# 

# Interim Report

# Team Name: EventGPS

(Github Page link - <https://nhriday.github.io/>)

Abhinay Kandi

Aniket Kochar

Mehul Singh

Nirmal Hriday

Rui Tu

## 1. User Scenario: The Characters

* Who is your target user?

We believe our primary user base to be of two categories,

- The everyday commuter, who makes use of a particular route on a daily basis in order to notify the user of any events on their way, the traffic conditions and provide an alternate route if available, thereby helping them to not only save time by avoiding a certain route but also provide an alternate route to reach their destination.

- The tourist, who is not aware of any events in and around the city. Tourists usually have their whole trip planned, they have plans for each and every day of their trip and even the slightest disturbance to the plan due to traffic along their route because of a music concert or a big sports event can damper their whole day and eventually the trip altogether. If notified of the traffic and information about the event ahead, we believe our application can be a major help in making a tourist’s time in Dublin hassle free.

That being said, the application is not limited to the mentioned type of users, but can be used generally by anyone who wishes to be notified about any events along their route

* Why are they important?

The target users, as mentioned

- are the everyday men and women who are the workforce of this city. We believe in making their daily commute a bit easier and tension free so that they can concentrate solely on their work and not worrying about getting to work or coming back home.

- are the people who come to Dublin to enjoy and experience the beautiful city it is. We believe in making their time here as beautiful as possible so that they can enjoy the city as they wish and not worry about unnecessary traffic they might have avoided with prior notification.

* What problem are you solving for them?

Traffic is a major issue in Dublin, here are a few news articles on the effect of traffic that also served as a great motivating factor for our project.



“On average, commuters said they spent approximately 43 minutes each way on their daily commute to Dublin city. Over a year, this averages out to 14 days behind the wheel and some say the proposed changes could add 10 days to the existing total”



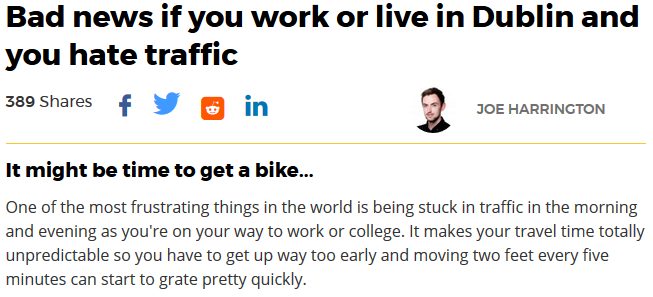
“A new survey says Almost three quarters of companies in Dublin have seen the negative effects of congestion on their business increase over the past three months. Dublin Chamber shows that 73% of its members said traffic problems in Dublin have had an increasingly negative impact on their business since the start of December.”



“Time lost due to aggravated traffic congestion in the greater Dublin area is costing the economy €350 million per year with the amount set to rise to €2 billion by 2033. The Department of Transport’s Economic and Financial Evaluation Unit has undertaken a research project to estimate the cost of aggravated congestion across Ireland’s transport system.”



“A new index of world cities has revealed that Dublin is the "slowest-moving" major city in the world when it comes to traffic congestion. The report by Bank of America Merrill Lynch found that during congested periods in the capital, the average driver reaches a speed of 7.5km/h, with peak hour speeds of 5.5km/h - slower than a horse and cart.”



The report says, “Traffic heading into and out of certain parts of Dublin city is an absolute nightmare and unfortunately, it's going to get a lot worse as traffic is set to increase by 50,000 journeys a day in the next nine years, according to the Department of Transport.”

The primary problem of our target users we aim to solve is that of avoiding unnecessary traffic. Usually, most daily commuters may use an application that provides them the route to their destination and also frequent commuters who are well versed with the route may not use any application at all. The problem with both these approaches are neither of them are aware of any events along their route which might have led to a high traffic scenario which eventually may result in the user stuck in traffic and getting late to work, getting home or simply put, reaching their destination. Our application helps them avoid this by notifying them about any events along their way, details about the event such as capacity and an estimation of the number of vehicles, start time and probable end time of the event, and most importantly an alternate if available so that they can reach their destination.

## 2. Technical Problem: The Setting

* Why does your system exist?

When we first set out to develop this application, we had two reference points. Ticketmaster and Google Maps. Ticketmaster is a website that sells tickets for almost every event in Ireland, specifically Dublin as we are building a database that consists of events in Dublin. Google Maps is an application used by almost everyone to reach their destination and its use ranges right from taking a walk to planning a trip. We believe our system exists as an effective culmination of the best features of the two applications

* Challenges

1. **Estimate end time** – The ticketmaster API does not give us the end time of the events, only the starting times. We estimated the end time of the events based on the type of event, capacity of the venue and historical data for the large venues.
2. **Predict traffic with radius** – The primary logic behind notifying the user about the event and the traffic around it, is the route of the user being in a specific vicinity or area around the venue. We decided to define an estimated area around the event by calculating the radius for a specific distance. This was done with help of a website ‘www.freemaptools.com’. The website allows us to enter the geographical co-ordinates of the venue and also the required radius and then draws an area around the location. We plan to use this to estimate the radius and notify the user about the event and the traffic around it.

Along with the radius, we also came up with something known as **Pre-start time** – in case of larger events, which is our primary target to predict traffic, the traffic will be higher or start to increase maybe an hour or two before the start of the event; and **Post-end time** – similarly, the traffic is high after an hour or two after the event has ended. We used this time difference to predict the traffic based on the type of events.

1. **Accessing Flask through our client** – The web API written in flask to process requests from the server**,** is to be accessed by public ports which were not made available by UCD. In order to access it using the public internet we need access to the ports on the server. We communicated with IT services in solving this problem.

1. **Data Refresh** – The data from ticketmaster API is unstructured and inconsistent. We encountered a lot of missing values including names of venues and events, capacity of the venue, starting time and other information. We carefully filled in the missing values by referring other entries and historical data.
2. **Locating venue along the route** – Locating the venue along the route is one of the primary objectives of the application. We plan to do this by calculating the distance between the geographical points of the venue and the points on the route to make sure it is as accurate as possible. The geolocations are obtained using Google Places API and subsequently used to calculate the distance

* Can you review other existing systems or products that address this problem?

1. **Google Maps** - Google Maps is a very effective navigation tool which notifies users about real time changes along the route. It also notifies the user of traffic ahead while commuting by displaying the intensity of traffic by different colors, red being the indication that traffic is high and also provides an alternate route(s). This does not give the user a clear idea of the traffic. We want to provide the user with an additional feature by notifying them about the traffic with information about why the traffic is high and an alternate route(s) if available. If the user is aware of a large-scale event ahead, say a music concert with about 10000 people. This helps the user get a clear idea of the situation and plan the commute accordingly. We plan to emulate the service provided by Google Maps in our own unique perspective

## 3. Technical Solution: The Plot

* What does the system do?

**Technology involved**

* **Front-end**:
* **Android**
* Our front-end client is written in Android, which is an interface to help the user navigate through the application.
* **Maps SDK**
* SDK used to manage access to Google Maps server and display the map layout in our client (application)
* **Google Places API**
* API used to get automated suggestions from Google Maps when a user types in the name of a location. It also helps in obtaining the geo-locations of the searched location
* **AwesomeSplash**
* used to create a custom splash screen, the screen that pops up when the app icon is clicked on. A library developed by Viktor Arsovski and is available at <https://github.com/ViksaaSkool/AwesomeSplash>
* **Back-end:**
* **Flask Server** – handle the requests from the client and call the Google Maps API for relevant information regarding routes and return the event details by accessing the database depending on the request, which is the date and time entered by the user or system time by default and the route information returned from the Google Maps API
* mySQL database – stores all structured event data including venue, start time, title, capacity primarily
* **Data:**
* Ticketmaster API – Information about events was extracted using Python

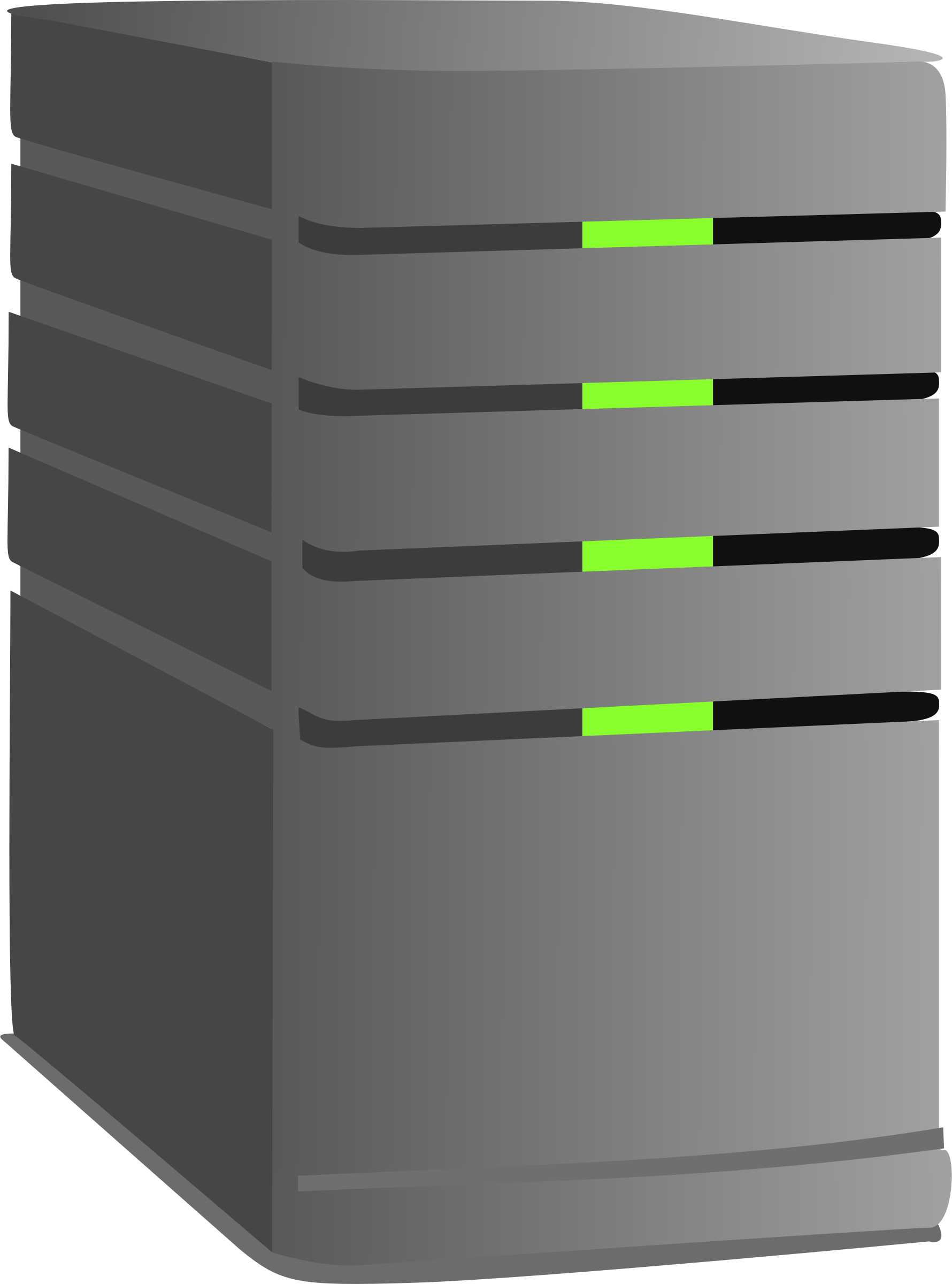
**The Process**

The very first step is the user entering the source and destination of the commute planned. This is done on our Android client to set the two parameters.





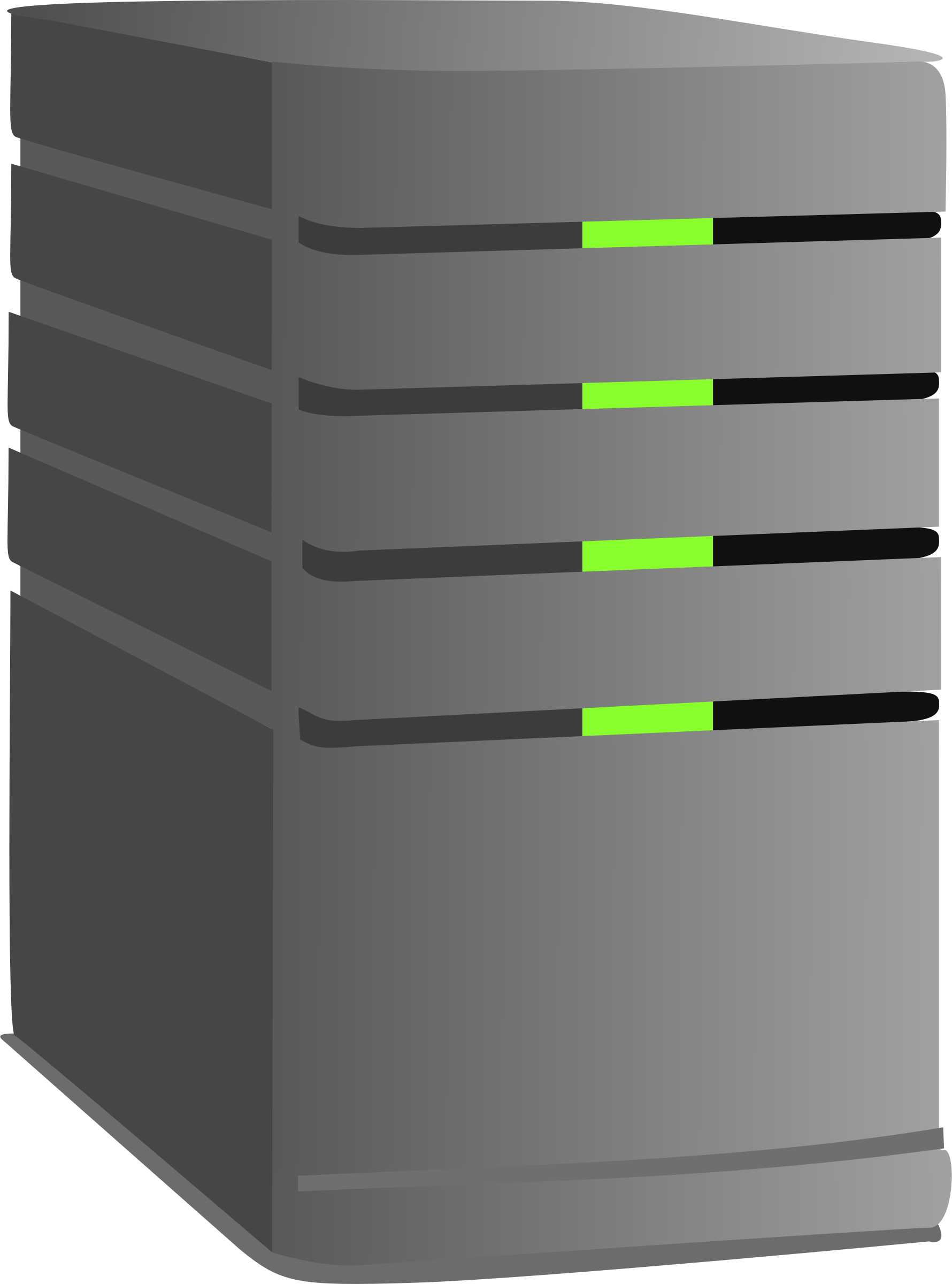
This request is sent to our back-end server, the client interacts with the server to get back the specific routes and event info to the user.





There is a process followed to get the routes, the server interacts with the Flask web API to do the same.

* The flask web API interacts with two entities and returns back two webpages.
* The first webpage is the routes specific to the user, by interacting with the Google direction API to obtain the geo-locations of the locations requested from the client. Using these geo-locations, a route to the preferred destination is displayed on the screen.
* The second webpage is the information about every event on the day, which is stored on the mySQL database. The web API interacts with the database to extract relevant data and sends it back to the client through the server.





The mySQL database stores data obtained by the ticketmaster API to get the information about events including

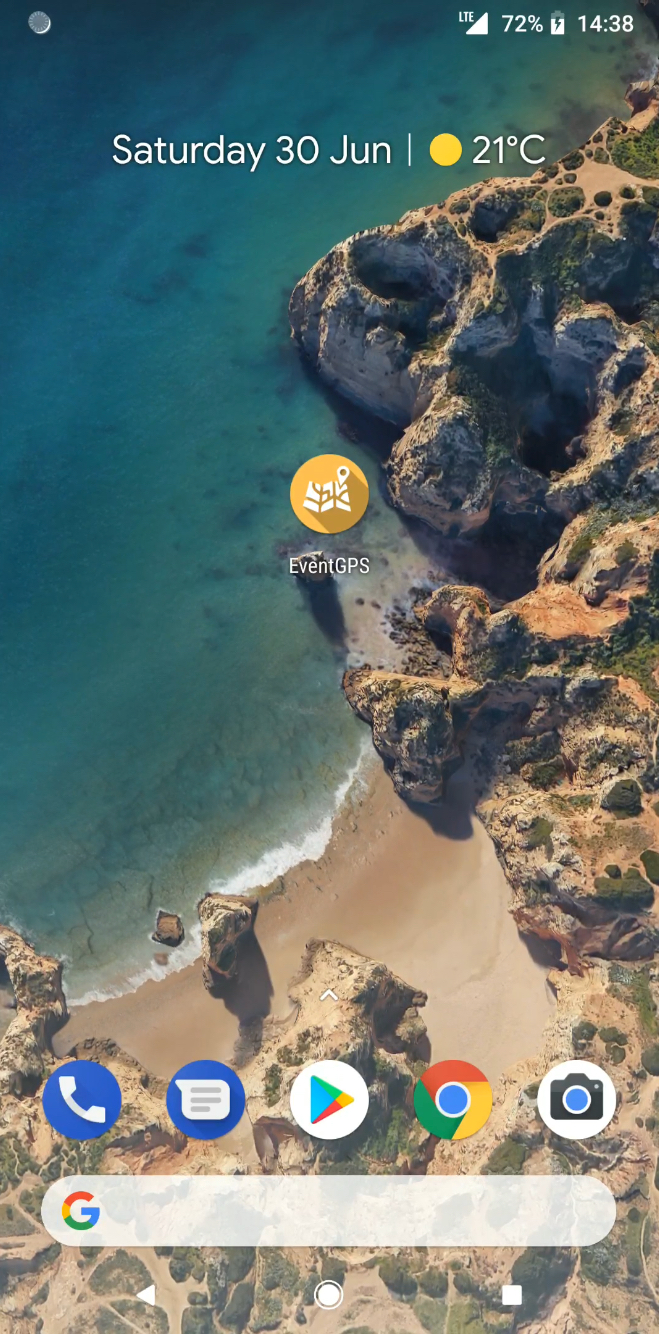
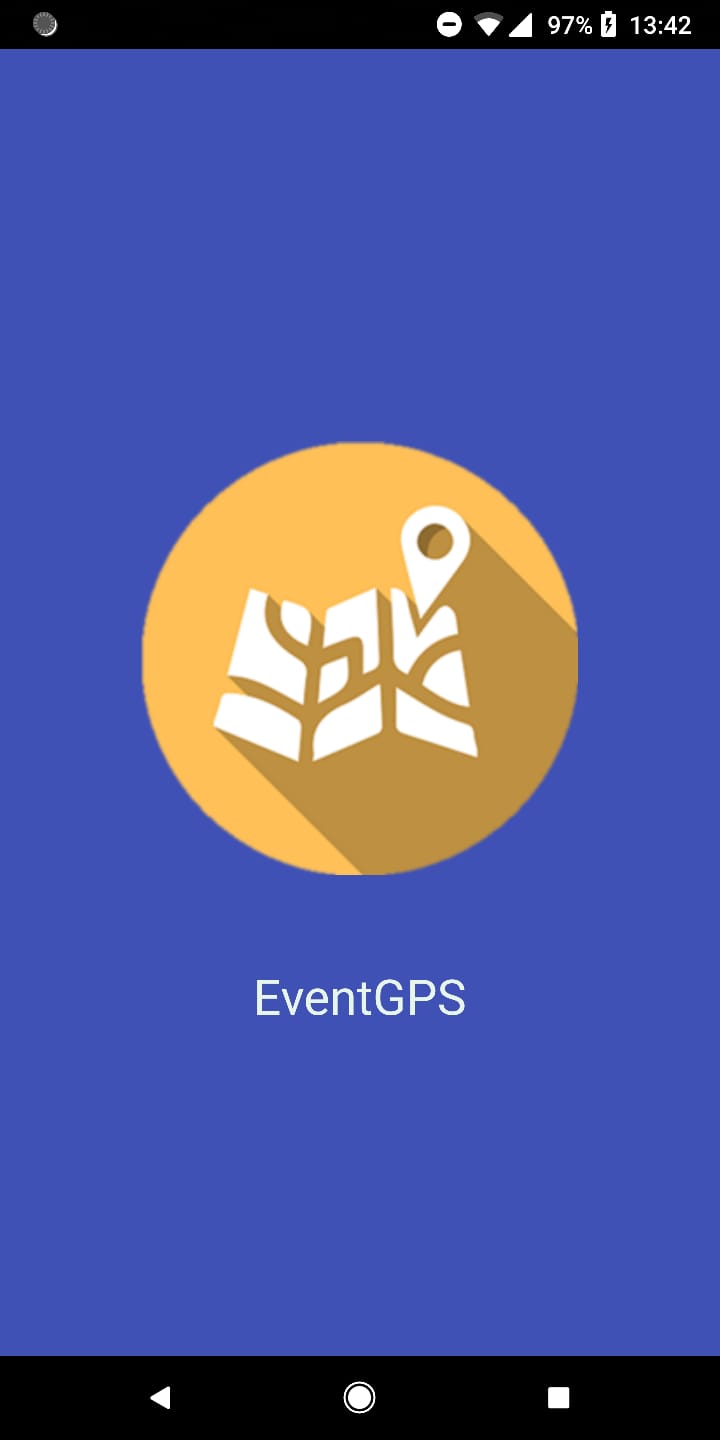
* + Event title & Venue
  + Start time, End time (estimated)
  + Capacity of the venue



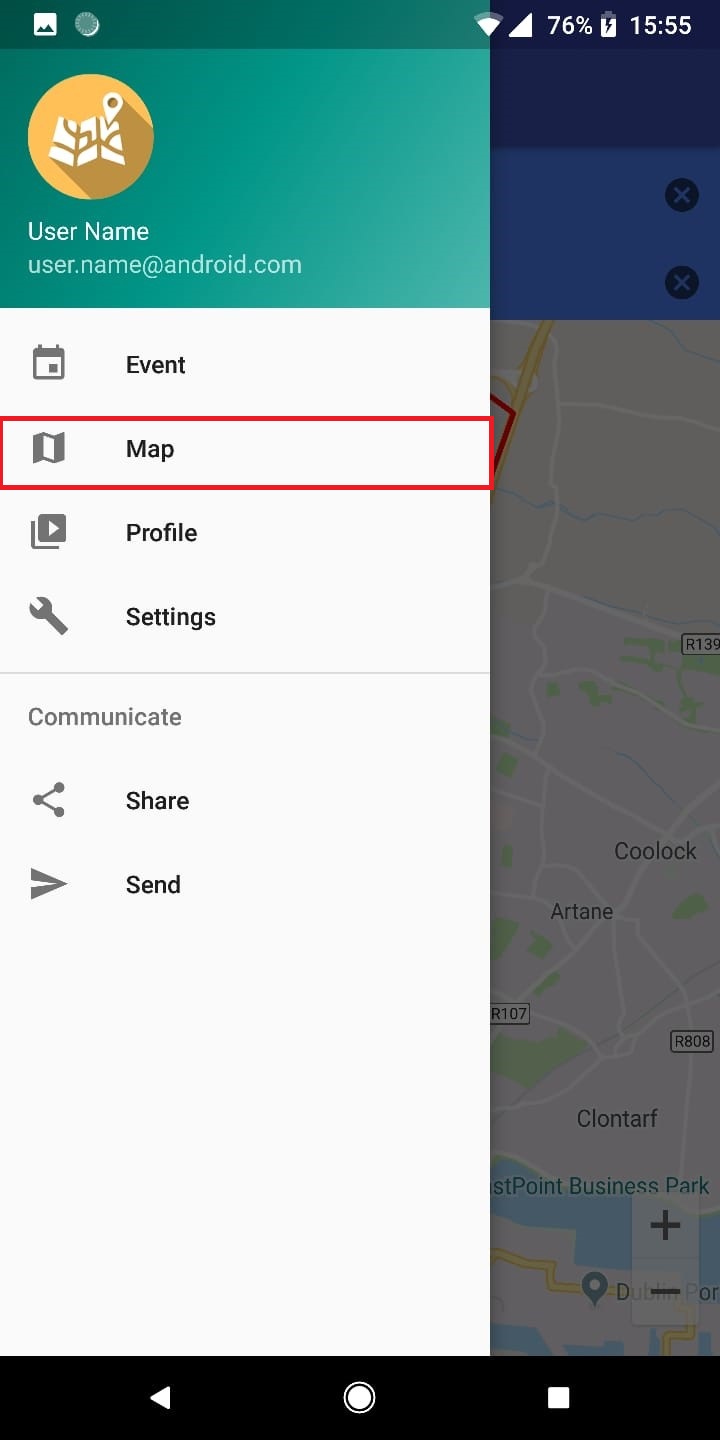


**App Walkthrough**

* **Start-up –** Click on the EventGPS app icon, the following screen is shown on the screen containing the app logo

* **Map Function –** this is the first page displayed when the app is opened, and can also be chosen from the side bar

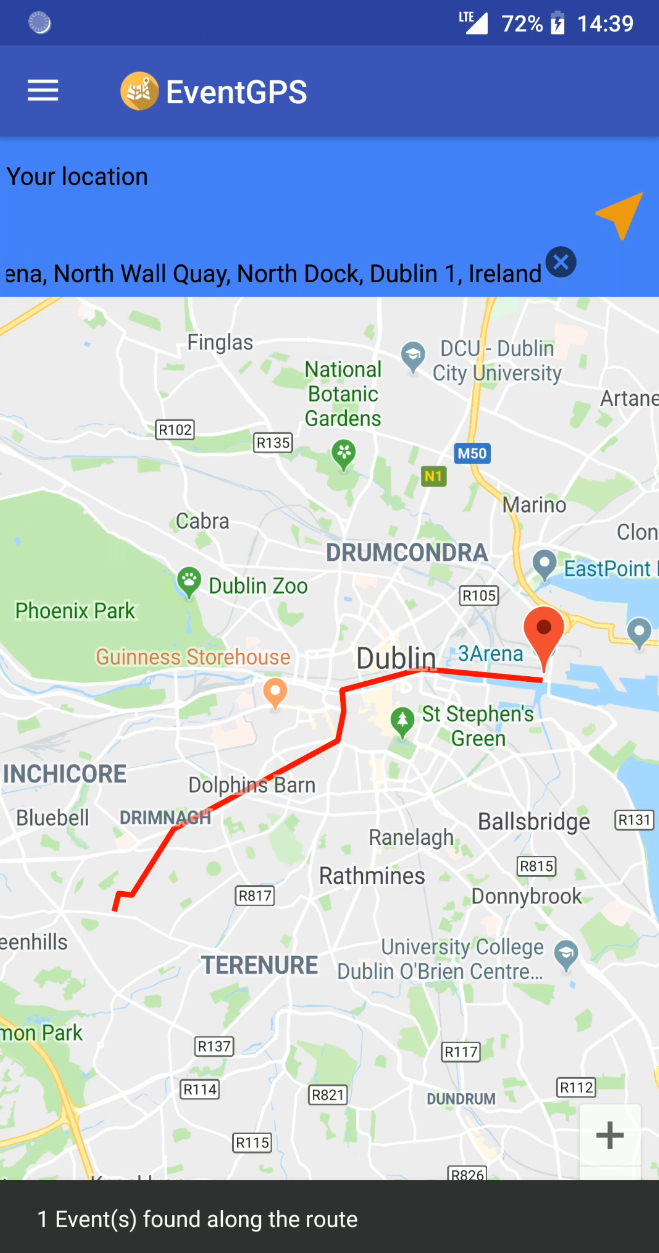
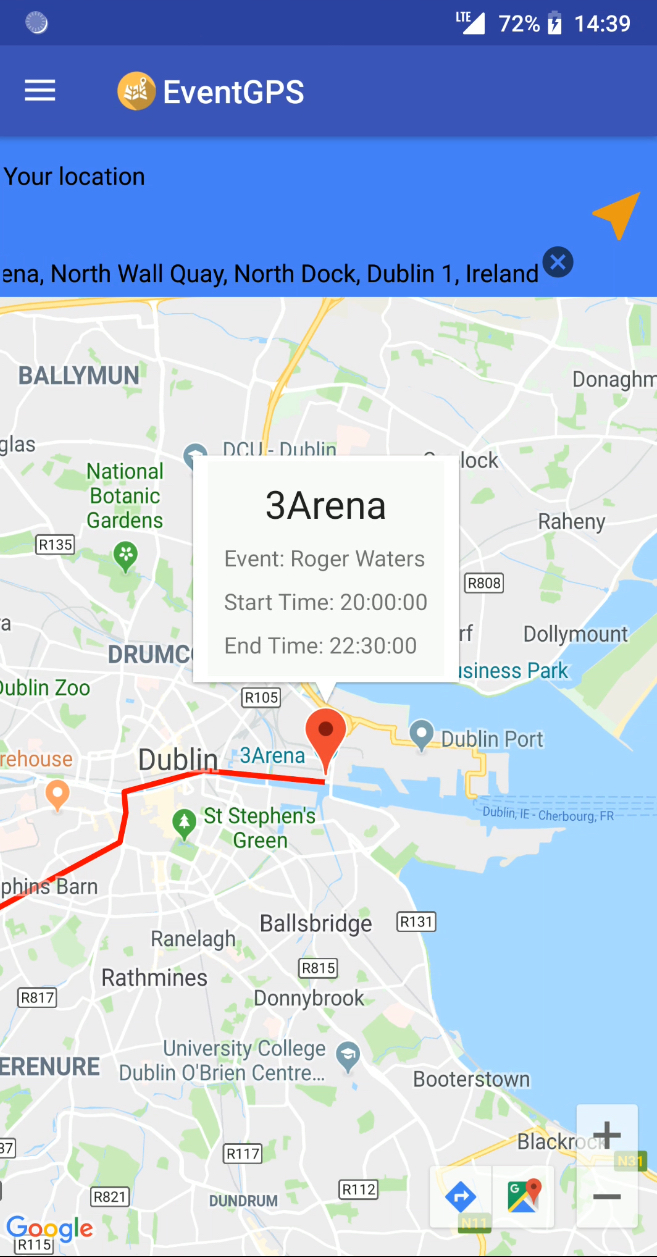
* Enter Source and Destination, the route is and the event along the route is displayed. The event is shown as a marker(), the marker can be clicked upon to view the event details. The user will be notified whether there is an event along the route or not on the bottom of the page

Example 1

**Source** – current location

**Destination** – 3 Arena

**Result –** 1 Event(s) found along the route, 3 Arena

Example 2

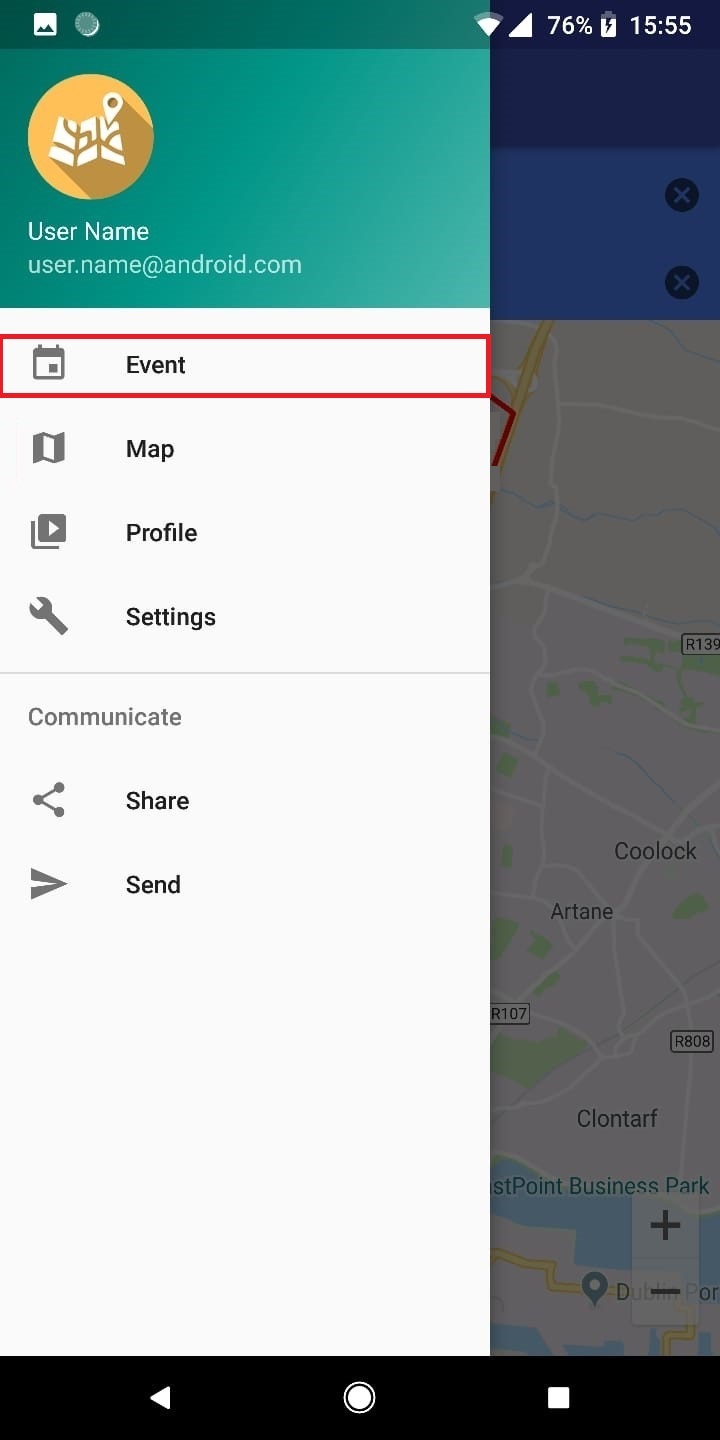
**Source** – Rathmines

**Destination** – Dublin Spire

**Result** – Currently no event



* Current day events – Choose the ‘Event’ option from the sidebar and click on the event calendar button on the bottom right-hand side of the screen. The events of the day in real time are shown with markers and can be clicked upon to view the event details.

## 4. Evaluation: The Reviews

* What does success look like for your system?

A user enters a preferred source and destination using our application, in

return the following is available on the display -:

* A route to the destination
* Event markers along the route
* Notifications to the user about the traffic conditions
* Event info including title, time details, capacity of the venue
* Traffic conditions close to the start/end time of the event
* Alternate route(s) if available

This would be a successful system

* How will you evaluate the system that you built?

We plan on evaluating our system during a live large-scale event at which we believe our system can be evaluated thoroughly. The event we have planned to evaluate our system at is a cricket match between India and Ireland at the Malahide Cricket Club Ground in Dublin

## 5. Conclusion: The Plan

* **Project management strategy**

We are following the Agile project management strategy. We are following a system where we have two-week sprints. Each of us have a specific task assigned and the scrum master (Mehul Singh) is the buffer for any type of information exchange. The following illustration represents the process flow

In addition to the above sprints, we discuss the progress and issues at least every two days over text messaging or Skype. At the end of each week, all of us make sure we meet face to face at UCD mostly to discuss the progress and get ready for the next sprint. Initially we divided ourselves into two teams of 2 and 3 to work on front-end and back-end development respectively. After the second sprint, the whole team took on the responsibility of writing the report and readying the presentation. All 5 of us met at the end of the second sprint and gave inputs on what each of them worked on and came up with the report and the presentation.

* **Biggest challenges**

1. **Accurate Routes:**

The challenge we believe is obtaining the accurate routes in order for the commuter to be able to use it effectively. Google Maps Direction API for Android does not provide routes and navigation information. We cannot get the routes by calling the Google Maps on the phone, only the Maps JavaScript API provides the routes.

We plan to overcome it by sending the geolocation data from the Google Places API to the server to make API calls instead of making HttpConnections to call the API by the client. This helps us to send back the exact route data back to the client by drawing polylines on the map. Obtaining the geo-locations also helps us to provide alternate routes.

1. **Data Refresh:**

This is the biggest challenge we faced. To elaborate on this topic, previously mentioned in the **Technical Problem 🡪 Challenges**, the event data is the integral part of the whole application. These are the following observations we did while we received data from the Ticketmaster API,

* The data is very inconsistent
* Geo-locations of venues were missing
* Information of some events are available on a partner site; the user is redirected to buy the tickets and event information is available on the partner site.
* Basic information including event name, time, date was missing. This is mostly for the smaller events
* API does not give us the capacity and end-time of the event
* Event details of only 1000 events can be obtained
* Each page contains only 20 events
* Almost every page has one or the other information missing

We had to carefully fill up all the missing values. All of us divided the tasks among ourselves by taking up 10 pages initially and fill up as much as possible and we ended up with a complete database of events with complete event information up to November 2018. We had to refer other sources to fill up data, we referred google for the capacity of the venue, end-time was predicted by observing historical data, type of event, popularity of the event, a deadline of 11pm set by Dublin City Council for a few events. Geo-locations of events were referred on Google Maps. Due to the inconsistency in the data when the API is called, it is not possible to update the data on a daily basis and provide the user with the most recent update of events.

* How will you use the time remaining to achieve a successful outcome?

**The road ahead**

According to our plan, we have 4 more sprints scheduled, each sprint task has been shown below individually,

* **Sprint 4**

* **Sprint 5**

* **Sprint 6**

* **Sprint 7 (Final week)**

* List of resources (software, papers, tutorials, books, stats, business indicators)
* Github
* Microsoft Word
* Microsoft Powerpoint
* <https://github.com/ViksaaSkool/AwesomeSplash>
* <https://pages.github.com>
* <http://flask.pocoo.org/docs/1.0>
* <https://www.tutorialspoint.com/flask>
* <https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask>
* <https://developer.ticketmaster.com/>
* <https://developers.google.com/maps/documentation/>
* Android Studio
* Jupyter Notebook
* Youtube
* Google
* Stack Overflow
* News articles - <https://www.rte.ie/news/business/2018/0316/948010-dublin-chamber-traffic/>

<https://www.irishtimes.com/news/ireland/irish-news/time-lost-in-dublin-traffic-costs-economy-350m-per-year-1.3066581>

<http://www.thejournal.ie/worlds-most-congested-cities-3377455-May2017/>

<https://www.joe.ie/news/bad-news-if-you-work-or-live-in-dublin-and-you-hate-traffic-472880>

<http://www.engineersjournal.ie/2017/06/20/dublin-congestion-e2bn-in-2033-without-corrective-action/>

*Team Name:*