερινότητας μας αλλά και χρήσιμα εργαλεία διευκόλυνσης της ζωής του ανθρώπου. Η όλο και ευρύτερη αποδοχή τους από τους ανθρώπους διευκολύνει

το έργο του τεχνολόγου επιστήμονα και αυτός με τη σειρά του προσβλέπει στην αποτελεσματική αξιοποίησή τους. Η εφαρμογή λοιπόν που δημιουργήσαμε ως επιστημονικό επίτευγμα είναι μέσο βοήθειας για το σύγχρονο άνθρωπο.

Συνειδητά και στοχευμένα επιλέχθηκε το θέμα της πτυχιακής εργασίας. Δεδομένης της δυνατότητας που προσφέρει η αξιοποίηση της τεχνολογίας στη σύγχρονη διαβίωση, ο στόχος για την υλοποίηση του έργου ήταν διπλός· αφενός, να διασαφηνιστεί η συμβολή της τεχνολογίας στην εξυπηρέτηση του ατόμου μέσω μιας εφαρμογής και αφετέρου η αξία της εφαρμογής της επιστήμης στα δύσκολα χρόνια που μας δοκιμάζουν. Με πιο απλά λόγια, στο έργο αυτό αποδεικνύεται τόσο η πρακτική εφαρμογή μιας τεχνολογικής έννοιας όσο και η καταλυτική συνεισφορά της τεχνολογίας στην ανθρώπινη επικοινωνία.

Στα επόμενα κεφάλαια αναλύεται διεξοδικά η προσπάθεια των γραφόντων, η τεχνογνωσία και τα μέσα που χρησιμοποιήθηκαν για να επέλθει το αποτέλεσμα. Η πτυχιακή εργασία τέλος, συνοδεύεται από ψηφιακή παρουσίαση και περιγραφή.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Υλοποίηση WEB Εφαρμογής

Η web εφαρμογή υλοποιήθηκε σε php και Mysql.

1.2 Βάση Δεδομένων.

Η βάση δεδομένων της εφαρμογής ονομάζεται carpool_db. Αποτελείται από τους εξής πίνακες:

- member. Πρόκειται για τον πίνακα που περιέχει τα στοιχεία του administrator
- tblcity. Περιέχει τις πληροφορίες για κάθε τοποθεσία
- tblrequest. Σε αυτόν τον πίνακα αποθηκεύονται τα αιτήματα για επιβίβαση σε κάποιο δρομολόγιο από κάποιον πιθανό επιβάτη.
- tblroute. Περιέχει τα δρομολόγια που υποβάλλονται από τους κατόχους των μεταφορικών μέσων
- tbltransportation. Περιέχει τα είδη των μεταφορικών μέσων
- tbluser. Περιέχει τα στοιχεία των χρηστών της εφαρμογής

1.3 Σελίδες PHP

1.3.1 Στοιχεία σύνδεσης

Dbs.php

Περιέχει τα στοιχεία σύνδεσης με τη βάση.

```
<?php
$host = "localhost";
$user = "root";
$pwd = "";
$db = "carpool_db";
?>
```

1.3.2 Κώδικας σύνδεσης βάσης

Carpooldbinfo.php

Περιέχει τον κώδικα για τη σύνδεση με τη βάση.

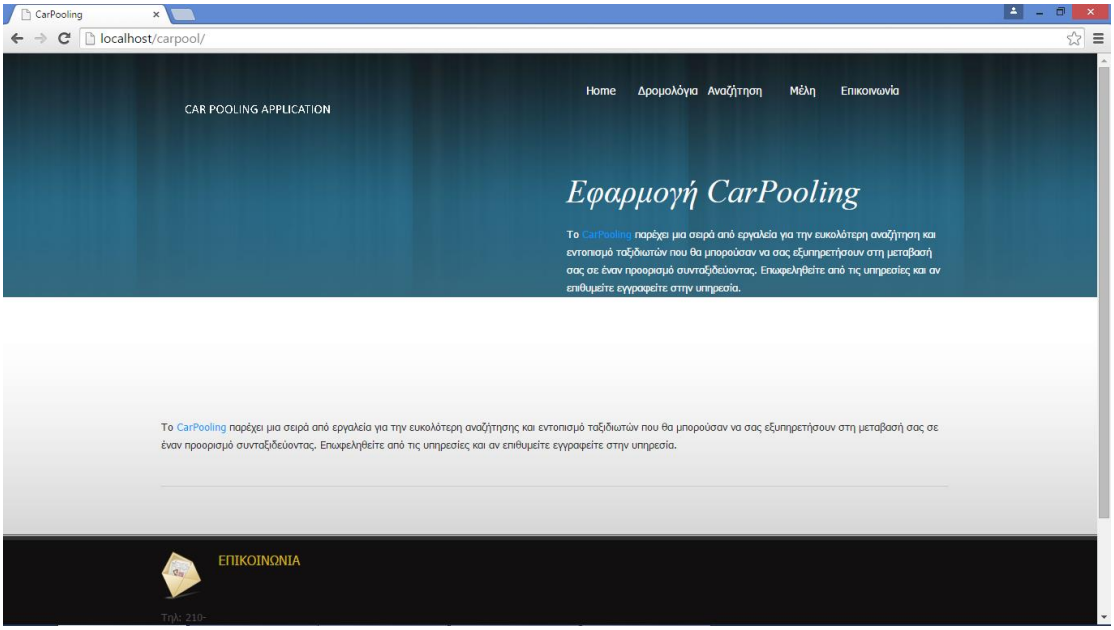
```
<?php
include './dbs.php';
$link = mysql_connect($host, $user, $pwd);
if (!$link)
{
   die ("Error: ". mysql_error());
}
else
{
   $db = mysql_select_db($db);
   if (!$db)
   {
       die("Error: ".mysql_error());
   }
}

mysql_query('set character set greek',$link);
mysql_query("SET NAMES 'greek'",$link);
?>
```

1.3.4 Κύρια σελίδα

Index.php

Πρόκειται για την αρχική σελίδα της web εφαρμογής όπου περιέχονται οι βασικές επιλογές:

- Τρέχοντα Δρομολόγια
- Αναζήτηση Δρομολογίων
- Σελίδα μελών



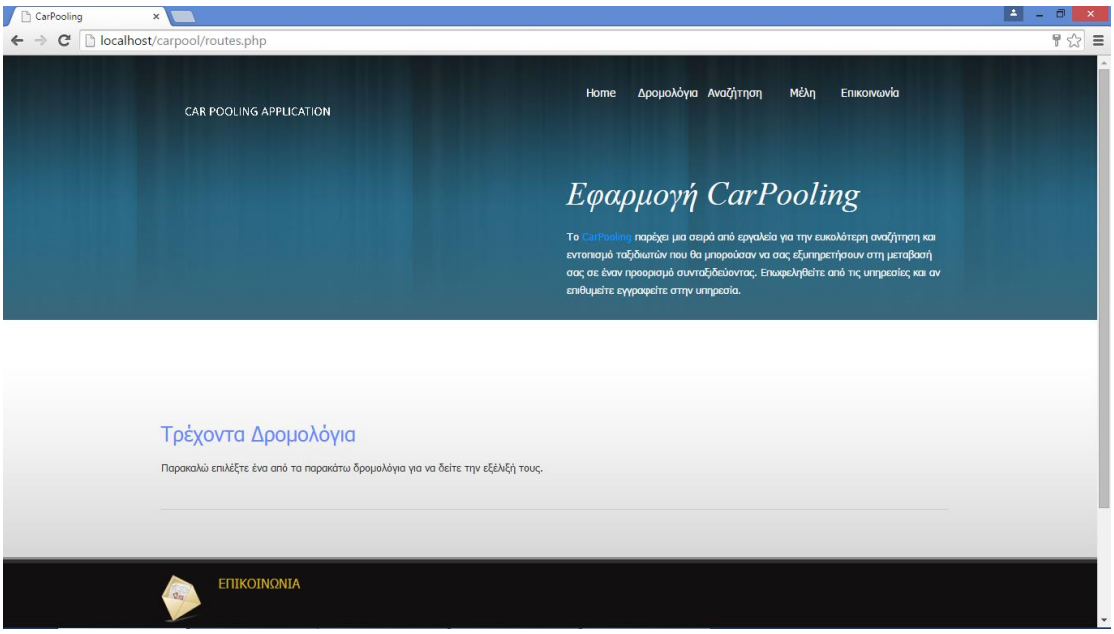
1.3.5 Τρέχοντα Δρομολόγια

Routes.php

Στη σελίδα αυτή εκτελείται ένα select query στον πίνακα tblroutes για να εμφανιστούν όλα τα δρομολόγια της τρέχουσας ημερομηνίας ακολουθούμενα από πληροφορίες για την περιοχή αναχώρησης και άφιξης και την ώρα.

```
<?php
$categories=mysql_query("select * from tblroute where departdate<= CURDATE() and arrivedate>=CURDATE() order by departdate,departtime desc");

$i=0;
while($i < mysql_num_rows($categories))
{
?>
<p><?php
$cities=mysql_query("select * from tblcity where cityid=".mysql_result($categories,$i,"departure"));

$city1=mysql_result($cities,0,"cityname");
$cities=mysql_query("select * from tblcity where cityid=".mysql_result($categories,$i,"destination"));

$city2=mysql_result($cities,0,"cityname");
```

```
                                                    echo
mysql_result($categories,$i,"departdate")."          ".mysql_result($categories,$i,"departtime")."
".$city1." - ".$city2;?> -
                                                    <a
href="routedetails.php?routeid=<?php                                                    echo
mysql_result($categories,$i,"id");?>">Προεπισκόπηση</a></p><hr/>
                                    <?php
                                                $i++;
                                        }
                                    ?>
```



Πατώντας ο χρήστης πάνω σε ένα από τα δρομολόγια μπορεί να δει περισσότερες πληροφορίες μεταβαίνοντας στη σελίδα routedetails.php.

1.3.6 Προεπισκόπηση δρομολογίου

routedetails.php

Στη σελίδα αυτή εμφανίζεται ένας χάρτης Google Map και η τρέχουσα τοποθεσία του οχήματος που εκτελεί το δρομολόγιο που επιλέχθηκε. Αυτό επιτυγχάνεται με τη χρήση εντολών Google Map Api. Στο συγκεκριμένο script εκτελείται ένα ερώτημα στον πίνακα tbltimeroute που περιέχει τις γεωγραφικές συντεταγμένες κάθε οχήματος ανά τακτά χρονικά διαστήματα. Το query ανακτά την τελευταία θέση του οχήματος του δρομολογίου που επιλέχθηκε και προσθέτει ένα marker (createMarker(point,placename)) στον χάρτη:

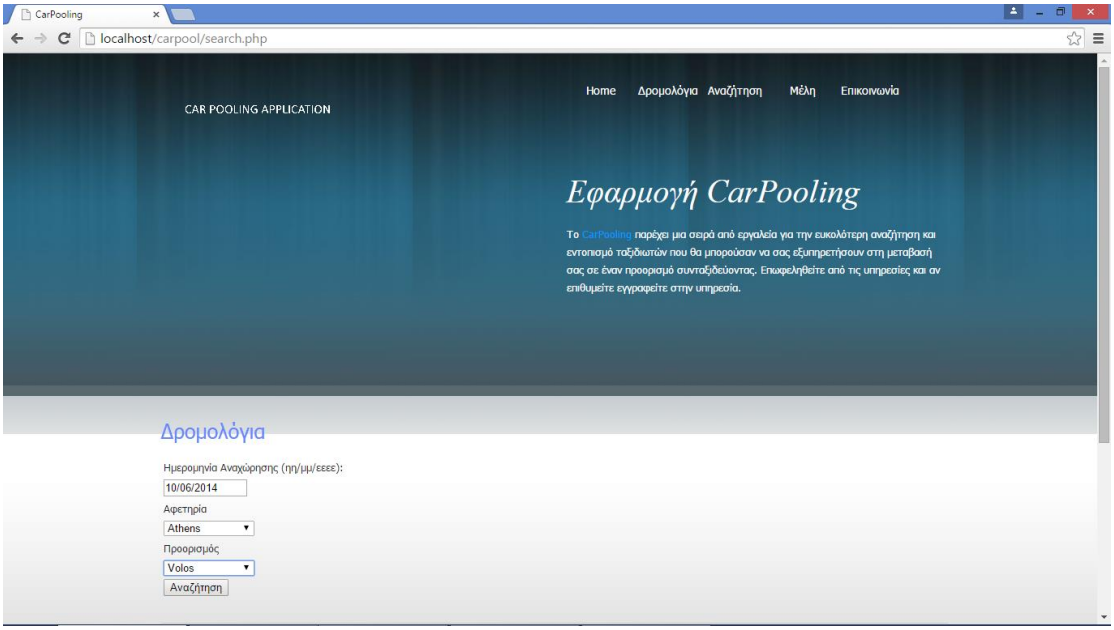
```
<SCRIPT type=text/javascript>
   var allmarkers = [];
   var category = [];
   function initialize() {
    if (GBrowserIsCompatible()) {
      var map = new GMap2(document.getElementById("map_canvas"));
      map.setCenter(new GLatLng<?php echo $mapXY.','.$cityzoom;?>);
      map.addControl(new GLargeMapControl());
      map.addControl(new GOverviewMapControl());
      map.addControl(new GMapTypeControl());
                                        map.setMapType(G_NORMAL_MAP);
      // Create a base icon for all of our markers that specifies the
      // shadow, icon dimensions, etc.
      var baseIcon = new GIcon(G_DEFAULT_ICON);
      baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.png";
      baseIcon.iconSize = new GSize(50, 50);
      baseIcon.shadowSize = new GSize(37, 34);
      baseIcon.iconAnchor = new GPoint(9, 34);
      baseIcon.infoWindowAnchor = new GPoint(9, 2);

      // Creates a marker whose info window displays the letter corresponding
      // to the given index.

    function createMarker(point,placename)
                                        {

       var letteredIcon = new GIcon(baseIcon);
                                          letteredIcon.image = "images/car.png";
       // Set up our GMarkerOptions object
       markerOptions = { icon:letteredIcon };
```

```
        var marker = new GMarker(point, markerOptions);

        return marker;
      }

      // Add  markers to the map at random locations
                                          i=0;
                                    <?php  $places=mysql_query("select * from tbltimeroute
where routeid='$routeid' order by id desc limit 1");
                                          $j=0;
                                          while($j < mysql_num_rows($places))
                                    {
                                    ?>
      var latlng = new GLatLng<?php echo mysql_result($places,$j,"X");?>;
                                          var placename = '<?php echo
mysql_result($places,$j,"routeid");?>';

       allmarkers[i]=createMarker(latlng,placename);
      map.addOverlay(allmarkers[i]);
                                           allmarkers[i].show();
       i++;
                                    <?php
                                          $j++;
                                          }
                                          ?>

                                    }
   }
```

1.3.7 Αναζήτηση

Search.php

Στη σελίδα αυτή ο επισκέπτης μπορεί να κάνει μια αναζήτηση δρομολογίων για συγκεκριμένη ημερομηνία, περιοχή και προορισμό.



Τα στοιχεία της φόρμας αποστέλλονται στη σελίδα availableroutes.php

1.3.8 Διαθέσιμα δρομολόγια

availableroutes.php

Στη σελίδα αυτή εκτελείται ένα query προς τη βάση για να ανακτηθούν τα στοιχεία των δρομολογίων με βάση τα κριτήρια της προηγούμενης φόρμας. Αυτό επιτυγχάνεται με τον παρακάτω php κώδικα:

```
<?php
$categories=mysql_query("select * from tblroute where departdate<= '$traveldate' and arrivedate>='$traveldate' and departure='$departure' and destination='$destination' order by departdate,departtime desc");
$i=0;
while($i < mysql_num_rows($categories))
{
?>
<p><?php
$cities=mysql_query("select * from tblcity where cityid=".mysql_result($categories,$i,"departure"));
```

```
                                        $city1=mysql_result($cities,0,"cityname");
                                                $cities=mysql_query("select * from
tblcity where cityid=".mysql_result($categories,$i,"destination"));
                                        $city2=mysql_result($cities,0,"cityname");
                                                echo
mysql_result($categories,$i,"departdate")."         ".mysql_result($categories,$i,"departtime")."
".$city1." - ".$city2;?> -
                                                <a
href="availableroutedetails.php?routeid=<?php                                   echo
mysql_result($categories,$i,"id");?>">Προεπισκόπηση</a></p><hr/>
                                        <?php
                                                $i++;
                                        }

                                        ?>
```

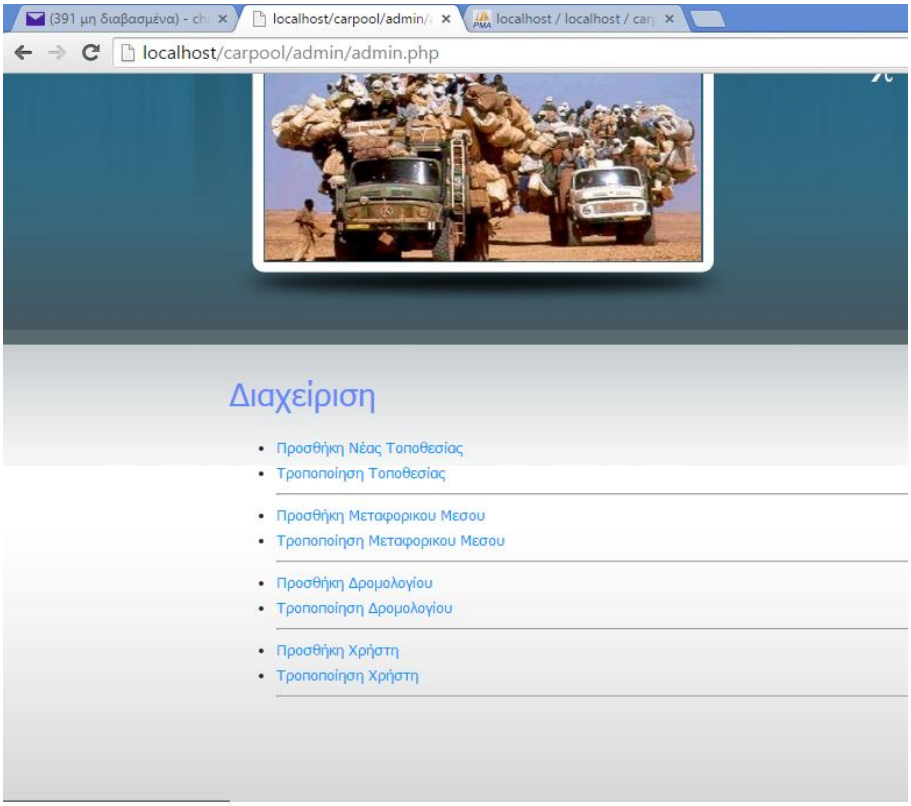
Στη συνέχεια επιλέγοντας ένα από τα δρομολόγια τα στοιχεία αυτού του δρομολογίου αποστέλλονται στη σελίδα availableroutesdetails.php όπου και εμφανίζονται τα στοιχεία του συγκεκριμένου δρομολογίου.

1.3.8 Διαχείριση

Admin.php

Στη σελίδα της διαχείρισης, ο διαχειριστής της εφαρμογής έχει τις εξής επιλογές:

- Τη διαχείριση των τοποθεσιών
- Τη διαχείριση των μεταφορικών μέσων
- Τη διαχείριση χρηστών
- Τη διαχείριση δρομολογίων (Η δυνατότητα υπάρχει πρωτίστως στους χρήστες)



1.3.10 Διαχείριση Τοποθεσιών

Η διαχείριση των τοποθεσιών περιλαμβάνει την προσθήκη νέας τοποθεσίας και την τροποποίηση μιας τοποθεσίας.

newcity.php

Η προσθήκη της τοποθεσίας γίνεται από τη σελίδα newcity.php. Σε αυτή υπάρχει μιαφόρμα με τα βασικά στοιχεία μιας τοποθεσίας. Ο προσδιορισμός της γεωγραφικ΄ζηνθέσης μιας τοποθεσίας γίνεται μέσα από ένα Google Map. Με διπλό κλικ ο διαχειριστής προσδιορίζει τη γεωγραφική θέση της τοποθεσίας. Αυτό επιτυγχάνεται με τη χρήση javascript για Google Maps:

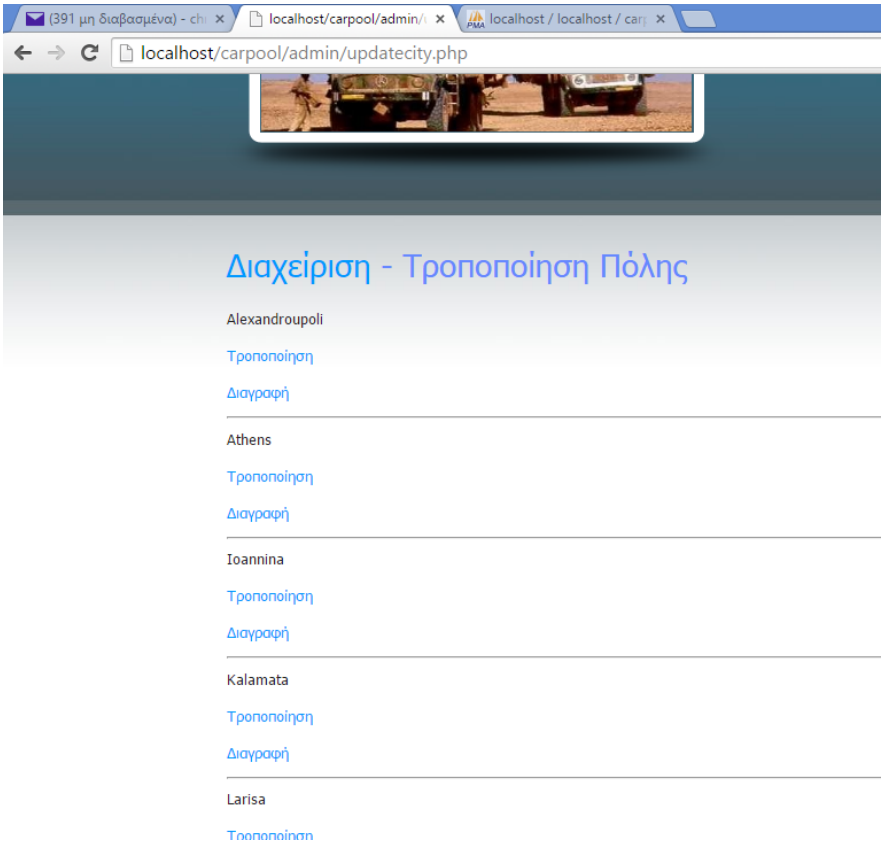
```
   function initialize() {
   //  if (GBrowserIsCompatible()) {
       var map = new GMap2(document.getElementById("map_canvas"));
       map.setCenter(new GLatLng(39.0618, 22.0605),3);
       //Αντικείμενο zoom control
       map.addControl(new GLargeMapControl());
//Αντικείμενο type control
       var mapControl = new GMapTypeControl();
       map.addControl(mapControl);
       GEvent.addListener(map, "click", function() {
         var center = map.getCenter();
         var photoxy=document.getElementById("photoXY");
                                        photoxy.value=center.toString();
       });
    // }
   }
```

Τα στοιχεία της φόρμας υποβάλλονται στη σελίδα addcity.php και με την εκτέλεση ενός insert query προστίθενται στην βάση:

```
include 'carpooldbinfo.php';
$citybasic = $_POST['citybasic'];
$citydetails = $_POST['citydetails'];
$priority=$_POST['priority'];
$cityname= $_POST['cityname' ];
$mapXY= trim($_POST['photoXY']);
mysql_query("insert into tblcity(priority,citydetails,cityname,mapXY,cityzoom,citybasic)
values(".$priority.",'".$citydetails."','".$cityname."','".$mapXY."',9,".$citybasic.")");
mysql_close($link);
```

Updatecity.php

Η τροποποίηση της τοποθεσίας γίνεται από τη σελίδα updatecity.php. Στη σελίδα αυτή εμφανίζονται όλες οι τοποθεσίες και επιλέγοντας μια από αυτές μπορεί να γίνει είτε διαγραφή ή τροποποίηση των στοιχείων.

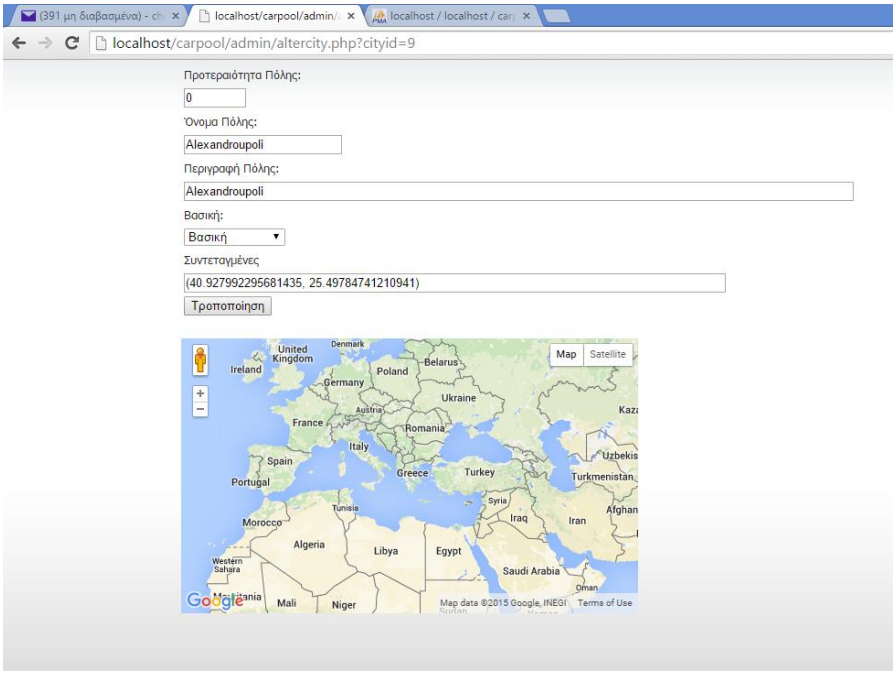


Deletecity.php

Η διαγραφή γίνεται με ένα delete query προς τη βάση

Altercity.php

Επιλέγοντας μια τοποθεσία για τροποποίηση εμφανίζεται μια φόρμα προσυμπληρωμένη με τα τρέχοντα στοιχεία της τοποθεσίας. Αυτό επιτυγχάνεται με την εκτέλεση ενός select query για τη συγκεκριμένη τοποθεσία στον πίνακα tblcity. Τα στοιχεία υποβάλλονται στη σελίδα changecity.php.

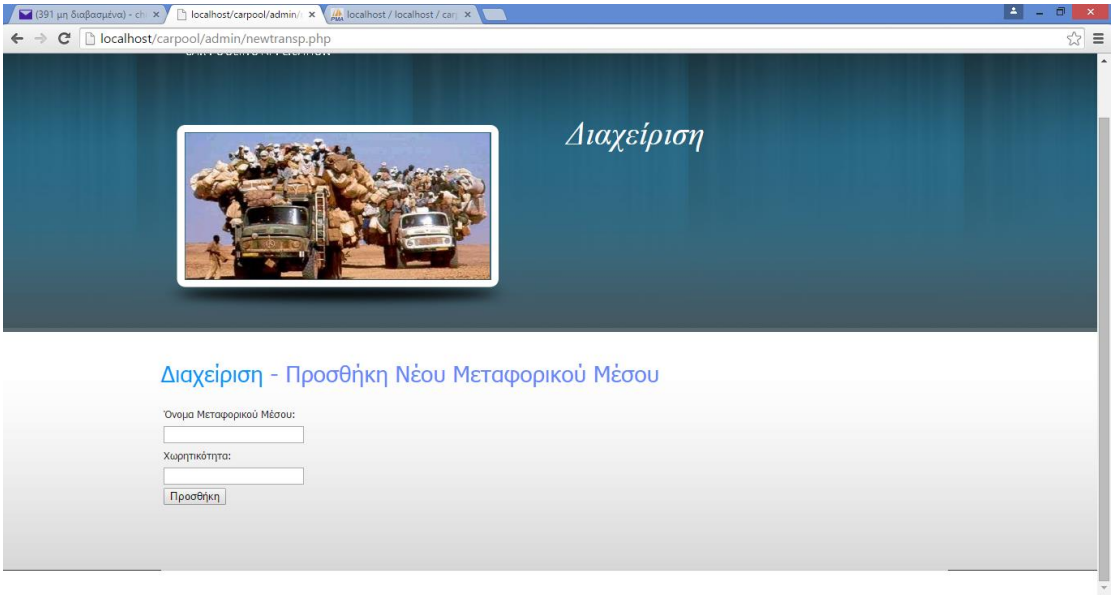


1.3.11 Διαχείριση μεταφορικών μέσων

Η διαχείριση των μεταφορικών μέσων περιλαμβάνει την προσθήκη νέου μέσου και την τροποποίηση του.

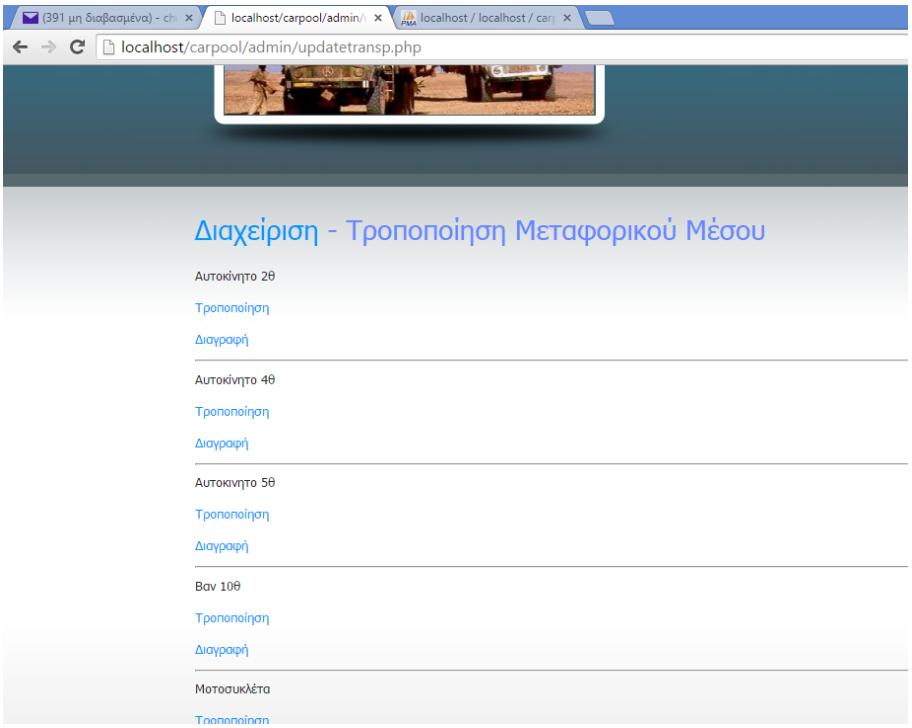
newtransp.php

Η προσθήκη νέου μέσου γίνεται από τη σελίδα newtransp.php. Σε αυτή υπάρχει μια φόρμα με τα βασικά στοιχεία ονόματος και χωρητικότητας.



Updatetransp.php

Η τροποποίηση του μεταφορικού μέσου γίνεται από τη σελίδα Updatetransp.php. Στη σελίδα αυτή εμφανίζονται τα μεταφορικά μέσα και μπορεί να γίνει είτε διαγραφή ή τροποποίηση των στοιχείων. Η διαγραφή γίνεται με ένα delete query προς τη βάση



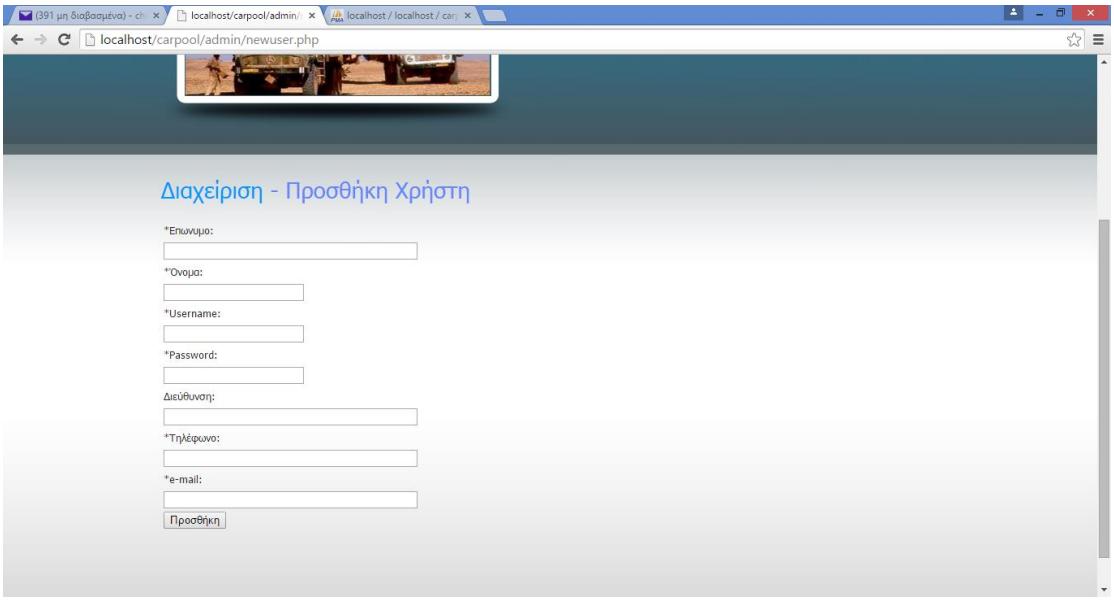
Altertransp.php

Επιλέγοντας ένα μέσο για τροποποίηση εμφανίζεται μια φόρμα προσυμπληρωμένη με τα τρέχοντα στοιχεία της τοποθεσίας. Αυτό επιτυγχάνεται με την εκτέλεση ενός select query Τα στοιχεία υποβάλλονται στη σελίδα changetransp.php, όπου εκτελείται το αντίστοιχο update query στον πίνακα tbltransportation

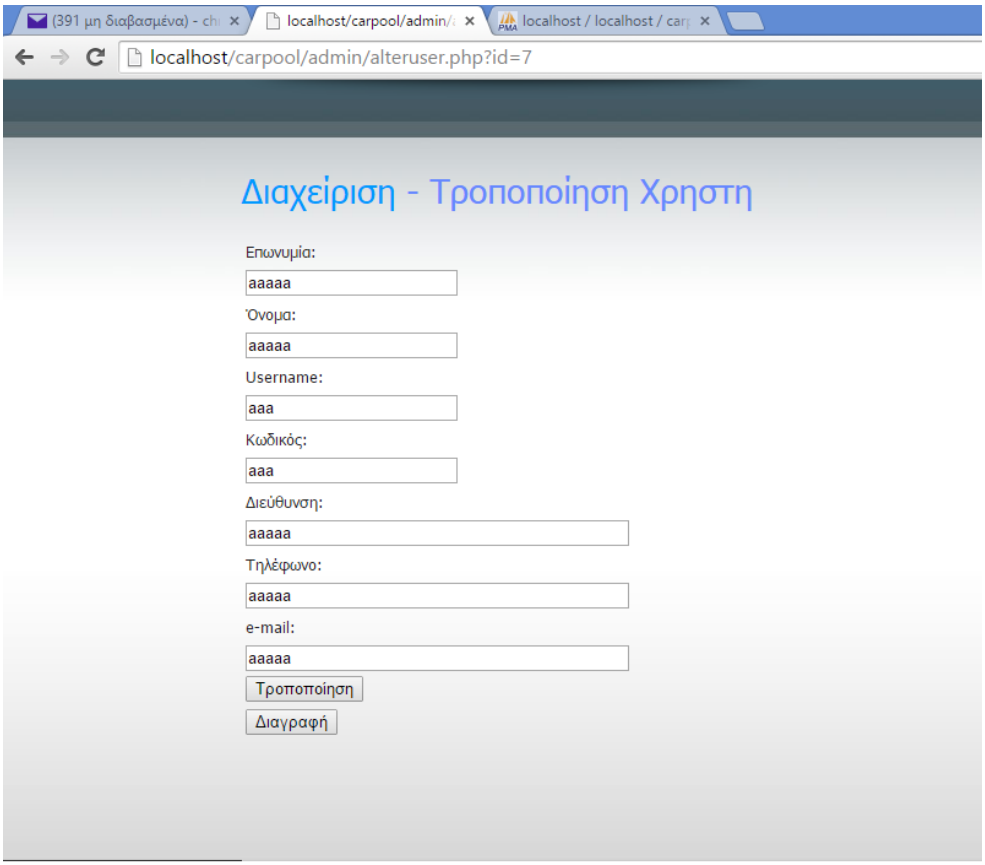
```
mysql_query("update tbltransportation set description='".$description."'
,capacity=".$capacity." where id=".$id);
```

1.3.12 Διαχείριση χρηστών

Η διαχείριση χρηστών περιλαμβάνει την προσθήκη και τροποποίηση των χρηστών. Η πρσοθήκη ενός χρήστη γίνεται από τη σελίδα newuser.php Σε αυτή περιέχεται μια φόρμα όπου συμπληρωνονται τα στοιχεία του χρήστη και αποστέλλονται στη σελίδα adduser.php για να εκτελεστεί το αντίστοιχο insert query προς τη βάση.



Η τροποποίηση στοιχείων χρηστών γίνεται από τη σελίδα updateuser.php. Στη σελίδα αυτή ο διαχειριστής επιλέγει τον χρήστη στον οποίο επιθυμεί να τροποποιήσει στοιχεία του. Στη συνέχεια εμφανίζεται η σελίδα alteruser.php η οποία και περιέχει τα στοιχεία του χρήστη που επιλέχθηκε μέσα σε μια φόρμα. Η ανάκτηση των στοιχείων γίνεται από τον πίνακα tbluser της βάσης με τη χρήση ενός select query. Με την επιλογή «Τροποποίηση» τα στοιχεία αποστέλλονται στη σελίδα change.php όπου και εκτελείται το αντίστοιχο update query. Αν πατηθεί η επιλογή Διαγραφή τότε καλείται ένα delete query στη σελίδα deleteuser.php.



Ο administrator έχει και τη δυνατότητα διαχείρισης των δρομολογίων η οποία και περιγράφεται στην επόμενη ενότητα και περιλαμβάνεται στις επιλογές του εξουσιοδοτημένου χρήστη-μέλους.

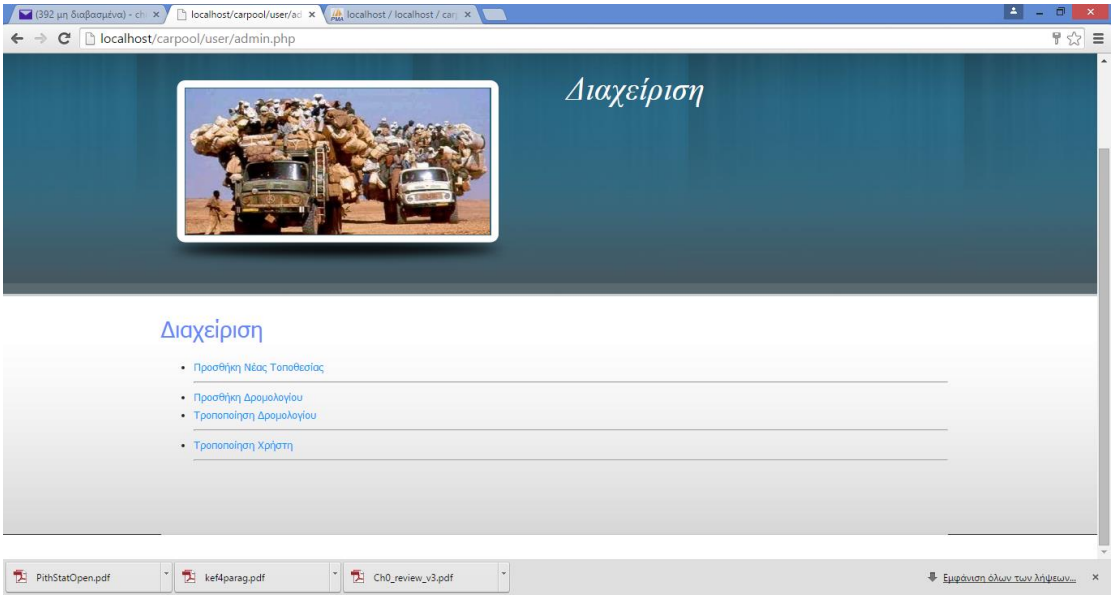
Περιοχή Μελών

Με την επιλογή μέλη ανοίγει η σελίδα member.php όπου εισάγεται το username και password του εξουσιοδοτημένου χρήστη – μέλους ο οποίος μπορεί να αναρτά δρομολόγια και επίσης να ζητάει να επιβιβαστεί σε ένα δρομολόγιο μέσω της android εφαρμογής.

Η ταυτοποίηση του χρήστη γίνεται από τη σελίδα login.php με τον παρακάτω κώδικα:

```
$username='';
$username=addslashes($_POST['username']);
$password='';
$password=addslashes($_POST['password']);


$query=mysql_query("select * from tbluser where username='".$username."' and pass='".$password."'");
if ((mysql_num_rows($query)==0))
   {
                                    mysql_close($link);
                                    ?>
                                    <script language="javascript">
                                    window.alert("Λάθος κωδικοί");
                                    window.location="./index.php";
                                    </script>
<?php    }
elseif(mysql_result($query,0,'usertype')=='usertp')
{
$_SESSION['currentuser']=$username ;
$_SESSION['usertype']="usertp" ;
$_SESSION['memberid']=mysql_result($query,0,"id") ;
?>

                                    <script language="javascript">
                                    window.alert("Συνδεθήκατε επιτυχώς");
                                    window.location="./user/admin.php";
                                    </script>
<?php }
```

```
else
{
$_SESSION['currentuser']=$username ;
$_SESSION['usertype']="membertp" ;
$_SESSION['memberid']=mysql_result($query,0,"id") ;
?>

                                        <script language="javascript">
                                        window.alert("Συνδεθήκατε επιτυχώς");
                                        window.location="./member/index.php";
                                        </script>
<?php }
```



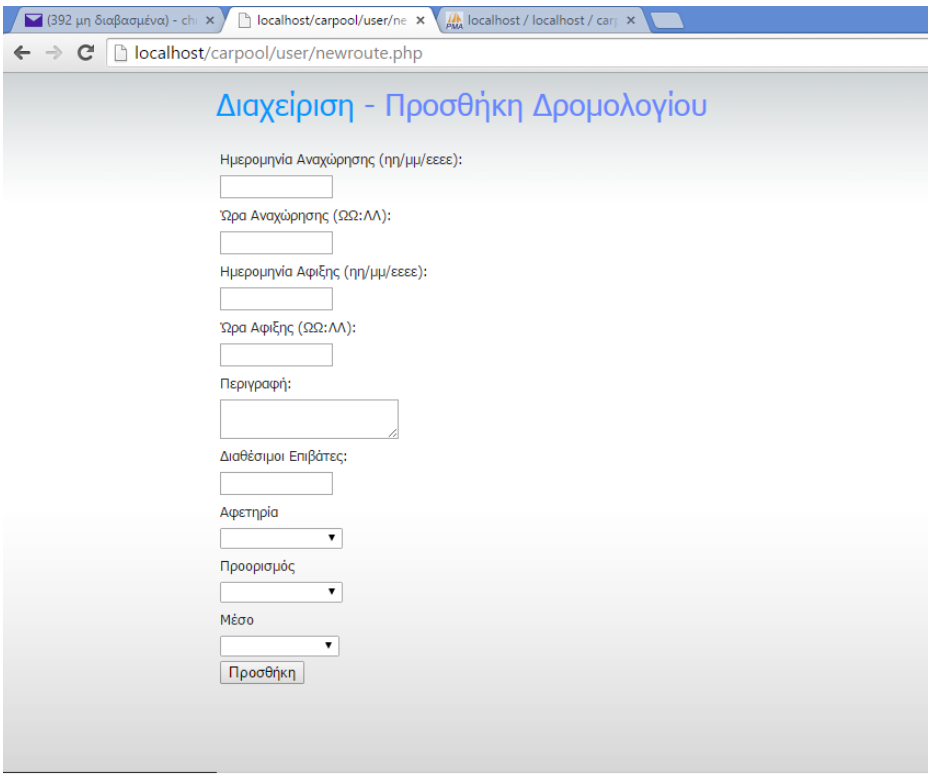
Οι επιλογές διαχείρισης τοποθεσίας και στοιχείων χρήστη είναι παρόμοιες με τον τρόπο που αυτές περιγράφτηκαν στην ενότητα του διαχειριστή.

1.3.12 Διαχείριση Δρομολογίων

Στη διαχείριση δρομολογίων ο χρήστης μπορεί είτε να προσθέσει ή να τροποποιήσει ένα δρομολόγιο που ο ίδιος έχει προσθέσει.

Newroute.php

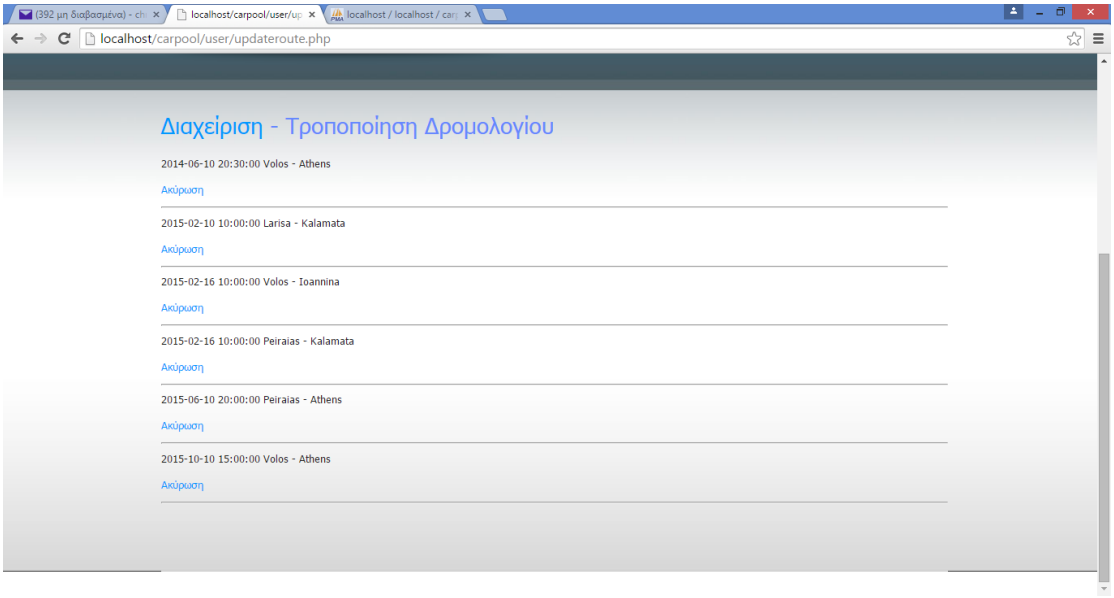
Στη σελίδα αυτή εμφανίζεται μια φόρμα εισαγωγής των στοιχείων του δρομολογίου που περιλαμβάνει την ημερομηνία και ώρα αναχώρησης και άφιξης, την αφετηρία και των προορισμό, το μέσο που θα χρησιμοποιηθεί, την περιγραφή και τις διαθέσιμες θέσεις. Με την επιλογή Προσθήκη τα στοιχεία αποστέλλονται στη σελίδα addroute.php όπου και εκτελείται το αντίστοιχο insert query.



Τροποποίηση Δρομολογίου

Με την επιλογή Τροποποίηση Δρομολογίου, ο χρήστης εισέρχεται στη σελίδα updateroute.php όπου εμφανίζονται με ημερολογιακή σειρά τα δρομολόγια που έχει προσθέσει ο ίδιος. Αυτό επιτυγχάνεται με ένα select query και τον ακόλουθο κώδικα ο οποίος ανακτά τα στοιχεία των δρομολογίων και των τοποθεσιών:

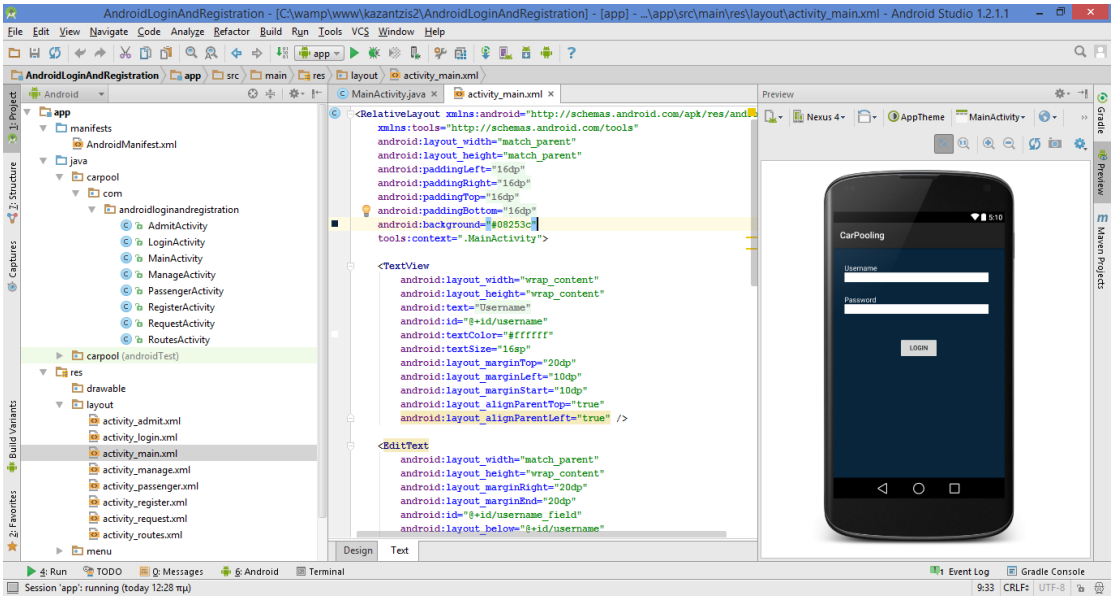
```
<?php $categories=mysql_query("select * from tblroute where driverid=".$_SESSION['memberid']." order by departdate,departtime desc");
$i=0;
while($i < mysql_num_rows($categories))
{
?>
<p><?php
$cities=mysql_query("select * from tblcity where cityid=".mysql_result($categories,$i,"departure"));

$city1=mysql_result($cities,0,"cityname");
$cities=mysql_query("select * from tblcity where cityid=".mysql_result($categories,$i,"destination"));

$city2=mysql_result($cities,0,"cityname");
echo mysql_result($categories,$i,"departdate")."  ".mysql_result($categories,$i,"departtime")." ".$city1." - ".$city2;?></p>
<p><a href="cancelroute.php?section=<?php echo mysql_result($categories,$i,"id");?>">Ακύρωση</a></p><hr/>
<?php
$i++;
}
?>
```

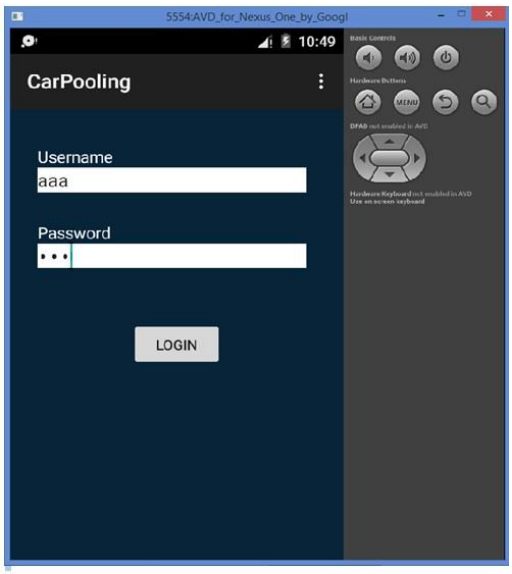


Για κάθε δρομολόγιο ο χρήστης μπορεί να πατήσει την επιλογή Ακύρωση και το δρομολόγιο θα διαγραφεί από τα δρομολόγια καθώς και από τις κρατήσεις των συνεπιβατών.

2. Android Εφαρμογή

Η Android εφαρμογή αναπτύχθηκε με τη χρήση του Android Studio v.1.2.1.1. Η αρχική οθόνη είναι η main activity η οποία χρησιμοποιείται για την είσοδο του χρήστη στην εφαρμογή. Το UI περιγράφεται από το activity_main.xml και περιέχει κυρίως δύο πεδία username και password και ένα κουμπί login.





Τα στοιχεία που εισάγονται στην φόρμα αποστέλλονται στη σελίδα για να γίνει η ταυτοποίηση του χρήστη και η ανάκτηση του id του.

```
//listener for the login button
    loginButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            enteredUsername = username.getText().toString();
            String enteredPassword = password.getText().toString();

            if(enteredUsername.equals("") || enteredPassword.equals("")){
                Toast.makeText(MainActivity.this, "Username or password must be filled", Toast.LENGTH_LONG).show();
                return;
            }
            if(enteredUsername.length() <= 1 || enteredPassword.length() <= 1){
                Toast.makeText(MainActivity.this, "Username or password length must be greater than one", Toast.LENGTH_LONG).show();
                return;
            }
            // request authentication with remote server4 sending username and password to androidlogin.php
            AsyncDataClass asyncRequestObject = new AsyncDataClass();
            asyncRequestObject.execute(serverUrl, enteredUsername, enteredPassword);
        }
    });
```

Στη συνέχεια καλέιται η AsyncDataClass asyncRequestObject η οποία στέλνει ένα request στη σελίδα *androidlogin.php και παραλαμβάνει το response που είναι ένα json result*

```
private class AsyncDataClass extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {
```

```
        HttpParams httpParameters = new BasicHttpParams();
        HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
        HttpConnectionParams.setSoTimeout(httpParameters, 5000);

        HttpClient httpClient = new DefaultHttpClient(httpParameters);
        HttpPost httpPost = new HttpPost(params[0]);

        String jsonResult = "";
        try {
            List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
            nameValuePairs.add(new BasicNameValuePair("username", params[1]));
            nameValuePairs.add(new BasicNameValuePair("password", params[2]));
            httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

            HttpResponse response = httpClient.execute(httpPost);
            jsonResult = inputStreamToString(response.getEntity().getContent()).toString();

        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return jsonResult;
    }
```

Κατόπιν αφού γίνει parsing του επιστρεφόμενου json αποστέλλεται το username στην κλάση

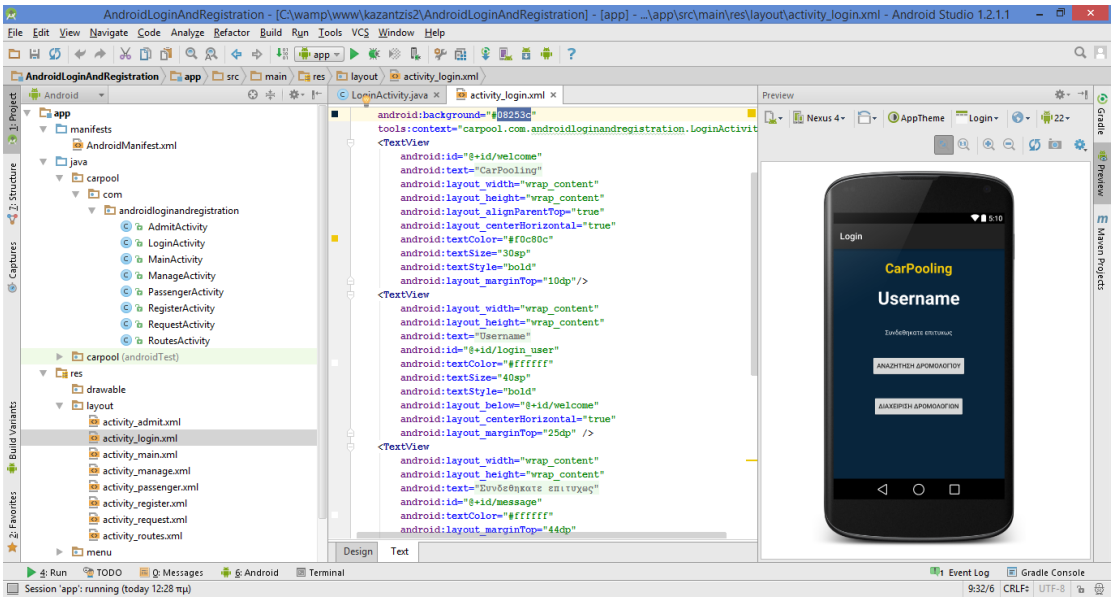
LoginActivity ακολουθούμενο από ένα μήνυμα επιτυχούς login.

```
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    System.out.println("Resulted Value: " + result);
    if(result.equals("") || result == null){
        Toast.makeText(MainActivity.this, "Server connection failed",
Toast.LENGTH_LONG).show();
        return;
    }
    int jsonResult = returnParsedJsonObject(result);
    if(jsonResult == 0){
        Toast.makeText(MainActivity.this, "Invalid username or password",
Toast.LENGTH_LONG).show();
        return;
```

```
    }
    if(jsonResult == 1){
        Intent intent = new Intent(MainActivity.this, LoginActivity.class);
        intent.putExtra("USERNAME", enteredUsername);
        intent.putExtra("MESSAGE", "You have been successfully login");
        startActivity(intent);
    }
}
```

2.1 LoginActivity

Η κλάση LoginActivity περιγράφεται από το activity_login.xml και περιέχει το username του χρήστη, ;το μήνυμα επιτυχούς εισόδου και τα κουμπιά επιλογών «Αναζήτηση Δρομολογίου» και «Διαχείριση Δρομολογίου»



Η κλάση παραλαμβάνει τα username και message και τα εμφανίζει στην οθόνη. Επίσης προσθετει τα κουμπιά και 2 listeners γι αυτά.

```
Intent intent = getIntent();
Bundle intentBundle = intent.getExtras();
loggedUser = intentBundle.getString("USERNAME");

String message = intentBundle.getString("MESSAGE");
//get username and message
TextView loginUsername = (TextView)findViewById(R.id.login_user);
TextView successMessage = (TextView)findViewById(R.id.message);
loginUsername.setText(loggedUser);
//show message
successMessage.setText(message);
//listener for button for managing a route
driverButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AsyncDataClass asyncRequestObject = new AsyncDataClass();
        asyncRequestObject.execute(serverUrl, loggedUser);

    }
});
//listener for button for searching for a route
searchButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        AsyncDataClass2 asyncRequestObject = new AsyncDataClass2();
        asyncRequestObject.execute(serverUrl2);

    }
});
```

Πατώντας το κουμπί Αναζήτηση Δρομολογίου καλείται η κλάση AsyncDataClass2 asyncRequestObject η οποία αποστέλλει ένα request στη σελίδα androidcities.php και παραλαμβάνει σαν response ένα json με τα ονόματα των τοποθεσιών .

```
private class AsyncDataClass2 extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {

        HttpParams httpParameters = new BasicHttpParams();
        HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
        HttpConnectionParams.setSoTimeout(httpParameters, 5000);

        HttpClient httpClient = new DefaultHttpClient(httpParameters);
        HttpPost httpPost = new HttpPost(params[0]);

        String jsonResult = "";
        try {

            HttpResponse response = httpClient.execute(httpPost);
            jsonResult = inputStreamToString(response.getEntity().getContent()).toString();

        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return jsonResult;
    }
```

Το json αφού γίνει parsing αποστέλλεται στην κλάση RegisterActivity σαν message με το περιεχόμενο result.

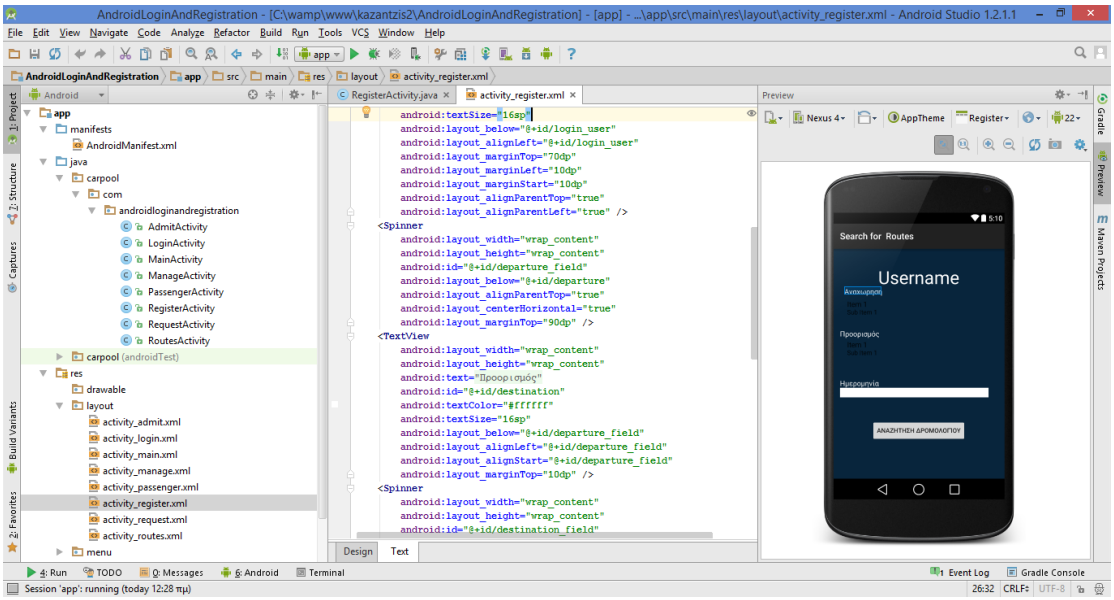
```
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    System.out.println("Resulted Value: " + result);
    if(result.equals("") || result == null){
        Toast.makeText(LoginActivity.this, "Server connection failed",
Toast.LENGTH_LONG).show();
        return;
    }
    Intent intent = new Intent(LoginActivity.this, RegisterActivity.class);
    intent.putExtra("USERNAME", loggedUser);
    intent.putExtra("MESSAGE", result);
    startActivity(intent);

}
```

2.2 RegisterActivity

Το UI της κλάσης ορίζεται από το activity _register.xml και περιέχει κυρίως δύο drop down menus, ένα text box για την εισαγωγή της ημερομηνίας αναζήτησης και ένα button για την υποβολή της αναζήτησης. Τα dropdown menu εμφανίζουν τις επιλογές από το message που έχει σταλεί από την προηγούμενη κλάση με τη βοήθεια του παρακάτω κώδικα:

```
//creates drop down menu for route options send from register activity
spinner1 = (Spinner)findViewById(R.id.departure_field);
try {
   stringArray1 = new ArrayList<String>();
   JSONArray arr = new JSONArray(message);
   for(int i = 0; i < arr.length(); i++){
      stringArray1.add(arr.getJSONObject(i).getString("cityid")+"
"+arr.getJSONObject(i).getString("details"));
   }
} catch (Exception e) {
   e.printStackTrace();
}


ArrayAdapter<String> adapter = new ArrayAdapter<String>(RegisterActivity.this,
      android.R.layout.simple_spinner_item,stringArray1);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner1.setAdapter(adapter);

//creates drop down menu for route options send from register activity
spinner2 = (Spinner)findViewById(R.id.destination_field);
try {
   stringArray2 = new ArrayList<String>();
   JSONArray arr = new JSONArray(message);
   for(int i = 0; i < arr.length(); i++){
      stringArray2.add(arr.getJSONObject(i).getString("cityid")+"
"+arr.getJSONObject(i).getString("details"));
   }
} catch (Exception e) {
   e.printStackTrace();
}


ArrayAdapter<String> adapter2 = new ArrayAdapter<String>(RegisterActivity.this,
      android.R.layout.simple_spinner_item,stringArray2);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
```

spinner2.setAdapter(adapter);



Στη συνέχεια μέσα στην κλάση προστίθεται ένας listener για το κουμπί αναζήτησης το οποίο και καλεί την AsyncDataClass asyncRequestObject
για την αποστολή ενός request προς τη σελίδα adnroidroutes.php η οποία εκτελεί ένα select query στον πίνακα των δρομολογίων για να βρεθούν τα διαθέσιμα δρομολόγια από την αφετηρία προς τον προορισμό που επιλέχθηκε και την ημερομηνία που εισήγαγε ο χρήστης.

```
Button signUpButton = (Button)findViewById(R.id.retrieve);
//creates listener for search button
signUpButton.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

       String enteredDep = spinner1.getSelectedItem().toString();
       String enteredDeparture = enteredDep.substring(0, enteredDep.indexOf(':'));

       String enteredDes = spinner2.getSelectedItem().toString();
       String enteredDestination = enteredDes.substring(0, enteredDes.indexOf(':'));
       String enteredTraveldate = traveldate.getText().toString();

       if(enteredDeparture.equals("") || enteredDestination.equals("") ||
enteredTraveldate.equals("")){
           Toast.makeText(RegisterActivity.this, "Invalid value",
Toast.LENGTH_LONG).show();
           return;
```

```
        }
        if(enteredDeparture.length() < 1 || enteredDestination.length() < 1){
            Toast.makeText(RegisterActivity.this, "Invalid value",
Toast.LENGTH_LONG).show();
            return;
        }
        // request authentication with remote server
        AsyncDataClass asyncRequestObject = new AsyncDataClass();
        asyncRequestObject.execute(serverUrl, enteredDeparture, enteredDestination,
enteredTraveldate);
    }
});
```

Η κλάση επιστρέφει ένα json που περιέχει πληροφορίες για τα δρομολόγια που εκτελούνται στη συγκεκριμένη διαδρομή τη συγκεκριμένη ημερομηνία.

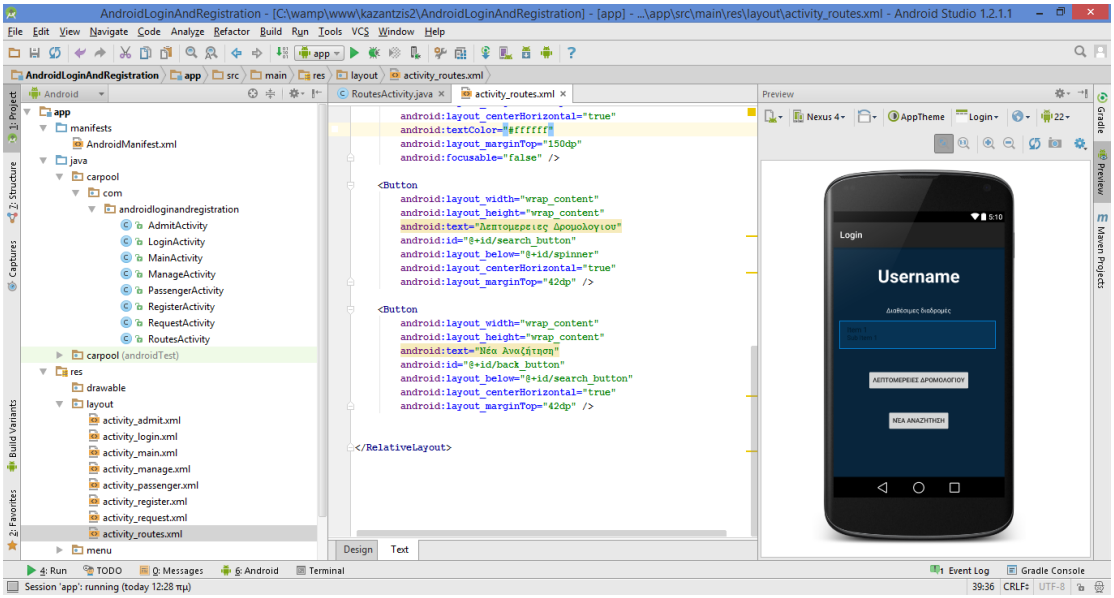
```
private class AsyncDataClass extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {

        HttpParams httpParameters = new BasicHttpParams();
        HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
        HttpConnectionParams.setSoTimeout(httpParameters, 5000);

        HttpClient httpClient = new DefaultHttpClient(httpParameters);
        HttpPost httpPost = new HttpPost(params[0]);

        String jsonResult = "";
        try {
            List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
            //sends traveldate, destination, departure  parameters to webpage androidroutes.php
            nameValuePairs.add(new BasicNameValuePair("departure", params[1]));
            nameValuePairs.add(new BasicNameValuePair("destination", params[2]));
            nameValuePairs.add(new BasicNameValuePair("traveldate", params[3]));
            httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
            //gets a json response from webpage androidroutes.php
            HttpResponse response = httpClient.execute(httpPost);
            jsonResult = inputStreamToString(response.getEntity().getContent()).toString();
            System.out.println("Returned Json object " + jsonResult.toString());

        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        //returns json with the available routes
```

```
        return jsonResult;
    }
```
Μετά την επιτυχή εκτέλεση του αιτήματος τα αποτελέσματα αποστέλλονται μέσα από ένα μήνυμα στην κλάση RoutesActivity

```
 protected void onPostExecute(String result) {
    super.onPostExecute(result);
    System.out.println("Resulted Value: " + result);
    if(result.equals("") || result == null){
        Toast.makeText(RegisterActivity.this, "Server connection failed",
Toast.LENGTH_LONG).show();
        return;
    }

        Intent intent = new Intent(RegisterActivity.this, RoutesActivity.class);
    //sends resulted routes as message to routes_activity
        intent.putExtra("MESSAGE",result);
       intent.putExtra("USERNAME", loggedUser);
        startActivity(intent);
 }
```

2.3 RoutesActivity

Η κλάση δέχεται ως μεταβλητή με όνομα message το αποτέλεσμα που περιέχει όλα τα δρομολόγια και τα εμφανίζει μέσα σε ένα drop down menu για να μπορεί να γίνει η επιλογή διαθέσιμου δρομολογίου. Το UI της κλάσης περιγράφεται από το activity_routes.xml και περιλαμβάνει ένα στοιχείο spinner για την εμφάνιση των δρομολογίων και δύο buttons, το ένα για την επιλογή του δρομολογίου και το άλλο για την επιστροφή στην αναζήτηση δρομολογίων.



Η κλάση λαμβάνει τα αποτελέσματα από την κλάση *register activity* και αφού μετατρέψει το json αποτέλεσμα σε πίνακα (array) γεμίζει το στοιχεί spinner. Στη συνέχεια προστίθενται δύο listeners, ένας για κάθε κουμπί. Σε περίπτωση που επιλεχθεί το κουμπί Λεπτομέρειες Δρομολογίου τότε το id του δρομολογίου αποστέλλεται η επιλεγμένη τιμή ως message στην κλάση RequestActivity

```
Button searchButton = (Button)findViewById(R.id.search_button);
Button backButton = (Button)findViewById(R.id.back_button);
Intent intent = getIntent();
Bundle intentBundle = intent.getExtras();
final String loggedUser = intentBundle.getString("USERNAME");
//routes resulted from register activity
String message = intentBundle.getString("MESSAGE");
JSONObject resultObject = null;
//creates drop down menu for route options send from register activity
spinner = (Spinner)findViewById(R.id.spinner);
try {
  stringArray = new ArrayList<String>();
   JSONArray arr = new JSONArray(message);
   for(int i = 0; i < arr.length(); i++){
      stringArray.add(arr.getJSONObject(i).getString("routeid")+"
```

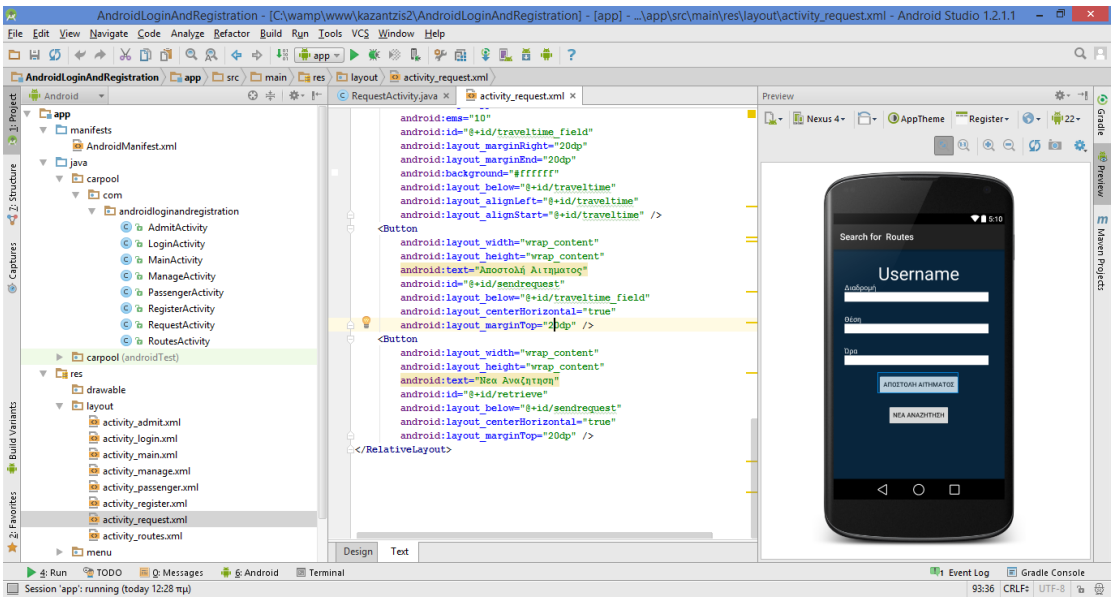
```
"+arr.getJSONObject(i).getString("details"));
    }
} catch (Exception e) {
    e.printStackTrace();
}


ArrayAdapter<String> adapter = new ArrayAdapter<String>(RoutesActivity.this,
        android.R.layout.simple_spinner_item,stringArray);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(adapter);
//spinner.setOnItemSelectedListener((AdapterView.OnItemSelectedListener) this);
TextView loginUsername = (TextView)findViewById(R.id.login_user);
TextView successMessage = (TextView)findViewById(R.id.message);
loginUsername.setText(loggedUser);

backButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(RoutesActivity.this, RegisterActivity.class);
        intent.putExtra("USERNAME",loggedUser);
        startActivity(intent);
    }
});
searchButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        selectedtext = spinner.getSelectedItem().toString();
        Intent intent = new Intent(RoutesActivity.this, RequestActivity.class);
        intent.putExtra("MESSAGE",selectedtext);
        intent.putExtra("USERNAME",loggedUser);
        startActivity(intent);
    }
});
```

2.4 RequestActivity

Η κλάση RequestActivity εμφανίζει τις λεπτομέρειες του δρομολογίου και δίνει τη δυνατότητα στο χρήστη να συμπληρώσει τη θέση (πχ χλμ) και την ώρα που υπολογίζει ότι θα βρίσκεται σε εκείνη τη θέση και στη συνέχεια να υποβάλλει αίτημα για επιβίβαση στο συγκεκριμένο σημείο. Το UI της κλάσης ορίζεται από activityrequest.xml και αποτελείται

από 3 text boxes όπου ο χρήστης μπορεί να εισάγει τιμές καθώς , από ένα button επιστροφής στη φόρμα αναζήτησης και από ένα button υποβολής του αιτήματος.

Πατώντας την επιλογή Αποστολή Αιτήματος, καλείται η AsyncDataClass asyncRequestObject η οποία και αποστέλλει τα στοιχεία που έχει εισάγει ο χρήστης μέσω ενός request στη σελίδα androidrequests.php.




```
routeId = (EditText)findViewById(R.id.route_field);
position = (EditText)findViewById(R.id.position_field);
traveltime = (EditText)findViewById(R.id.traveltime_field);
Button signUpButton = (Button)findViewById(R.id.sendrequest);
Button newSearchButton = (Button)findViewById(R.id.retrieve);
Intent intent = getIntent();
Bundle intentBundle = intent.getExtras();
String message = intentBundle.getString("MESSAGE");
loggedUser = intentBundle.getString("USERNAME");
TextView loginUsername = (TextView)findViewById(R.id.login_user);
loginUsername.setText(loggedUser);
```

```
routeId.setText(message);

newSearchButton.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {

      Intent intent = new Intent(RequestActivity.this, RegisterActivity.class);
      intent.putExtra("USERNAME", loggedUser);
      startActivity(intent);
   }
});
signUpButton.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      String enteredRoute = routeId.getText().toString();
      String enteredPosition = position.getText().toString();
      String enteredTraveltime = traveltime.getText().toString();
      String userid = loggedUser;

      if(enteredRoute.equals("") || userid.equals("")){
         Toast.makeText(RequestActivity.this, "Invalid value",
Toast.LENGTH_LONG).show();
         return;
      }

      // request authentication with remote server
      AsyncDataClass asyncRequestObject = new AsyncDataClass();
      asyncRequestObject.execute(serverUrl, enteredRoute,userid,
enteredPosition,enteredTraveltime);
   }
});
```

Στέλνοντας τις παραμέτρους routeid, userid, position και traveltime στη σελίδα εκτελείται ένα insert query προς τον πίνακα tblreservation αφού πρώτα γίνει ο απαιτούμενος έλεγχος.

```
HttpParams httpParameters = new BasicHttpParams();
HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
HttpConnectionParams.setSoTimeout(httpParameters, 5000);

HttpClient httpClient = new DefaultHttpClient(httpParameters);
HttpPost httpPost = new HttpPost(params[0]);

String Result = "";
try {
```

```
    List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
    nameValuePairs.add(new BasicNameValuePair("routeid", params[1]));
    nameValuePairs.add(new BasicNameValuePair("userid", params[2]));
    nameValuePairs.add(new BasicNameValuePair("position", params[3]));
    nameValuePairs.add(new BasicNameValuePair("traveltime", params[4]));
    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

    HttpResponse response = httpClient.execute(httpPost);
    Result = inputStreamToString(response.getEntity().getContent()).toString();
    System.out.println("Returned  " + Result.toString());

} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
return Result;
```

Στη συνέχεια ο χρήστης επιστρέφει στην αρχική οθόνη.

```
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    System.out.println("Resulted Value: " + result);
    if(result.equals("") || result == null){
        Toast.makeText(RequestActivity.this, "Server connection failed",
Toast.LENGTH_LONG).show();
        return;
    }

        Intent intent = new Intent(RequestActivity.this, LoginActivity.class);
        intent.putExtra("MESSAGE",result);
     intent.putExtra("USERNAME", loggedUser);
        startActivity(intent);
}
```

2.5 Διαχείριση Δρομολογίων

Στη αρχική οθόνη αν ο χρήστης επιλέξει τη διαχείριση δρομολογίων, ενεργοποιείται η κλάση ManageActivity. Αυτό γίνεται αφού κληθεί ο αντίστοιχος listener για το κουμπί Διαχείριση Δρομολογίων. Κατά την επιλογή καλείται μια συνάρτηση με την οποία αποστέλλεται το id του χρήστη στη σελίδα και επιστρέφεται ένα σύνολο δρομολογίων που έχουν αναρτηθεί από τον συγκεκριμένο χρήστη.

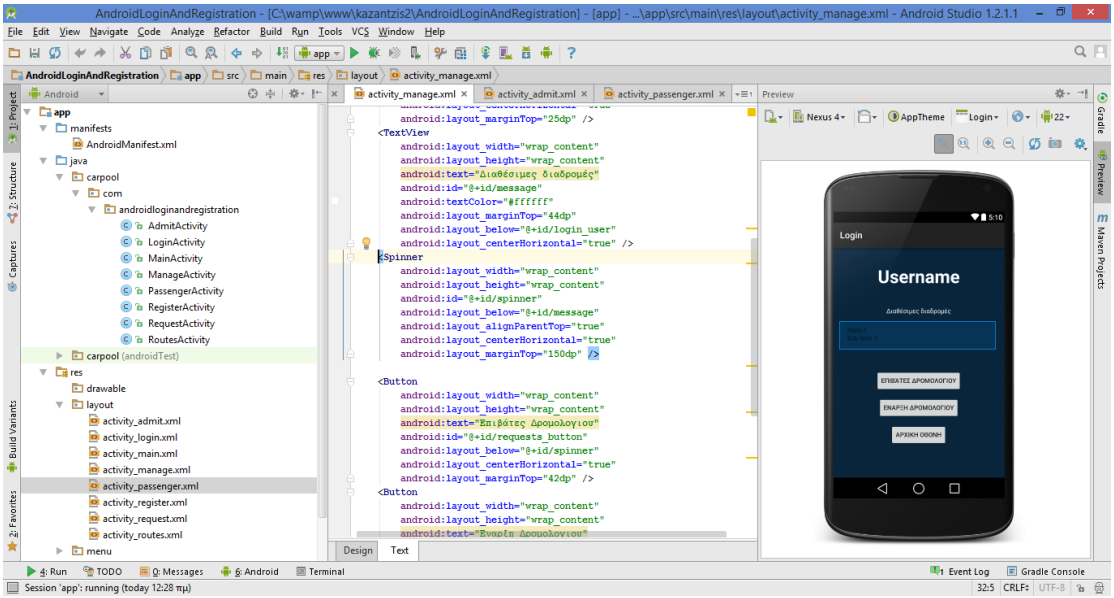
ManageActivity

Στην κλάση αυτή ο χρήστης μπορεί να επιλέξει ένα από τα δρομολόγια τα οποία αυτός έχει προσθέσει. Τα δρομολόγια εμφανίζονται σε ένα drop down menu το οποίο λαμβάνει ως τιμές το routeid και details του δρομολογίου από το json που αποστέλεται από την κλάση loginActivity. Το UI της κλάσης ορίζεται από το activity_manage.xml και περιλαμβάνει εκτός από το dropdown menu με τα δρομολόγια, 3 buttons για την επιλογή «Έναρξη Δρομολογίου» ή «Επιβατες Δρομολογίου» ή «Αρχική Οθόνη»

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_manage);
Button requestsButton = (Button)findViewById(R.id.requests_button);
Button startButton = (Button)findViewById(R.id.start_button);
Button backButton = (Button)findViewById(R.id.back_button);
Intent intent = getIntent();
Bundle intentBundle = intent.getExtras();
loggedUser = intentBundle.getString("USERNAME");
//routes resulted from previous activity
String message = intentBundle.getString("MESSAGE");
```

```
JSONObject resultObject = null;
//creates drop down menu for route options send from register activity
spinner = (Spinner)findViewById(R.id.spinner);
 try {
  stringArray = new ArrayList<String>();
   JSONArray arr = new JSONArray(message);
   for(int i = 0; i < arr.length(); i++){
      stringArray.add(arr.getJSONObject(i).getString("routeid")+"
"+arr.getJSONObject(i).getString("details"));
   }
} catch (Exception e) {
   e.printStackTrace();
}
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(ManageActivity.this,
      android.R.layout.simple_spinner_item,stringArray);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(adapter);
```



Για την λειτουργία των buttons ορίζονται οι αντίστοιχοι listeners:

```
backButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(ManageActivity.this, LoginActivity.class);
        intent.putExtra("USERNAME", loggedUser);
        startActivity(intent);
    }
});
requestsButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        selectedtext = spinner.getSelectedItem().toString();

        // request authentication with remote server
        AsyncDataClass asyncRequestObject = new AsyncDataClass();
        asyncRequestObject.execute(serverUrl, selectedtext);


    }
});
startButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        selectedtext = spinner.getSelectedItem().toString();

        // request authentication with remote server
        AsyncDataClass2 asyncRequestObject = new AsyncDataClass2();
        asyncRequestObject.execute(serverUrl2, selectedtext);


    }
});
```

Επιλέγοντας την «Αρχική Οθόνη», ο χρήστης μεταβαίνει στην αρχική οθόνη. Επιλέγοντας «Έναρξη Δρομολογίου», τα στοιχεία του δρομολογίου αποστέλλονται στη AsyncDataClass2 asyncRequestObject.

Εκεί με τη χρήση μιας συνάρτησης προσομοίωσης των συντεταγμένων GPS, παράγεται μια γεωγραφική θέση του οχήματος Χ, η οποία και αποστέλλεται ανά 1 δευτερόλεπτο στη σελίδα insertGPS.php και στον πίνακα tbltimeroute όπου κρατιέται για κάθε δευτερόλεπτο η θέση για ένα δρομολόγιο.

```
    private class AsyncDataClass2 extends AsyncTask<String, Void, String> {
        private String getPosition()
        {
            Random r = new Random();
            String pos;
```

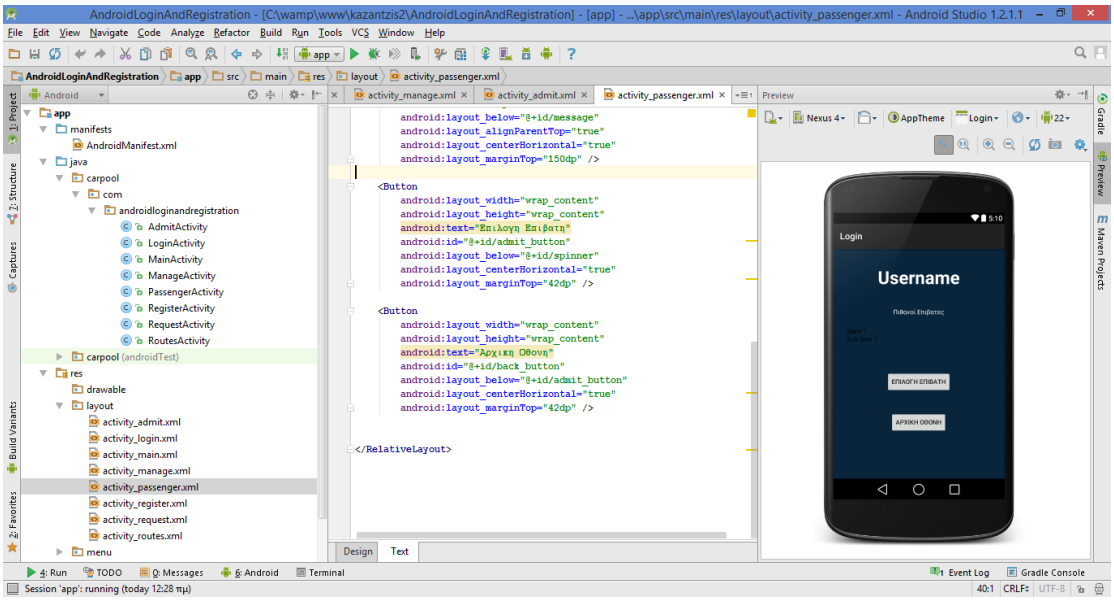
```
        Double i1 = r.nextInt(50) /100.0+37.962556151367316;
        pos="("+Double.toString(i1)+", 23.740721557617224)";
        return pos;
    }
    @Override
    protected String doInBackground(String... params) {

        HttpParams httpParameters = new BasicHttpParams();
        HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
        HttpConnectionParams.setSoTimeout(httpParameters, 5000);

        HttpClient httpClient = new DefaultHttpClient(httpParameters);
        HttpPost httpPost = new HttpPost(params[0]);
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
        String jsonResult = "";
        for(;;)
//send position to insertGPS every 1 second of time
        try {
            Thread.sleep(1000);

            nameValuePairs.add(new BasicNameValuePair("routeid", params[1]));
            nameValuePairs.add(new BasicNameValuePair("X",getPosition()));
            httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

            HttpResponse response = httpClient.execute(httpPost);


        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        catch(Exception e)
        {
            e.printStackTrace();

        }

    }
```

Επιλέγοντας κάποιο από τα δρομολόγια και πατώντας την επιλογή «Επιβάτες Δρομολογίου», ο χρήστης μεταβαίνει στην κλάση PassengerActivity όπου εμφανίζονται οι επιβάτες που έχουν κάνει αίτημα επιβίβασης. Αυτό επιτυγχάνεται με την κλήση της κλάσης η οποία στέλνει ένα request στη σελίδα androidpassengers.php και λαμβάνει σαν response τους πιθανούς επιβάτες που έχουν κάνει αίτημα για επιβίβαση στη συγκεκριμένη διαδρομή.

```
protected String doInBackground(String... params) {

   HttpParams httpParameters = new BasicHttpParams();
   HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
   HttpConnectionParams.setSoTimeout(httpParameters, 5000);

   HttpClient httpClient = new DefaultHttpClient(httpParameters);
   HttpPost httpPost = new HttpPost(params[0]);

   String jsonResult = "";
   try {
      List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
      nameValuePairs.add(new BasicNameValuePair("routeid", params[1]));
      httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

      HttpResponse response = httpClient.execute(httpPost);
      jsonResult = inputStreamToString(response.getEntity().getContent()).toString();

   } catch (ClientProtocolException e) {
      e.printStackTrace();
   } catch (IOException e) {
      e.printStackTrace();
   }
   return jsonResult;
}
@Override
protected void onPreExecute() {
   super.onPreExecute();
}
@Override
protected void onPostExecute(String result) {
   super.onPostExecute(result);
   System.out.println("Resulted Value: " + result);
   if(result.equals("") || result == null){
      Toast.makeText(ManageActivity.this, "Server connection failed",
Toast.LENGTH_LONG).show();
      return;
   }


    Intent intent = new Intent(ManageActivity.this, PassengerActivity.class);
    intent.putExtra("USERNAME", loggedUser);
    intent.putExtra("MESSAGE", result);
     startActivity(intent);

}
```

PassengerActivity

Το UI της κλάσης ορίζεται από το activitypassenger.xml και περιέχει ένα dropdown menu με τους πιθανούς επιβάτες καθώς και τα buttons «επιλογή επιβάτη» και «αρχική οθόνη».



Η κλάση λαμβάνει σαν message μεταβλητή τα αποτελέσματα με τους πιθανούς επιβάτες έχοντας το id της αίτησης και τα λεπτομερή στοιχεία τα οποία και εμφανίζονται στο spinner (dropdown menu) με τον παρακάτω κώδικα:

```
//creates drop down menu for requests for the route send from manage activity
spinner = (Spinner)findViewById(R.id.spinner);
try {
   stringArray = new ArrayList<String>();
   JSONArray arr = new JSONArray(message);
   for(int i = 0; i < arr.length(); i++){
      stringArray.add(arr.getJSONObject(i).getString("requestid")+"
"+arr.getJSONObject(i).getString("details"));
   }
} catch (Exception e) {
   e.printStackTrace();
}
```

Κατόπιν υπάρχουν δύο listeners για το κάθε button. Με την επιλογή «Επιλογή Επιβάτη» καλείται η κλάση AsyncDataClass asyncRequestObject

```
admitButton.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      selectedtext = spinner.getSelectedItem().toString();

      AsyncDataClass asyncRequestObject = new AsyncDataClass();
      asyncRequestObject.execute(serverUrl, selectedtext);


   }
});
```

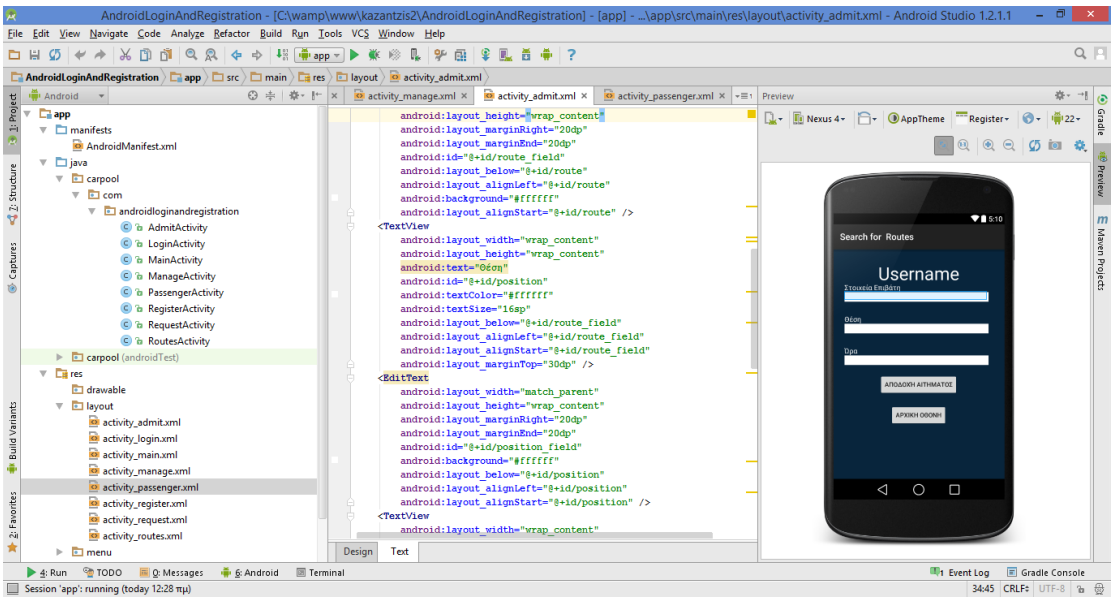
Η κλάση αποστέλλει ένα http request στη σελίδα **androidadmit.php**
Και λαμβάνει σαν response τα στοιχεια για το συγκεκριμένο requestid τα οποία και τα στέλνει με τη μορφή μεταβλητής message στην κλάση AdmitActivity

```
protected String doInBackground(String... params) {

   HttpParams httpParameters = new BasicHttpParams();
   HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
   HttpConnectionParams.setSoTimeout(httpParameters, 5000);

   HttpClient httpClient = new DefaultHttpClient(httpParameters);
   HttpPost httpPost = new HttpPost(params[0]);

   String jsonResult = "";
   try {
      List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
      nameValuePairs.add(new BasicNameValuePair("requestid", params[1]));
      httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

      HttpResponse response = httpClient.execute(httpPost);
      jsonResult = inputStreamToString(response.getEntity().getContent()).toString();

   } catch (ClientProtocolException e) {
      e.printStackTrace();
   } catch (IOException e) {
```

```
            e.printStackTrace();
        }
        return jsonResult;
    }
```

AdmitActivity

Το UI της κλάσης περιγράφεται από το activity_admit.xml και περιέχει 3 edit text και 2 buttons , ένα για επιστροφή στην «Αρχική Οθόνη» και ένα για την «Αποδοχή Αιτήματος»



Στα πεδία εμφανίζονται τα στοιχεία του πιθανού επιβάτη, η χιλιομετρική θέση και η ώρα. Τα δεδομένα αυτά παραλαμβάνονται από τη Passengeractivity κλάση με τη μορφή μεταβλητής (message). Τα πεδία θέση και ώρα μπορούν να μεταβληθούν από τον χρήστη. Κατά την αποδοχή του αιτήματος καλείται η κλάση η οποία αποστέλλει ένα http request προς τη σελίδα **androidupdate**.php και στη συνέχεια εκεί εκτελείται ένα update query το οποίο κάνει το αίτημα ενεργό και αποστέλει ένα email στον επιβάτη για ενημέρωση.

```
private class AsyncDataClass extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... params) {

        HttpParams httpParameters = new BasicHttpParams();
        HttpConnectionParams.setConnectionTimeout(httpParameters, 5000);
```

```
    HttpConnectionParams.setSoTimeout(httpParameters, 5000);

    HttpClient httpClient = new DefaultHttpClient(httpParameters);
    HttpPost httpPost = new HttpPost(params[0]);

    String Result = "";
    try {
       List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);
       nameValuePairs.add(new BasicNameValuePair("requestid", params[1]));
       nameValuePairs.add(new BasicNameValuePair("position", params[2]));
       nameValuePairs.add(new BasicNameValuePair("traveltime", params[3]));
       httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

       HttpResponse response = httpClient.execute(httpPost);
       Result = inputStreamToString(response.getEntity().getContent()).toString();
       System.out.println("Returned  " + Result.toString());

    } catch (ClientProtocolException e) {
       e.printStackTrace();
    } catch (IOException e) {
       e.printStackTrace();
    }
    return Result;
 }
```

Αποτίμηση

Στην παρούσα πτυχιακή εργασία, αναλύθηκε η έννοια του *σενεπιβατισμού* σαν κοινωνικό φαινόμενο αλλά κυρίως το πώς μπορεί αυτή έμπρακτα να εφαρμοστεί. Η αποτύπωση αυτής της ιδέας ήταν φυσικά η δημιουργία της αντίστοιχης εφαρμογής που προσφέρεται στο ενδιαφερόμενο κοινό (και η αντίστοιχη ιστοσελίδα) και του επιτρέπει πλήθος επιλογών. Παρουσιάστηκε λεπτομερώς η φιλοσοφία πίσω από το εγχείρημα, η διαδικασία, ο τρόπος λειτουργίας και η εφαρμογή του λογισμικού. Ο χρήστης εξυπηρετείται σε φιλικό περιβάλλον πλοήγησης, ενώ πρακτικά η εφαρμογή εξοικονομεί χρόνο, κόπο και χρήματα.

Πέραν όμως του υλικοτεχνικού επιτεύγματος, η εφαρμογή αποτέλεσε χαρακτηριστικό δείγμα της τεχνολογίας στην υπηρεσία του ανθρώπου, ενώ παράλληλα

κατέστησε σαφές το πώς μέσα σαν τα κινητά τηλέφωνα και το διαδίκτυο συμβάλουν πλέον καταλυτικά στο σύγχρονο τρόπο ζωής. Δεν είναι τυχαία η απήχηση που γνωρίζουν, καθώς είναι χαρακτηριστική η ευκολία στην πρόσβασή τους. Τα κινητά τηλέφωνα αλλά και το ίντερνετ αποτελούν πια μια καθημερινότητα, ο αριθμός των κατόχων/συνδρομητών διαρκώς πολλαπλασιάζεται και δε θα ήταν υπερβολική η εντύπωση πως κρίνονται απαραίτητα για την ποικιλία των δραστηριοτήτων που παρέχουν.

Αντί επιλόγου, παρατίθεται ένα χρήσιμο συμπέρασμα. Η διαρκής επινόηση νέων τεχνολογικών μέσων και τα βήματα προόδου της επιστήμης καθημερινά, αποτελούν το καλύτερο υπόβαθρο για τη βελτίωση του βιοτικού επιπέδου αλλά και την εξυπηρέτηση των ανθρώπινων αναγκών. Κάθε έμπνευση ή καινοτόμος ιδέα όταν συνοδεύεται από την επιστημονική επάρκεια αλλά και ηθική που οφείλει να διέπει το δημιουργό, μπορεί να αποτελέσει χρήσιμο εργαλείο και να συμβάλει στην περεταίρω εξέλιξη της ανθρώπινης φύσης. Το τμήμα στα πλαίσια του οποίου υλοποιήθηκαν η ανωτέρω εφαρμογή και η συνοδευτική εργασία, αποτελεί χαρακτηριστικό παράδειγμα αυτού του συμπεράσματος, καθώς εφοδιάζει τους σπουδαστές με γνώση, ενώ παράλληλα επιτρέπει την πρωτοβουλία και τη δημιουργία Και αν ερωτήματα και προβληματισμοί γεννώνται σχετικά με τη ραγδαία βελτίωση της τεχνολογίας, είναι χρέος του επιστήμονα να διαλύει της αμφιβολίες και να δημιουργεί, με γνώμονα το ωφέλιμο.

Εφαρμογές σαν και αυτή, η οποία σχετίζεται με το carpooling, είναι ένα μόνο μέρος της ανθρώπινης δημιουργίας. Ανάλογες ιδέες πρέπει να ενισχύονται και να υλοποιούνται, βοηθώντας την καθημερινή ζωή και εμπειρία και τελικά την ανάπτυξη και την εξυπηρέτηση.