

Design & Development of Web & Mobile Apps

Final Project Report |



***PATH** Mobile App*

Group 4

Charlie DeCaro | Parth Hiren Rao | Rafael Garcia Cano Da Costa | Ruxi Pan | Nitesh Saurav

Executive Summary & Background

As part of an innovation team at the Port Authority of New York & New Jersey, Charlie DeCaro's role is to uncover innovative ways to generate new non-toll, non-fare revenues for the agency, as well as improve efficiency and customer service through better technology. The PATH, a NY-NJ commuter train service owned by the agency, currently runs at approximately a \$300 million / year loss. It is also subject to delays due to the age of the system and damage from Hurricane Sandy. While these issues will take several years to fix, Charlie thought there might be a quicker and less costly way to increase quality of commute in the meantime, while generating a new source of revenue. This is how the idea of a PATH app was born.

A simple app for scheduling, delay alerts, and system maps, would help commuters check train times and easily stay up to date on service issues, while generating revenue through advertising. After some discussion with agency stakeholders, they agreed it would be a good idea for our team at Stern to take a stab at building a prototype as our class project, and if successful, they would allow us to present for senior leadership.

In the end, we were in fact successful in building application, with most of the features initially proposed. We built the front end / client side using Dreamweaver, JQuery and JQuery mobile, HTML, CSS, and JavaScript. We built the server side using NodeJS, which is a JavaScript runtime environment. We then stored the data in MongoDB for persistence, and made the data available through ExpressJS, which is a web application framework.

Description of Problem

Currently, if PATH riders want to check train schedules, they have to use the Port Authority website, which is old and difficult to navigate. It takes at least 4-5 clicks just to get the timetables page. On a phone, it is even worse, since there is no mobile site – only a desktop

site – so the navigation is clumsy and the small buttons are hard to press. The same issue exists if a user wants to find a map of the system.

If a user wants to receive real time alerts, they have to subscribe to a Twitter feed, which means in order to find the alerts, they have to open up Twitter, and dig through a bunch of other non-PATH tweets until they come across an alert. The alternative is to sign up for text or e-mail alerts, which are notoriously unreliable and often come late or not at all.

Solution

Our solution is an official PATH mobile app, which allows users to quickly and easily access timetables for any PATH route, based on the current day and time. Being that the app is connected to the PATH's official alerts RSS feed, users can receive alerts real time without having to dig around. Finally, riders can access the PATH system map, which is useful if they are new to the area and unfamiliar with the stations along each route.

Value Proposition

The Port Authority of New York & New Jersey currently does not have an official app for the PATH train. There are other third party apps for schedules available, but none of them include real time alerts, and do not generate any revenue for the agency. The app we have built will be an official PATH app, that incorporates both scheduling and alerts, in addition to maps, while serving as a mid step toward a larger, future PATH app that includes mobile ticket purchasing and mobile station-entry capabilities. Real-time alerts provide particular value over the user's current experience of searching Twitter or signing up for text alerts, given the alerts are more easily accessible through the app, and more current. If riders want to check the PA website for alerts or schedules, they have to click through 3-4 different links to get to the

information they need, which is difficult to do on a mobile device, as the PA only has a desktop site. This app makes access to this information immediate, frictionless, and condenses it into one place to reduce unnecessary navigation. It also benefits the agency, as they too can consolidate alerts communications into one place, and eventually incorporate mobile payments technology into this platform.

Competition & Competitive Advantage

Currently, there are several third party apps on the market for PATH scheduling. Each of them are paid apps, and do not have real-time alert capability.

Our PATH app will be an official one for the agency, and since Charlie is an employee, this allows unfettered access to real time alerts data, which the competition cannot access. The app will also be free to download, vs. the other apps' \$1-\$4 dollar price tags.

From a branding perspective, since this app will bear the official PATH logo and name, we will have the benefit of riders instinctively trusting it over that of a third party.

Finally, this app will lay the groundwork for the agency to incorporate real-time scheduling changes in 2017, when the positive train control system is installed. It also gives them a platform on which to build in mobile ticket purchasing and contactless fare payments, which are on the horizon for 2019-20.

Target Market

Our target market consists of both daily commuters and occasional users of the PATH. Commuters in particular are best-served by a mobile app, since they are most likely to be time sensitive in their travel needs. Any delays could mean being late to work and having to plan alternate transportation. Reliable, easy-to-access real-time alerts would benefit them most for

this reason. Access to scheduling would also help them in the sense that they will always know approximately when the next train is coming.

Monetization

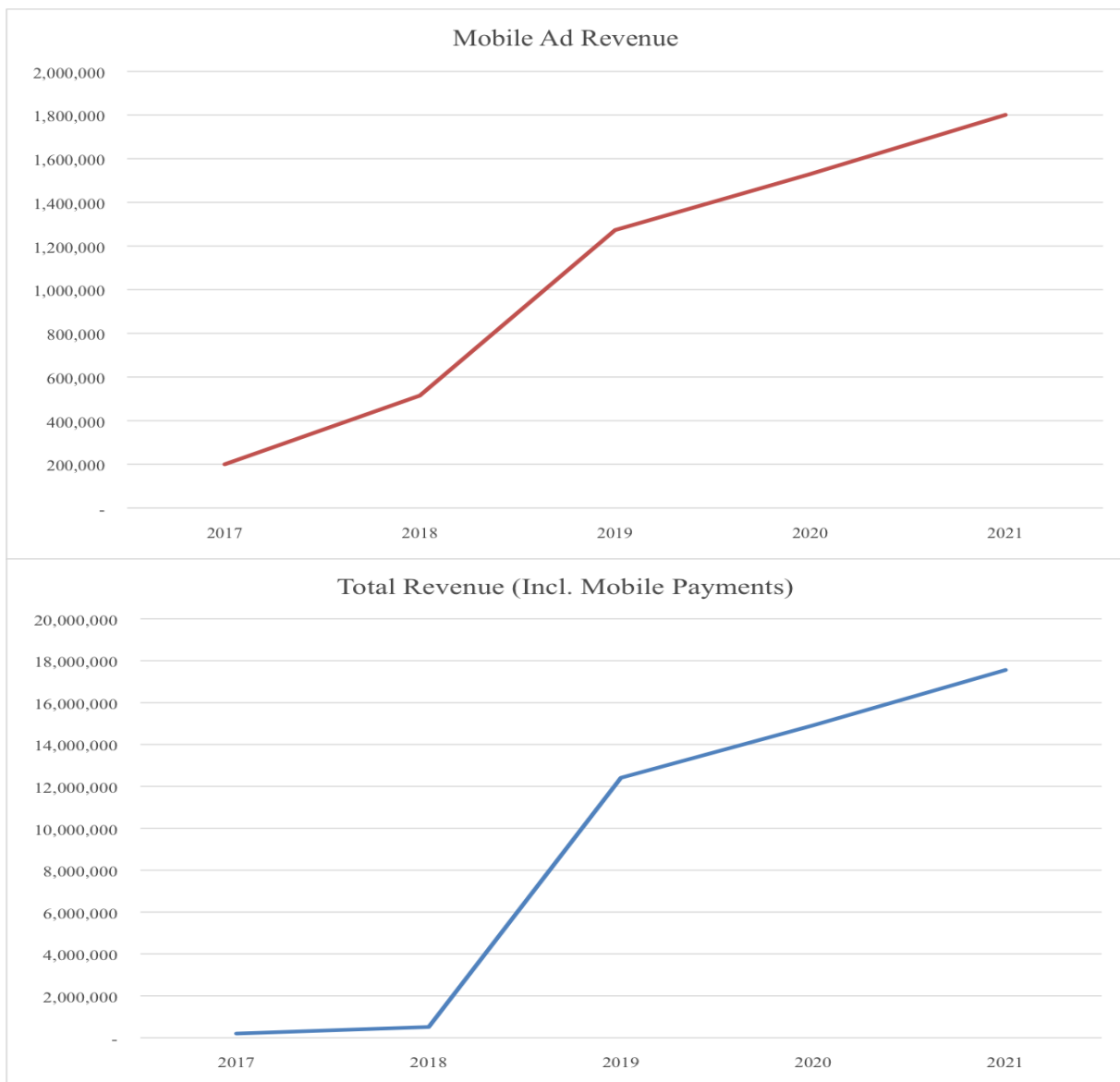
Monetization will be achieved through advertising. xAd is a firm that specializes in integrating location based advertising capabilities into mobile applications. Our intent would be to integrate xAd's ad network and technology into the app, in order to generate revenue through location based advertising campaigns, which xAd's CEO has publically indicated generate \$30 CPM, on average. We used a more conservative \$20 CPM in our projections below.

Financial Projection

	2017	2018	2019	2020	2021
Annual Ride Volume	100,000,000	103,000,000	106,090,000	109,272,700	112,550,881
Ride Volume Growth	3%	3%	3%	3%	3%
Adoption Rate	10%	25%	30%	35%	40%
Users	10,000,000	25,750,000	31,827,000	38,245,445	45,020,352
Impressions Per Ride	1	1	2	2	2
Annual Impressions	10,000,000	25,750,000	63,654,000	76,490,890	90,040,705
in Thousands (for CPM calc.)	10,000	25,750	63,654	76,491	90,041
CPM (xAd)*	20	20	20	20	20
Total Ad Revenue	200,000	515,000	1,273,080	1,529,818	1,800,814
Increased Yield From Mobile Payments**			11,139,450	13,385,906	15,757,123
(\$0.16 min. increase per sale based on prior study)					
Total Revenue incl. Mobile Payments	200,000	515,000	12,412,530	14,915,724	17,557,937
* Assuming Port Authority can strike a deal with xAd, whose CEO has claimed that Xad's location based mobile advertising network generates up to \$30 CPM on average; conservative estimate of \$20 used in projection					
** PA will introduce contactless mobile payments in 2019, which will layer into this application					

Marketing / Growth Strategy

Fortunately, the Port Authority has an advertising relationship with JC Decaux, which allows them to advertise throughout PATH stations, platforms, and train interiors for little to no cost. This will be the primary method by which riders will be informed of the PATH app. The PA will also make use of loudspeaker announcements in stations. Finally, the PA has a strong social media presence on Facebook, Instagram, and Twitter, which it will use for PSA's about the app. The remainder of adoption growth among riders will be achieved through word of mouth.



Problems Encountered

One problem we encountered was the initial formatting of the timetable data. We had to do quite a bit of data cleaning and reformatting in order to get the data into useable tables for querying. Another issue we ran into was getting the station-to-station functionality to work. We realized that if the user selects a “to” station and a “from” station that are on different routes, we would need to build a sort of navigation algorithm that alerts users that they will have to transfer at a particular station at a certain time. Given the time constraints of the project, we decided to amend this function to only allow users to pick stations on the same route. In the future, this functionality could be more fully built out.

Another issue was that in trying to include some of the logic on the client side, we realized there would be some difficulty in integrating with PhoneGap. For this reason, we had to move the logic to the server side. For example, when fetching alert data from the PATH RSS feed and parsing the XML using Node modules, this introduced complications when running Node with PhoneGap. We had to move the fetching and parsing algorithms to the server side.

Lessons Learned

Given that each member had a different background or set of skills, we ended up learning a lot from each other. For instance, those of use who had strong front-end skills were able to learn from those with strong back-end and server skills by observing their work and code. Another lesson learned is that it is often difficult to communicate ideas between those with different backgrounds. Sometimes you have to learn to say something in a different or less technical way for those with less or different experience to grasp a concept.

Next Steps

Next steps for the app would be to launch among a limited group of people and test to work out any kinks. Once this version of the app has been tested, then we would aim to get it onto the Apple App Store, Google Play Store, and any other mobile stores for general public download. Next, we would present to PATH executive leadership to demonstrate the functionality and revenue potential. Finally, once the PA launches mobile payment capabilities for riders in 2019, this could be incorporated into the existing app, allowing users to purchase tickets and gain access to stations via swiping the phone through the turnstiles.

Conclusion

What we can conclude after completing this project is that what may seem like a simple application can prove much more complex when trying to implement it. For instance, designing and building the front end was fairly easy at first. However, once we started building the back end, we realized that we needed to rearrange the front end, i.e. move some logic from the client side to the server side, reorder code, and rename quite a few variables.

Also, we learned that each component of the project – front end design, server side and database, client side, data cleaning, XML parsing, etc. - all occupy very different universes of thinking, and are relatively simple on their own. The difficulty lies in linking all of these components together, and communicating to each other how each of our components work, so that they can be properly linked.

Additionally, we have may have “bitten off more than we could chew” in terms of all the features we expected we could deliver in the time frame allotted. In the future, given a similar task, we will be better prepared to establish reasonable feature lists and roadmaps.

APPENDIX

A. SAMPLE CODE SCREEN SHOTS (for full code visit: <https://github.com/rgc292/mobileApp>)

Client Side

Index.js:

```
var d = new
Date();

var weekday = new Array(7);
weekday[0]= "Sunday";
weekday[1] = "Monday";
weekday[2] = "Tuesday";
weekday[3] = "Wednesday";
weekday[4] = "Thursday";
weekday[5] = "Friday";
weekday[6] = "Saturday";


var n = weekday[d.getDay()];
// var n = "Sunday";
var globalData = null;
var hashTable = {};
var SaturdayHashTable = {};
var SundayHashTable = {};
var MondayHashTable = {};
var TuesdayHashTable = {};
var ThursdayHashTable = {};
var WednesdayHashTable = {};
var FridayHashTable = {};


var timeTable = {
  "Hoboken to 33rd":["HOBOKEN","CHRISTOPHER ST","9TH ST","14TH ST","23RD
ST","33RD ST"],
  "33rd to Hoboken":["33RD ST","23RD ST","14TH ST","9TH ST","CHRISTOPHER
ST","HOBOKEN"],
  "Hoboken to WTC":["HOBOKEN","NEWPORT","EXCHANGE PLACE","WORLD TRADE
CENTER"],
  "WTC to Hoboken":["WORLD TRADE CENTER","EXCHANGE
PLACE","NEWPORT","HOBOKEN"],
  "Journal Square to 33rd":["JOURNAL SQUARE","GROVE
ST","NEWPORT","CHRISTOPHER ST","9TH ST","14TH ST","23RD ST","33RD ST"],
```

```

    "33rd to Journal Square":["33RD ST","23RD ST","14TH ST","9TH
ST","CHRISTOPHER ST","NEWPORT","GROVE ST","JOURNAL SQUARE"],
    // "Journal Square to 33rd via Hoboken":["JOURNAL SQUARE","GROVE
ST","NEWPORT","HOBOKEN","CHRISTOPHER ST","9TH ST","14TH ST","23RD
ST","33RD ST"],
    // "33rd to Journal Square via Hoboken":["33RD ST","23RD ST","14TH
ST","9TH ST","CHRISTOPHER ST","HOBOKEN","NEWPORT","GROVE ST","JOURNAL
SQUARE"],
    "Journal Square to Hoboken":["JOURNAL SQUARE","GROVE
ST","NEWPORT","HOBOKEN"],
    "Hoboken to Journal Square":["HOBOKEN","NEWPORT","GROVE ST","JOURNAL
SQUARE"],
    "Newark to WTC":["NEWARK","HARRISON","JOURNAL SQUARE","GROVE
ST","EXCHANGE PLACE","WORLD TRADE CENTER"],
    "WTC to Newark":["WORLD TRADE CENTER","EXCHANGE PLACE","GROVE
ST","JOURNAL SQUARE","HARRISON","NEWARK"]
};

```

```

for(key in timeTable){
    hashTable[key] = {};
    // MondayHashTable[key] = {};
    // TuesdayHashTable[key] = {};
    // WednesdayHashTable[key] = {};
    // ThursdayHashTable[key] = {};
    // FridayHashTable[key] = {};
    // SaturdayHashTable[key] = {};
    // SundayHashTable[key] = {};
    for(var i = 0; i < timeTable[key].length; i++) {
        hashTable[key][timeTable[key][i]] = {};
        // SundayHashTable[key][timeTable[key][i]] = {};
        // MondayHashTable[key][timeTable[key][i]] = {};
        // TuesdayHashTable[key][timeTable[key][i]] = {};
        // WednesdayHashTable[key][timeTable[key][i]] = {};
        // ThursdayHashTable[key][timeTable[key][i]] = {};
        // FridayHashTable[key][timeTable[key][i]] = {};
        // SaturdayHashTable[key][timeTable[key][i]] = {};
    }
}

```

```

var timeTableData = {
    // "Hoboken to 33rd": []
}

$(document).ready(function(){
    $("a.selection").click(function(){
        $("h3#timetable_header").text(this.text+" - "+n);

        var array_name = timeTable[this.id];
        var path = this.id;
        // console.log('path ==', path);
        $('#timetable_display tr').not(':first').remove();
        if(array_name == null) {
            $("#error").html("<p>No train is scheduled for today for this
path.</p>");
            return;
        }
        var arrayLength = array_name.length;
        // $("#timetable_display tr").remove();

        $("#column_headers").html('');
        for (i = 0; i < arrayLength; i++) {
            $("#column_headers").append("<th>" + array_name[i] + "</th>");
        }

        var pathTable = timeTableData[path];
        console.log('path is ==', path);
        console.log('path table ==', pathTable);
        var html = '';
        $("#timetable_error").html("");
        if(pathTable == null || pathTable.length == 0 ) {
            $("#timetable_error").html("<p>No train is scheduled for today
for this path.</p>");
        }
        else {
            for(var i = 1; i < pathTable.length; i++) {
                var row = '';

```

```

        for(var j = 0 ; j < array_name.length ; j++) {
            row = row + '<td>' + pathTable[i][array_name[j]] + '</td>';
        }

        html += '<tr>' + row + '</tr>';
    }
    $('#timetable_display tr').first().after(html);
    html = '';
}

});

$.ajax({
    type: 'GET',

    // The URL to make the request to.
    url: 'http://websys3.stern.nyu.edu:7004/api/timesheets',
    contentType: 'application/json',
    xhrFields: {
        // The 'xhrFields' property sets additional fields on the
XMLHttpRequest.
        // This can be used to set the 'withCredentials' property.
        // Set the value to 'true' if you'd like to pass cookies to the
server.
        // If this is enabled, your server must respond with the header
        // 'Access-Control-Allow-Credentials: true'.
        withCredentials: false
    },

    headers: {
        // Set any custom headers here.
        // If you set any non-simple headers, your server must include
these
        // headers in the 'Access-Control-Allow-Headers' response header.
    },

    success: function(data) {

```

```

        // Here's where you handle a successful response.
        // console.log('data is ==', data);
        globalData = data;
        organiseData(data);
    },

    error: function(error) {
        globalData = error;
        console.log('error is ==', error);
    }
});

$.get("http://websys3.stern.nyu.edu:7004/api/timesheets",
function(data){
    // console.log('data is ==', data);
    // alert("Data: " + data);
    // });

$("#alert_icon").click(function(){
$.ajax({
    type: 'GET',

    // The URL to make the request to.
    url: 'http://websys3.stern.nyu.edu:7004/api/alerts',
    contentType: 'application/json',
    xhrFields: {
        withCredentials: false
    },

    headers: {

    },

    success: function(data) {
        // console.log('Alerts is ==', data);
        globalAlerts = data.alert;
        showAlerts(data.alert);
    },

```

```

        error: function(error) {
            globalAlerts = error;
            console.log('error is ==', error);
        }
    });
});

```

```

// Station To Station Click logic
$("#search_button").click(function() {
    // console.log('coming in search_button logic');
    var from = $("#from")[0].options[$("#from")[0].selectedIndex].value;
    var to = $("#to")[0].options[$("#to")[0].selectedIndex].value;
    console.log('From is ==', from);
    console.log('to is ==', to);

    var possiblePath = findPossiblePath(from, to);
});
});

```

```

var organiseData = function(data) {
    for (var i = 0 ; i < data.length ; i++){
        if(data[i].day == 'Weekday') {
            hashTable[data[i].route][data[i].station] = data[i];
        }
        else if(data[i].day == 'Monday') {
            if(MondayHashTable[data[i].route] == null) {
                MondayHashTable[data[i].route] = {};
            }
            MondayHashTable[data[i].route][data[i].station] = {};
            MondayHashTable[data[i].route][data[i].station] = data[i];
        }
        else if(data[i].day == 'Tuesday') {
            if(TuesdayHashTable[data[i].route] == null) {
                TuesdayHashTable[data[i].route] = {};
            }
        }
    }
}

```

```

    }
    TuesdayHashTable[data[i].route][data[i].station] = {};
    TuesdayHashTable[data[i].route][data[i].station] = data[i];
}
else if(data[i].day == 'Wednesday') {
    if(WednesdayHashTable[data[i].route] == null) {
        WednesdayHashTable[data[i].route] = {};
    }
    WednesdayHashTable[data[i].route][data[i].station] = {};
    WednesdayHashTable[data[i].route][data[i].station] = data[i];
}
else if(data[i].day == 'Thursday') {
    if(ThursdayHashTable[data[i].route] == null) {
        ThursdayHashTable[data[i].route] = {};
    }
    ThursdayHashTable[data[i].route][data[i].station] = {};
    ThursdayHashTable[data[i].route][data[i].station] = data[i];
}
else if(data[i].day == 'Friday') {
    if(FridayHashTable[data[i].route] == null) {
        FridayHashTable[data[i].route] = {};
    }
    FridayHashTable[data[i].route][data[i].station] = {};
    FridayHashTable[data[i].route][data[i].station] = data[i];
}
else if(data[i].day == 'Saturday') {
    if(SaturdayHashTable[data[i].route] == null) {
        SaturdayHashTable[data[i].route] = {};
    }
    SaturdayHashTable[data[i].route][data[i].station] = {};
    SaturdayHashTable[data[i].route][data[i].station] = data[i];
}
else if(data[i].day == 'Sunday') {
    if(SundayHashTable[data[i].route] == null) {
        SundayHashTable[data[i].route] = {};
    }
    SundayHashTable[data[i].route][data[i].station] = {};
    SundayHashTable[data[i].route][data[i].station] = data[i];
}
}

var todayHashTable = {};

```

```

console.log('value of n ==', n);
switch (n) {
  case "Sunday":
    todayHashTable = SundayHashTable;
    break;
  case "Monday":
    todayHashTable = MondayHashTable;
    break;
  case "Tuesday":
    todayHashTable = TuesdayHashTable;
    break;
  case "Wednesday":
    todayHashTable = WednesdayHashTable;
    break;
  case "Thursday":
    todayHashTable = ThursdayHashTable;
    break;
  case "Friday":
    todayHashTable = FridayHashTable;
    break;
  case "Saturday":
    todayHashTable = SaturdayHashTable;
    break;
  default:
    todayHashTable = hashTable;
}

if(n != 'Saturday' && n != 'Sunday') {
  console.log('hashTable in if ==', hashTable);
  for(path in todayHashTable) {
    timeTableData[path] = [];
    for(station in todayHashTable[path]) {
      for(key in todayHashTable[path][station]) {
        if(key != "_id" && key != "day" && key != "route" && key !=
"station" && parseInt(key) != 153 && todayHashTable[path][station][key]
!= "") {
          var tempObj = {};
          tempObj[station] = todayHashTable[path][station][key]
          timeTableData[path][parseInt(key)] = $.extend({},
timeTableData[path][parseInt(key)], tempObj);
        }
      }
    }
  }
}

```



```

    }
}

for(path in hashTable) {
    var j = 0;
    if(timeTableData[path] == null || timeTableData[path].length == 0){
        timeTableData[path] = [];
    }
    else {
        j = timeTableData[path].length - 1;
    }

    for(station in hashTable[path]) {
        for(key in hashTable[path][station]) {
            if(key != "_id" && key != "day" && key != "route" && key !=
"station" && parseInt(key) != 153 && hashTable[path][station][key] !=
"") {
                var tempObj = {};
                tempObj[station] = hashTable[path][station][key]
                timeTableData[path][parseInt(key) + j] = $.extend({},
timeTableData[path][parseInt(key) + j], tempObj);
            }
        }
    }
}

// console.log("hashTable before==", hashTable);
// console.log('todays hashtable ==', todayHashTable);
// console.log('todays timetable before ==', timeTableData);

else {
    for(path in todayHashTable) {
        timeTableData[path] = [];
        for(station in todayHashTable[path]) {

```

```

        for(key in todayHashTable[path][station]) {
            if(key != "_id" && key != "day" && key != "route" && key !=
"station" && parseInt(key) != 153 && todayHashTable[path][station][key]
!= "") {
                var tempObj = {};
                tempObj[station] = todayHashTable[path][station][key]
                timeTableData[path][parseInt(key)] = $.extend({},
timeTableData[path][parseInt(key)], tempObj);
            }
        }
    }
}

// console.log("hashTable after==", hashTable);
console.log("timeTableData afer==", timeTableData);
}

```

```

var showAlerts = function (alerts) {
    // $('#alert_content li').not(':first').remove();
    // var pathTable = timeTableData[path];
    var html = '';
    if(alerts.length == 0) {
        //html = 'No recent service alerts at this time.';
        $('#alert_error').html("<p>No recent service alerts at this
time.</p>");
    }
    else {
        $('#ul#alert_content > li').remove() ;
        // $('#ul#alert_content').append('<li><a id="Hoboken to 33rd"
class="selection" href="#TimetableContainer">Hoboken to 33rd</a></li>');
        for(var i = 0; i < alerts.length; i++) {
            html += '<li class="alert_items"> <div>' + alerts[i].description +
'</div>' + '<div class="date">' + alerts[i].date + '</div></li>';
        }
        // html = 'something';
        // console.log('coming here and html is==', html);
        $('#ul#alert_content').append(html);
        html = '';
    }
}

```

```
}
```

```
var findPossiblePath = function(from, to) {
```

```
    var possibleFromPaths = {};
```

```
    var possibleToPaths = {};
```

```
    var possiblePaths = {};
```

```
    for(key in timeTable) {
```

```
        for(var i=0; i < timeTable[key].length; i++) {
```

```
            if(timeTable[key][i] == from) {
```

```
                possibleFromPaths[key] = i;
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
    for(key in possibleFromPaths) {
```

```
        for(var i=0; i < timeTable[key].length; i++) {
```

```
            if(timeTable[key][i] == to && i > possibleFromPaths[key]) {
```

```
                possiblePaths[key] = {from:possibleFromPaths[key], to:i};
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
    //$("ul#search_result > li").remove();
```

```
    //$("ul#search_result").html("<p><strong>Route Results:
```

```
</strong></p><br>");
```

```
    //var i =10;
```

```
    //for(key in possiblePaths) {
```

```
        //var html = '<li><a id=' + i + ' class="search_selection"
```

```
href="#SearchTableContainer">' + key + '</a></li><br>';
```

```
        //$("ul#search_result").append(html);
```

```
        //i++;
```

```
    //}
```

```
    $("#result_container").html("<p><strong>Route Results:
```

```
</strong></p><br>");
```

```
    $("#result_container").append('<ul data-role="listview"
id="search_result" style="list-style: none"></ul>');
    var i =10;
```

```

        for(key in possiblePaths) {
            var html = '<div class="route_result"
style="width:75%"><br><li><a id=' + i + ' class="search_selection"
href="#SearchTableContainer">' + key + '</a></li><br></div><br>';
            $("#ul#search_result").append(html);
            i++;
        }

$("a.search_selection").click(function(){
    $("h3#searchtable_header").text(this.text+" - "+n);

    var array_name = timeTable[this.text];
    var path = this.text;
    // console.log('path ==', possiblePaths[path].from);
    $('#searchtable_display tr').not(':first').remove();
    if(array_name == null || array_name.length == 0) {
        //html = 'No train is scheduled for today for this path.'
        $("#error").html('<p>No direct routes available at this
time.</p>');
        return;
    }
    var arrayLength = array_name.length;
    // $("#timetable_display tr").remove();

    $("#search_column_headers").html('');
    for (i = possiblePaths[path].from; i <= possiblePaths[path].to;
i++) {
        $("#search_column_headers").append("<th>" + array_name[i] +
"</th>");
    }

    var pathTable = timeTableData[path];
    console.log('path is ==', path);
    console.log('path table ==', pathTable);
    var html = '';
    if(pathTable == null || pathTable.length == 0) {
        $("#searchtable_error").html("<p>No direct routes available at
this time.</p>");
    }

```

```

else {
    var currentTime = new Date();
    var currentHours = currentTime.getHours();
    var currentMinutes = currentTime.getMinutes();

    var i = 1;
    for(; i < pathTable.length; i++) {

if(compareTimeGreater(pathTable[i][array_name[possiblePaths[path].from]],
{hours:currentHours, minutes:currentMinutes})) {
    break;
    }
    }
    console.log('is final value == ', i);
    console.log('pathtables final value == ', pathTable.length);
    $("#searchtable_error").html("");
    if(i == pathTable.length) {
        $("#searchtable_error").html("<p>No direct routes available at
this time.</p>");
    }
    else {
        for(;i < pathTable.length; i++) {
            var row = '';
            for(var j = possiblePaths[path].from ; j <=
possiblePaths[path].to ; j++) {
                row = row + '<td>' + pathTable[i][array_name[j]] + '</td>';
            }

            html += '<tr>' + row + '</tr>';
        }
        $('#searchtable_display tr').first().after(html);
        html = '';
    }

}

});

```

```

    console.log('possiblePaths ==', possiblePaths);
    return possiblePaths;
}

```

```

compareTimeGreater = function(str, currentTime) {

```

```

    str.trim();
    var ampm = str.substr(str.length-2, 2);
    ampm.trim();
    // console.log('ampm ==', ampm);
    var time = str.substr(0,5);
    time.trim();
    var hours = parseInt(time.split(':')[0]) + (ampm == 'PM' ? 12 : 0);
    var minutes = parseInt(time.split(':')[1]);
    if(hours == 12){
        hours = 0;
    }
    else if(hours == 24){
        hours = 12;
    }
}

```

```

    console.log('comparing ==' + str + ' :with == ', hours + ' and ' +
currentTime.hours );

```

```

    var answer = false;
    if(hours > currentTime.hours) {
        answer = true;
    }
    else if(hours == currentTime.hours){
        console.log('with hours true ==' + str + ' :with hours== ', hours + '
and ' + currentTime.hours + ' :with minutes ==' + minutes + ' and ' +
currentTime.minutes);
        if(minutes >= currentTime.minutes){
            answer = true;
            console.log('with minutes true ==' + str + ' :with == ', hours + '
and ' + currentTime.hours);
        }
    }
}

```

```

    }
    return answer;
    // console.log(str + 'of hours are ==' + hour);
}

```

Index.html:

```

<!doctype
e html>

<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PATH Mobile App</title>
    <style>
        .pathlogo{
            text-align:center;
        }
        #menu{
            background-image: url(Images/background-path.png);
            background-repeat:no-repeat;
            background-position:center;
        }
        .icon{
            position:relative;
            text-align: center;
            top: 1em;
            margin-bottom: .5em;
        }
        #column_headers{
            font-size: .8em;
        }
        #timetable_header{
            width:auto;
            overflow:visible;
        }
        #system_map{
            width:"100%";
        }
        #timetable_scroll{
            overflow:scroll;
        }
        #searchtable_scroll{

```

```

        overflow:scroll;
    }
</style>
<link href="jquery.mobile.prior/jquery-mobile/jquery.mobile.theme-
1.3.0.min.css" rel="stylesheet" type="text/css">
<link href="jquery.mobile.prior/jquery-
mobile/jquery.mobile.structure-1.3.0.min.css" rel="stylesheet"
type="text/css">
<link href="./css/index.css" rel="stylesheet" type="text/css">
<script src="jquery.mobile.prior/jquery-mobile/jquery-
1.11.1.min.js"></script>
<script src="jquery.mobile.prior/jquery-mobile/jquery.mobile-
1.3.0.min.js"></script>

<!-- <script src="http://code.jquery.com/jquery-
1.11.1.min.js"></script> -->
<script src="http://code.jquery.com/jquery-1.12.3.js"></script>
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></scr
ipt>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.7/angular.min.js"><
/script>
<script
src="https://cdn.datatables.net/1.10.12/js/jquery.dataTables.min.js"></scrip
t>
<script src="js/index.js"> </script>
</head>

<body>
<div data-role="page" id="page1">
    <div data-role="header" data-theme="f">
        <div class="pathlogo">
            
        </div>
    </div>
    <div id="menu" data-role="content">
        <ul data-role="listview" data-inset="true">
            <div class="icon">

```



```

        <li><a id="schedules" href="#Schedules"></a></li>
    </div>
    <div class="icon">
        <li><a href="#Alerts" id="alert_icon"></a></li>
    </div>
    <div class="icon">
        <li><a href="#Map"></a></li>
    </div>

</ul>
</div>
<div align="center" data-role="footer" data-theme="f">
    <h1></h1>
</div>
</div>

```

```

<div data-role="page" id="Schedules">
    <div data-role="header" data-theme="f">
        <div class="pathlogo" data-role="content"></div>
        <a href="#page1" data-role="button" data-theme="b">Back</a>
    </div>
    <div align="center" data-role="header" data-theme="b">
        <h3>Schedules</h3>
    </div>
    <div id="datesummary">
        <p></p>
    </div>
    <ul data-role="listview">
        <li><a href="#Timetable">Timetables</a></li>
        <li><a href="#StationToStation">Station to Station</a></li>
    </ul>

    <div align="center" data-role="footer" data-theme="b">
        <h1></h1>
    </div>

```

```

</div>

<div data-role="page" id="Alerts" data-theme:"f">
  <div data-role="header" data-theme="f">
    <div class="pathlogo" data-role="content"></div>
    <a href="#page1" data-role="button" data-theme="b">Back</a>
  </div>
  <div align="center" data-role="header" data-theme="b">
    <h3>PATH Alerts</h3>
  </div>
  <div data-role="content" >
    <ul id="alert_content">

      </ul>
    <div id="alert_error"></div>
  </div>
  <div align="center" data-role="footer" data-theme="b">
    <h1></h1>
  </div>
</div>

```

```

<div data-role="page" id="Map" data-theme:"b">
  <div data-role="header" data-theme="f">
    <div class="pathlogo" data-role="content">
  </div>
  <a href="#page1" data-role="button" data-theme="b">Back</a>
</div>
<div align="center" data-role="header" data-theme="a">
  <h3>Map</h3>
</div>
<div id="system_map" align="center"></div>
<div align="center" data-role="footer" data-theme="a">
  <h1></h1>
</div>
</div>

```

```

<div data-role="page" id="Timetable" data-theme="c">
  <div data-role="header" data-theme="f">
    <div class="pathlogo" data-role="content"></div>
    <a href="#Schedules" data-role="button" data-
theme="b">Back</a>
  </div>
  <div align="center" data-role="header" data-theme="b">
    <h3>Timetables</h3>
  </div>
  <ul data-role="listview">
    <li><a id="Hoboken to 33rd" class="selection"
href="#TimetableContainer">Hoboken to 33rd</a></li>
    <li><a id="33rd to Hoboken" class="selection"
href="#TimetableContainer">33rd to Hoboken</a></li>
    <li><a id="Hoboken to WTC" class="selection"
href="#TimetableContainer">Hoboken to WTC</a></li>
    <li><a id="WTC to Hoboken" class="selection"
href="#TimetableContainer">WTC to Hoboken</a></li>
    <li><a id="Journal Square to 33rd" class="selection"
href="#TimetableContainer">JSQ to 33rd</a></li>
    <li><a id="33rd to Journal Square" class="selection"
href="#TimetableContainer">33rd to JSQ</a></li>
    <li><a id="Journal Square to Hoboken" class="selection"
href="#TimetableContainer">JSQ to Hoboken</a></li>
    <li><a id="Hoboken to Journal Square" class="selection"
href="#TimetableContainer">Hoboken to JSQ</a></li>
    <!-- <li><a id="JSQ to 33rd via Hoboken" class="selection"
href="#TimetableContainer">JSQ to 33rd via Hoboken</a></li>
    <li><a id="33rd to Journal Square via Hoboken" class="selection"
href="#TimetableContainer">33rd to JSQ via Hoboken</a></li> -->
    <li><a id="Newark to WTC" class="selection"
href="#TimetableContainer">Newark to WTC</a></li>
    <li><a id="WTC to Newark" class="selection"
href="#TimetableContainer">WTC to Newark</a></li>
  </ul>
  <div align="center" data-role="footer" data-theme="b">
    <h1></h1>
  </div>
</div>

<div data-role="page" id="StationToStation" data-theme="c">

```

```
<div data-role="header" data-theme="f">
  <div class="pathlogo" data-role="content"></div>
  <a href="#Schedules" data-role="button" data-
theme="b">Back</a>
</div>
<div align="center" data-role="header" data-theme="b">
  <h3>Station to Station</h3>
</div>
<div data-role="fieldcontain">
  <label for="selectmenu" class="select"><strong>From:</strong></label>
  <select name="selectmenu" id="from">
    <option value="9TH ST">9th St.</option>
    <option value="14TH ST">14th St.</option>
    <option value="23RD ST">23rd St.</option>
    <option value="33RD ST">33rd St.</option>
    <option value="CHRISTOPHER ST">Christopher St.</option>
    <option value="EXCHANGE PLACE">Exchange Place</option>
    <option value="GROVE ST">Grove St.</option>
    <option value="HARRISON">Harrison</option>
    <option value="HOBOKEN">Hoboken</option>
    <option value="JOURNAL SQUARE">Journal Square</option>
    <option value="NEWARK">Newark</option>
    <option value="NEWPORT">Newport</option>
    <option value="WORLD TRADE CENTER">World Trade Center</option>
  </select>
</div>
```

```
<div data-role="fieldcontain"><label for="selectmenu"
class="select"><strong>To:</strong></label>
  <select name="selectmenu" id="to">
    <option value="9TH ST">9th St.</option>
    <option value="14TH ST">14th St.</option>
    <option value="23RD ST">23rd St.</option>
    <option value="33RD ST">33rd St.</option>
    <option value="CHRISTOPHER ST">Christopher St.</option>
    <option value="EXCHANGE PLACE">Exchange Place</option>
    <option value="GROVE ST">Grove St.</option>
    <option value="HARRISON">Harrison</option>
    <option value="HOBOKEN">Hoboken</option>
    <option value="JOURNAL SQUARE">Journal Square</option>
    <option value="NEWARK">Newark</option>
```

```

        <option value="NEWPORT">Newport</option>
        <option value="WORLD TRADE CENTER">World Trade Center</option>
    </select>
</div>

    <div id="search_button" data-role="button">Search</div>
    <div class="separator"></div>
    <div id="result_container">
        <!--<ul data-role="listview" id="search_result" data-theme="c">
        </ul>!-->
    </div>
    <div align="center" data-role="footer" data-theme="b">
        <h1></h1>
    </div>
</div>

<div data-role="page" id="SearchTableContainer" data-theme="f">
    <div data-role="header" data-theme="f">
        <div class="pathlogo" data-role="content"></div>
        <a href="#StationToStation" data-role="button" data-theme="b">Back</a>
    </div>
    <div align="center" data-role="header" data-theme="b">
        <h3 id="searchtable_header"></h3>
    </div>
    <div data-role="content">
        <div id="searchtable_scroll">
            <table id="searchtable_display" style="width:100%">
                <tr id="search_column_headers">
                </tr>
            </table>
        </div>
        <div id="searchtable_error"></div>
    </div>
    <div align="center" data-role="footer" data-theme="b">
        <h1></h1>
    </div>
</div>

<div data-role="page" id="TimetableContainer" data-theme="f">

```

```

<div data-role="header" data-theme="f">
  <div class="pathlogo" data-role="content"></div>
  <a href="#Timetable" data-role="button" data-theme="b">Back</a>
</div>
<div align="center" data-role="header" data-theme="b">
  <h3 id="timetable_header"></h3>
</div>
<div data-role="content">
  <div id="timetable_scroll">
    <table id="timetable_display" style="width:100%">
      <tr id="column_headers">
      </tr>
    </table>
  </div>
  <div id="timetable_error"></div>
</div>
<div align="center" data-role="footer" data-theme="b">
  <h1></h1>
</div>
</div>

</body>
</html>

```

External Stylesheet (Index.css)

```

th, td {
    border: 1px solid #dcdfe5;
    text-align: center;
}

th{
    height: 25px;
    font-size: 14px;
    /*text-decoration: underline;*/
    color: #0099ff;
}

```

```
td{
    height: 25px;
    padding: 1px;
}

li.alert_items{
    /*height: 18px;*/
    font-size: 19px;
    line-height: 130%;
    margin-bottom: 16px;
    margin-top: 16px;
    display: 'block';
    /*flex-direction: 'column';*/
}

.date{
    color: #0099ff;
    font-size: 14px;
}

#timetable_error{
    color: red;
    font-size: 20px;
    text-align: center;
    text-decoration: underline;
}

#searchtable_error{
    color: red;
    font-size: 20px;
    text-align: center;
    text-decoration: underline;
}

#search_button{
    width: 100px;
    height: 35px;
    margin: auto;
}
```

```
    text-align: center;
    position: relative;
    left: 80px;
}

.separator{
    height: 2px;
    margin-top: 8px;
    margin-bottom: 4px;
    background-color: #0099ff;
}

.route_result{
    border: 1px solid #C8C8C8;
    /*border-bottom: 1px solid grey;*/
    padding-left: 4px;
    text-align:center;
    border-radius: 25px;
    box-shadow: 1px 1px 1px #888888;
}

.search_selction{
    vertical-align: central;
}

a.search_selection:link{
    color: black;
    font-weight: bold;
    text-decoration: none;
}
```


Server Side

Server.js (run the server as a whole):

```
var express = require('express');
var path = require('path');
var bodyParser = require('body-parser');

var mongojs = require('mongojs');

data = require('./data_to_mongo');

db = mongojs('mongodb://localhost:27017/mobile', ['timesheets']);

var index = require('./routes/index');
var timesheets = require('./routes/timesheets');
var alert = require('./routes/alert');

var port = 7004;

var app = express();

// view
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');
app.engine('html', require('ejs').renderFile);

// client folder
app.use(express.static(path.join(__dirname, 'client')));

// standard middleware body parser
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: false}));

app.use('/', index);
app.use('/api/timesheets', timesheets);
app.use('/api/alerts', alert);

app.listen(port, function() {
  console.log('Server running on port ' + port);
});
```

Data to mongo.js (move timetables from .csv into MongoDB):

```
command1 = 'mongoimport -d mobile -c timesheets --type csv --file timesheet.csv --headerline';  
command2 = 'mongo mobile --eval "db.dropDatabase()"';
```

```
const exec = require('child_process').exec;
```

```
load_data = function(){  
    exec(command2, (error, stdout, stderr) => {  
        if (error) {  
            console.error('exec error: ${error}');  
            return;  
        }  
        else {  
            /*console.log('Works');*/  
            exec(command1, (error, stdout, stderr) => {  
                if (error) {  
                    console.error('exec error: ${error}');  
                    return;  
                }  
            });  
        }  
    });  
};  
  
setInterval(function(){  
    /*console.log('Interval');*/  
    load_data();  
}, 24*60*60*1000);  
  
module.exports = load_data();
```

Timesheets.js (query data from MongoDB and make available through routes):

```
var express = require('express');
var router = express.Router();
/*var mongojs = require('mongojs');
var db = mongojs('mongodb://localhost:27017/mobile', ['timesheets']);*/

// Get all timesheets
router.get('/', function(req, res, next) {
  db.timesheets.find(function(err, timesheets) {
    if (err) {
      res.send(err);
    }
    res.json(timesheets);
  });
});

// Get single timesheet
router.get('/:id', function(req, res, next) {
  db.timesheets.findOne({ _id: mongojs.ObjectId(req.params.id)}, function(err, timesheet) {
    if (err) {
      res.send(err);
    }
    res.json(timesheet);
  });
});

module.exports = router;
```

Index.js and Index.html (for checking if server is up and running):

```
var express = require('express');
var router = express.Router();

router.get('/', function(req, res, next) {
  res.render('index.html');
});

module.exports = router;
```

```
<!DOCTYPE html>

<html>
  <head>
    <title>Path_Mobile</title>
  </head>

  <body>
    <h1>Test rendering content!</h1>
  </body>
</html>
```

Alerts.js (parse data from Path RSS feed into JSON object and make it available to client):

```
var request = require('request');
var cheerio = require('cheerio');
var parse = require('xml-parser');
var inspect = require('util').inspect;
var express = require('express');
var router = express.Router();
var pageToVisit = "http://rss.paalerts.com/rss.aspx?PATH";
/*console.log("Visiting page " + pageToVisit);*/

router.get('/', function(req, res, next) {
  request(pageToVisit, function(error, response, xml) {
    if(error) {
      console.log("Error: " + error);
      return res.send(error);
    }
    // Check status code (200 is HTTP OK)
    // console.log("response is: " + response);
    if(response.statusCode === 200) {
      // Parse the document body
      // console.log('html == ', xml );
      var obj = parse(xml);
      // console.log(inspect(obj, { colors: true, depth: Infinity }));
      var items = obj.root.children[0].children;
      var realItems = [];
      // console.log('items, are ==', items);
      for(var i = 0 ; i < items.length ; i++ )
      {
        if(items[i].name === 'item')
        {
          realItems.push(items[i]);
        }
      }

      var actualItems=[];
      for(var i=0;i<realItems.length;i++)
      {
        var tempObj = {};
        for(var j=0; j< realItems[i].children.length; j++)
        {
          if(realItems[i].children[j].name === 'description') {
            tempObj.description = realItems[i].children[j].content;
          }
          if(realItems[i].children[j].name === 'pubDate') {
            tempObj.date = realItems[i].children[j].content;
          }
        }
        actualItems.push(tempObj);
      }

      /*console.log("simplified items =", actualItems);*/
      return res.json({alert: actualItems});
    }
  });
});

module.exports = router;
```

B. USER INSTRUCTIONS

- a. Open app
- b. Select Alerts
 - i. View alerts feed
- c. Select Map
 - i. View map
- d. Select Schedules
 - i. Pick a route
 - ii. View timetable for route

C. SCREENSHOTS

