**Final Report – Don’t Stop**

**9 Apr 2015**

**ECE 1778**

Word Count: 2,467

1. **Introduction**

Transit agencies collect data such as passenger counts, and time-at-location data, in order to produce key performance indicators which help the agency measure and improve its efficiency and quality of service. This app is intended to assist transit agencies – i.e., not transit *passengers* – with accurate collection, storage, and analysis of field data generated by surface vehicles (i.e., buses and streetcars) while in operation.

In this pursuit, the app performs the following functions:

* Collection of location and speed data;
* Cleaning and processing of gathered data;
* Communication with a remote server to gather pertinent data about a route of interest, including estimates of vehicle dwell times (time loading and unloading passengers at stops); and
* Description of results for a given trip.

Specifically, the app can be used to assist transit agencies with route planning. Existing desktop software (being produced as part of Alec’s thesis) allows a transit agency to (virtually) add/delete stops to a route to estimate how much time this would cost/save the bus and passengers, based on historical ridership patterns. However, the desktop software relies on average historical ridership to estimate vehicle dwell times; this app provides more accurate data by *measuring* dwell times, rather than estimating them, as well as providing related metrics. These metrics are discussed in greater detail in Section 2.

Currently, to determine such time savings/costs of adding or deleting stops to a route, transit agencies have two options: i) change a route in reality and spend months collecting new performance and ridership data, or ii) spend large amounts of time and money completing sophisticated modelling of existing routes. The goal the desktop software is to enable transit agencies to estimate instantly the impacts of route changes, and the goal of the app is to both validate and improve the results in real-time while the user rides a bus. This has significant implications for improving the efficiency and accuracy of results of the transit planning process.

1. **Overall Design**

The app records GPS data and vehicle speed in real-time, twice per second, and cleans the data to determine the true vehicle dwell times and average vehicle speed over the course of a trip. This data is used to produce the following key performance indicators:

* Average trip speed;
* Total average passenger boardings and alightings;
* Total actual vehicle dwell time, and actual vehicle dwell times by stop; and
* Total estimated dwell times, and estimated dwell times by stop (from historical data).

The app also produces a plot showing the vehicle’s real-time speed profile. Refer to Section 3 for a screenshot of this function.

A block diagram showing the functional design of the app can be seen in Figure 1 below:

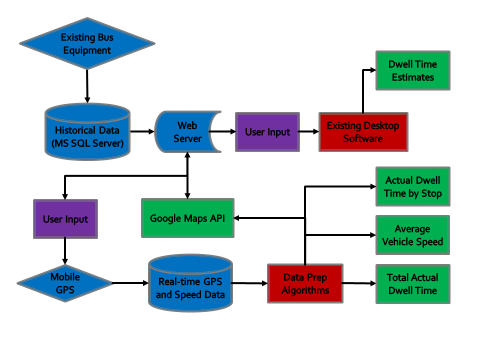
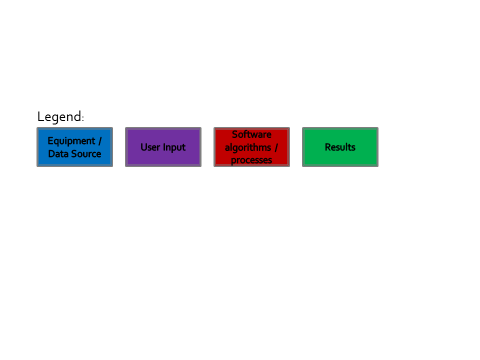


Figure 1: Block diagram showing app functionality

As the block diagram shows, the user’s selected route-information input will communicate with a server, which is in turn connected to a SQL Server database hosting the transit agency’s historical data. (This historical data is generated by existing equipment installed on vehicles). The user then sets desired filters/parameters for a route, and proceeds with analysis. The server returns existing dwell time estimates for the route of interest, and the mobile device’s GPS will gather data. Finally, the app computes results (key performance indicators).

1. **Statement of Functionality**

By and large, the app works as intended. Some functionality was adjusted, some was achieved by means other than that originally envisioned, and some intended minor functionality was cut.

Successful app functionality:

The core of the app functioned well, and is described as follows.

Navigation within the app is simple. Four pages are available to the user (as seen in Figures 2-5 below). Four icons are visible at the base of each page, with each icon serving as a link to one page. When using the app, the user proceeds from the first page to the last, from left to right.

Page 1: User Input page (Figure 2):

An example instance of a transit agency’s database was set up on a server, and connected to the internet. Filters were set up to allow the user to specify what information should be returned from the server, primarily using a series of dropdown menus, and data was successfully returned from the server to the app based on these requests.

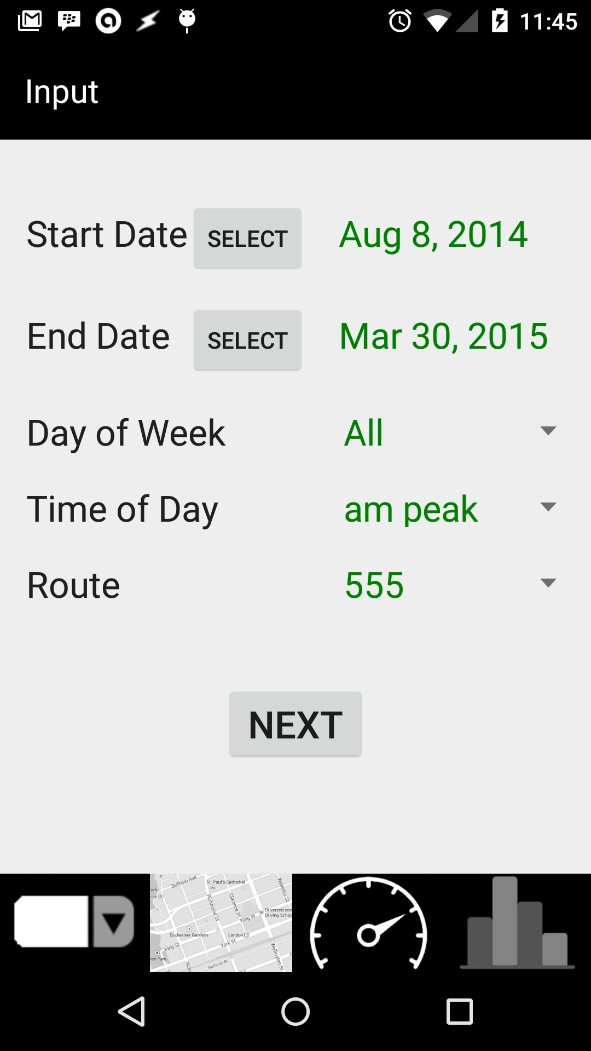
The “Next” button tells the app to contact the server to retrieve the route information specified, and then to show a plot of the route in Google Maps.

Figure : App page 1 (User Input)

Page 2: Map of Route page (Figure 3):

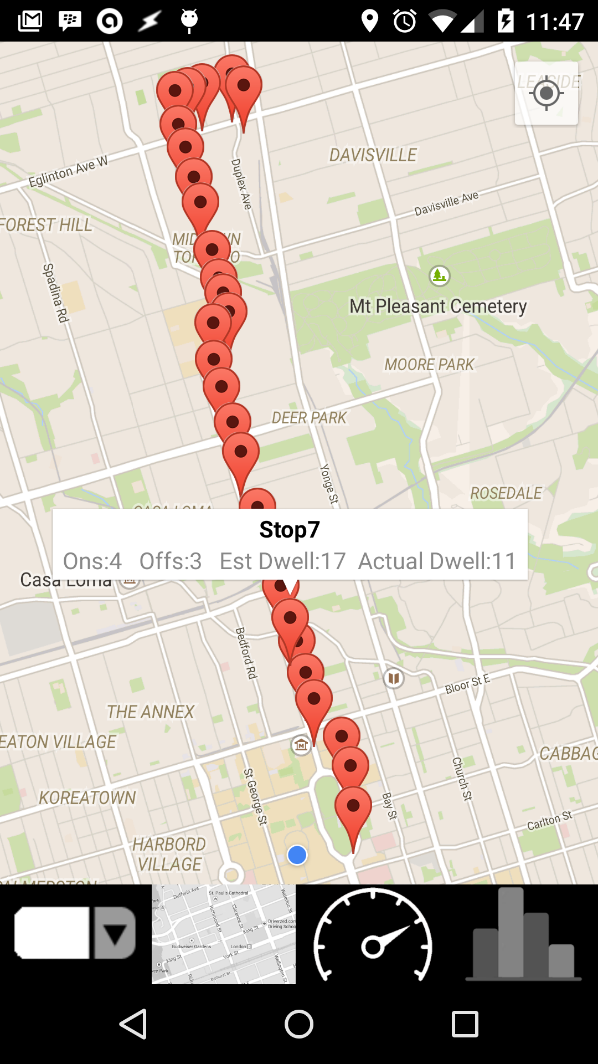
****When the user presses the “Next” button on Page 1 as described above, they are taken to a Google Maps plot of their selected route in order to confirm they have selected the correct route. This page also displays information pertinent to individual stops (returned by the server), if requested, though information related to the current trip is only made available to the user after they have completed the trip while using the app. Figure 3below shows the plot following a completed trip, where all information is present.

Figure : App page 2 (Google Maps route plot)

Page 3: Real-time Speed Page (Figure 4):

Once the user has reviewed the route plotted in Google Maps, they are ready to begin their trip. Page 3 presents the user with “Start” and “Stop” buttons, with the expectation that the user will press “Start” as the trip begins, and “Stop” once finished. When the user presses “Start”, a plot showing the vehicle’s real-time speed appears, changing dynamically as the vehicle travels. This plot continues until “Stop” has been activated. Similarly, the user is presented with a numerical reading of the current speed of the vehicle once the “Start” button is pressed.

When the “Stop” button is activated, the trip is assumed to be finished, and the data recorded by the app is saved to file.

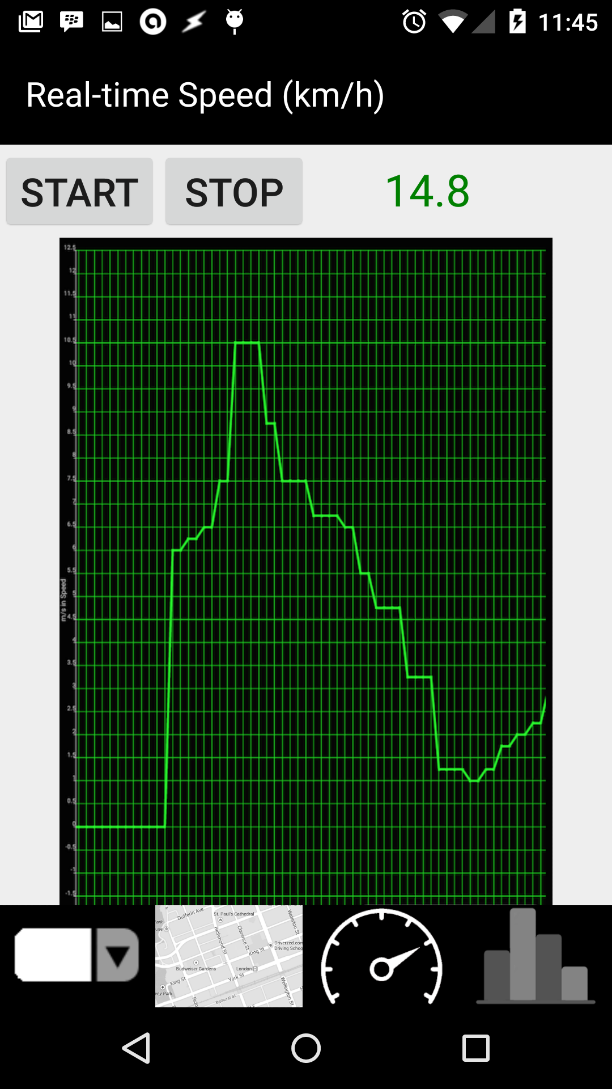


Figure : App page 3 (Real-time Speed Plot)

Page 4: Results Page (Figure 5):

This page displays results of a trial once the trip is complete (i.e., once the user has pressed “Stop”). These results are calculated using both historical data (returned by the server) and data gathered by the mobile device which is saved to file as described under “Page 3” above. The app shows the user the historical estimate of vehicle dwell times, the true dwell times recorded by the vehicle for the recent trip, and the average speed of the bus during the recent trip.

Aggregate information is displayed on the results page (as shown in Figure 5 below), but stop-specific information is shown using Google Maps (refer to Figure 3 above).

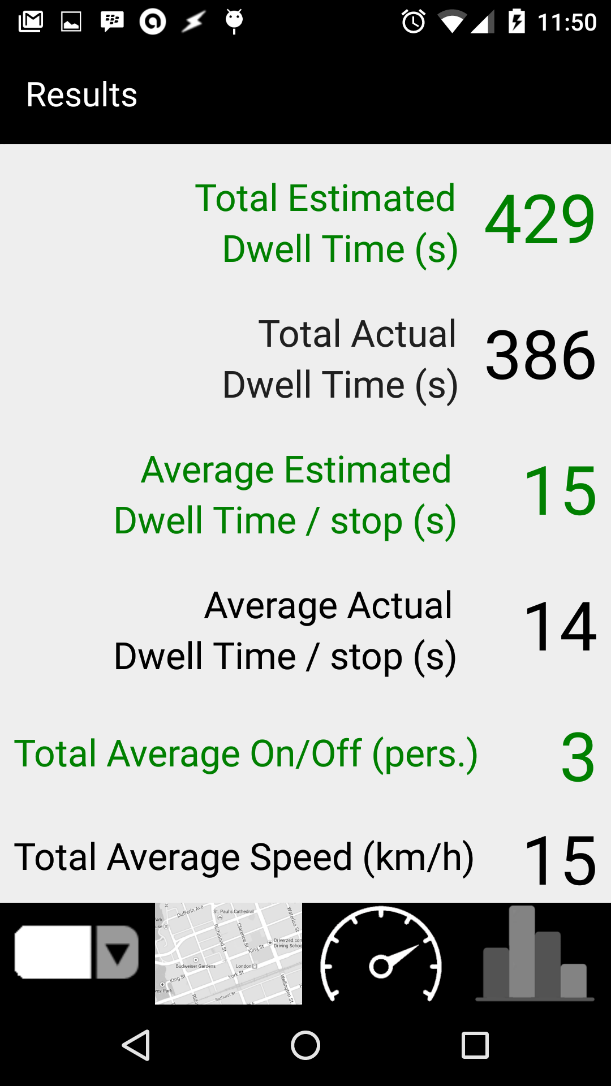
Finally, the app and server were programmed such that it is trivial to return to the app text-based results produced by the desktop tool being developed separately. This would enable the user to conduct desktop-level analysis while in the field, at meetings with city staff, etc. However, an interface for displaying this information cleanly and meaningfully was not produced as part of this app, as this functionality was considered of secondary importance. Producing effective visualizations, the ability to drill up and down, and smooth functionality – that is, producing quality dashboards – is itself a significant undertaking and was therefore left as a future work item (see Section 7 for more information).

Figure : App page 4 (Results)

Unsuccessful app functionality:

Geo coordinates were not recorded with as much accuracy as desired. The mobile device appears to experience occasional flakiness when trying to connect to GPS, or lag time therein. (It was found that when geo data appeared accurate, as confirmed manually while riding a transit vehicle, the algorithm written to identify stops and dwell times worked very well. However, when geo data was erroneous, the algorithm produced inaccurate results). While the results obtained were accurate enough for proof of concept, improved accuracy would be needed before developing this app into a commercially viable tool (see Section 7 for more information).

Initially, we intended to take measurements from the accelerometer to help determine when vehicles were arriving at and departing from stops. Additionally, we wanted to determine period when vehicles were cruising at maximum inter-stop speed. However, readings from the accelerometer proved too jittery to be useful. Similarly, looking back at speed plots produced by the app, it is usually impossible to determine a specific maximum speed achieved by vehicles between stops, as speeds tend to change very frequently, and this is compounded by any error in GPS readings.

1. **Learning Experience**

One of the most useful and transferable skills gained through production of this app is the ability to communicate information between a webserver and an internet-connected device. This necessary programming was relatively straightforward (once achieved), and is a crucial skill for any future app production. Similarly, connecting with Google Maps via its API was a good learning experience and something that may well come in handy in many future capacities, including outside the world of mobile app development.

We learned that calibration of and/or extensive testing of sensors is a necessary part of designing apps which take advantage these sensors. At the outset of the project, we had thought that simply engaging the sensors would be sufficient; however, like most equipment which produces and stores data, mobile phones require considerable data cleaning, leading to a classic “garbage in, garbage out” situation.

Similarly, it is necessary to produce data cleaning algorithms which are robust enough to handle a wide range of eventualities. Extensive testing of the app is required because it is impossible to envision every way in which an app can malfunction. For example, we discovered that rotating the app while recording GPS data can throw off the recording process unless otherwise prevented.

Finally, we learned that building an app is like any other project: it is better to complete a small number of tasks well, with quality results, rather than tackle too many tasks and produce lesser results.

1. **Contribution by Group Members**

Alec (Apper) was responsible for the following:

1. Planning, vision, and overall direction, including specification of functionality and features.
2. Management (including storage, cleaning, and analysis) of data. This included existing warehoused data, new data gathered by the mobile device, and interaction of the two datasets. Alec was responsible for creation of logic governing data analysis algorithms.
3. Primary communicator (writing, presentations).

Yunlei and Yue (programmers) worked very closely together, and were responsible for the following:

1. Generation and retrieval of the data, and programming of the algorithms in Java.
2. Plotting real-time GPS data.
3. Two-way communication with server.
4. Setting up Google Maps API, and embedding data.
5. Constructing the app, and tweaking appearance, functionality, and stability.
6. **Apper Context**

With more than half the world’s population now living in urban areas (United Nations, 2011), it is becoming increasingly clear that efficient urban transportation is critical for continued worldwide growth. Combining this shift to urban regions with a projected world population growth from approximately seven to 11 billion people between 2012 and 2100 (United Nations, 2012), large-scale methods of transportation which minimise costs (financial, environmental, spatial) are no longer an option so much as an urgent necessity. Public transit systems represent the most obvious single tool to meet this need.

Minimising costs while maximising quality and customer service is essential to the success of transit agencies and, by extension, the municipalities that fund them. Historically, transit agencies have relied on manual gathering and analysis of transit data (e.g., surveying passengers, estimating timeliness of buses, etc.) to make decisions related to operations and planning. However, manual data gathering and analysis is slow and labour-intensive, with high marginal cost, and the net result has been reliance upon incomplete and infrequent analysis of sparse data sets. Automatically generated transit data, however, can be gathered in very large quantities at very low marginal cost. Analysis of this so-called ‘big data’ can be automated and completed in real time, also at very low marginal cost.

Alec’s MASc research also involves creation of data analytics software to analyse automatically generated data from sensors (such as passenger counters, GPS devices, etc.) on buses to enable fact-based decision-making by transit agencies.

The app created for this course will, when fully rolled out by a transit agency, allow for validation and improvement of the algorithm Alec is using to produce his thesis’s desktop software. For example, currently, Alec relies on a number of assumptions about a bus’s dwell time (theoretically this information is already being recorded by the buses, but the data is dirty and often erroneous, and therefore cannot be relied upon). By using the mobile app, dwell time assumptions can be replaced with actual data for the trip in question, and for the sake of other (future) trips, the assumptions can be more finely tuned. Ultimately, this app can be used to improve the quality collected by transit agencies, and in so doing, help set the stage for more efficient transit with better customer service.

1. **Future Work**

The most important thing to improve with the app moving forwards is to improve the accuracy with which the GPS records data, in order to be confident that results are consistent and correct. Even if results are largely correct, if consistency if not achieved, it will be difficult to place complete confidence in results. Further, the accelerometer could be incorporated, given sufficient time to clean the data it produces.

A number of changes could also be made to the app to expand its functionality.

In the short term, additional performance indicators could be built using supplemental data from the server, in conjunction with gathering more data using the app. For example, it would be feasible to calculate the on-time performance of vehicles (the relative frequency with which vehicles are on time at a stop), and the average time/distance that a user is on board a transit vehicle.

In the medium term, it would be possible to make the desktop tool accessible via the mobile interface, allowing the user to review or refresh desktop-powered analysis while on site, or in meetings with city planners or politicians. Ideally, this would include production of visualisations and/or other means of cleanly displaying desktop-level results. Similarly, it would be possible to control the mobile tool from a desktop, meaning that once installed on a bus, the operator would never again need to manually start and stop the tool – this could be left to someone whose primary responsibility is transit planning. Towards this end, it would also be feasible to add a degree of automation to the app; that is, the app could automatically detect the route of interest, automatically determine when a trip begins and ends, etc.

Finally, as new sensors are added to mobile phones, features could be added to enable the app to count passengers; measure the indoor air quality in transit vehicles; automate payments of passengers without having to tap on/off; and perhaps improve customer comfort by reducing jerkiness of vehicle movement by providing feedback to bus drivers.

# References

United Nations. (2011). *World Urbanization Prospects, the 2011 Revision*. Retrieved October 20, 2014, from United Nations, Department of Economic and Social Affairs: http://esa.un.org/unup/CD-ROM/Urban-Rural-Population.htm

United Nations. (2012). *Probabilistic Population Projections based on the World Population Prospects: The 2012 Revision*. Retrieved October 20, 2014, from United Nations, Department of Economic and Social Affairs: http://esa.un.org/unpd/ppp/Figures-Output/Population/PPP\_Total-Population.htm