# Extra Feature Essay

To auto-suggest the closest stop for the user:

Obtain the user’s current location.

Obtain the user destination.

* Check against past destinations from the same approximate source.
* Check against searches for items in the area the user has frequented.

Check how fast the user has been walking or moving at the time he uses the app

– It could indicate if they are in a hurry or not to get to the destination.

Extrapolate the closest known bus stop using location data,

* Check for the last searches for a bus stop in the same area.

Check the next arrival and departure time of the bus.

– It is pointless directing the user to the closes bus stop if he has to wait for a long time, when directing him to one slightly further, or possibly more transit hops, but can get him to his destination quicker or in a less crowded transit.

Interface with Google’s traffic with the app to indicate congestion on the desired transit route.

* Compare other bus (or passenger) data from Google maps against the RTPI expected arrival and departure times to see if previous buses experienced delays.
* If there is congestion, build approximate alternative suggestion.
* For repeated delays, suggest to the user an alternative time (flexi-time) for commuting and the time saving difference. Interact with congested time periods vs slightly off-peak hours of the RTPI tables.

Interface to Google’s traffic app for anonymous crowd sourced information to indicate the amount of people waiting for a transit bus.

* If there is congestion at a particular stop, build approximate alternative suggestion.
* Take into account like how crowded the bus may be at that particular time – another bus stop with a similar bus and destination may not be as crowded.

Offering the user alternatives would be ideal to help them achieve their goal, but in a better or easier way.

Having the third party applications interface with each other via JSONP could be easily done.