Crowdsourcing Based Train Tracking

Hewage G. H. R. G.
Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka
rajith.11@cse.mrt.ac.lk

***Abstract*—A considerable fraction from the general population use trains as their means of transportation in Sri Lanka [1]. Even though the railway system fulfils the transportation needs of a lot of people, being one of the cheapest means of travelling, it has its own disadvantages as well, such as the regular delays of trains. Most of the time, the passengers get to know about the delays of trains after they have reached the railway stations. An effort made to reduce the effect this particular issue is discussed through this paper. The proposed solution is a mobile application powered by crowdsourcing, which let the users of the application know where a particular train is at a particular moment. Thus by letting them know whether the train in which they are willing to travel is on schedule or not and arrange other means of transport as necessary.**

Keywords—crowdsourcing; train-tracking; client-server

I. Introduction

People chose the railway system as an affordable medium of transportation and they often suffer from the consequences which are occurred due to delays in trains [2]. Since trains are not as frequent as buses, the information about train delays, reaching the passengers when they are at the railway stations, would not be of much help. If they have a method to track the locations of the trains and check whether the trains are falling behind the usual schedules, they would not need to go the railway stations. This would save their time and avoid them from the frustration of having to wait at the railway station for hours.

In the solution that will be discussed throughout this paper, the passengers will be able to track the locations of the trains through a mobile application. The data about the trains' locations will be fed into the system by the users of the application. These could be the users waiting for a train at a railway station or the users who are travelling on a train. The information that will be gathered regarding a particular train is whether or not it has left a particular station. Based on the gathered input from users, another user can check which stations a required train has passed as of a particular moment.

Since this is an application which is based on crowdsourcing, the reliability of the information provided by the users and the quantity of the information received, have a high impact on the accuracy of the results provided by the application. If the users provided incorrect data, then the application would not show the required behavior. When the collected data is less, the confidence level of the results of the application would be less.

II. Literature Review

A. *Use of the application*

The public transportation services failing to provide a quality service affects the productivity of the individuals as well as the country as a whole, because the majority of the work force use public transportation to get to their working places. When they fail to arrive on time, it is the time that could have been used to do something worthwhile for the country that gets wasted. Even though a mobile application is not the optimum solution for this, it could give a temporary solution, which can be developed into a better one as time passes by.

Most of the time people travel several kilometers to reach their nearest railway station. And when they miss a train there or a train is significantly late, they may not be able to wait for another train or turn around and go back to a bus stand where they can reach their required destination because by the time another train comes or they reach a bus stand, it would be already too late. Hence it would be really profitable for a passenger to know, whether he could reach the railway station before the train leaves there, when he is at a place like a bus stand since that would give him an alternative way of transportation if he's to miss the train.

B. *Feasibility of the application*

Whenever we use public transportation, we can see that almost everyone is using a mobile phone and smartphones are not rare to find among them. Statistics show that the number of mobile internet users has been doubled almost each year from 2009 [3] (up to 2013). The recent data may suggest that it has increased even more [4]. Hence a mobile application can be seen as a solution with the potential to make an impact.

Another project which aims at tracking trains in real time is being under way which uses Global Positioning System (GPS) and it was put into operation in the coastal line of the railway network in June 2014 [5]. While the GPS method would not rely on the information of the public, it would have a higher accuracy whereas a mobile application of this sort would not require a capital as such.

There are other similar approaches taken for the problem that this paper tries to address. The Indian Train Locator [6] and Live Train Status [7] are two such mobile applications

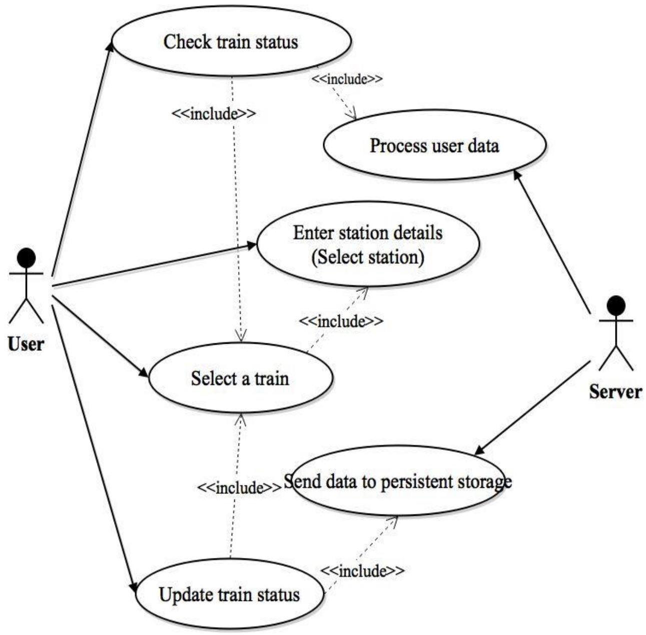

Fig. 1. Use case diagram.

which provide location information on trains in India. But there are no such mobile applications in the Sri Lankan context.

The feasibility of the project has to be considered on the angle whether crowdsourcing can be used in an application like this which is to be used by the crowd described earlier in this discussion. Crowdsourcing has become an emerging trend around the world and it has been widely used in areas like maps and traffic information collecting [8]. The twitter page @road_lk is one great example usages of crowdsourcing in Sri Lanka to gather traffic data [9]. This is an active community and it evidently shows that crowdsourcing can be successful in the local community.

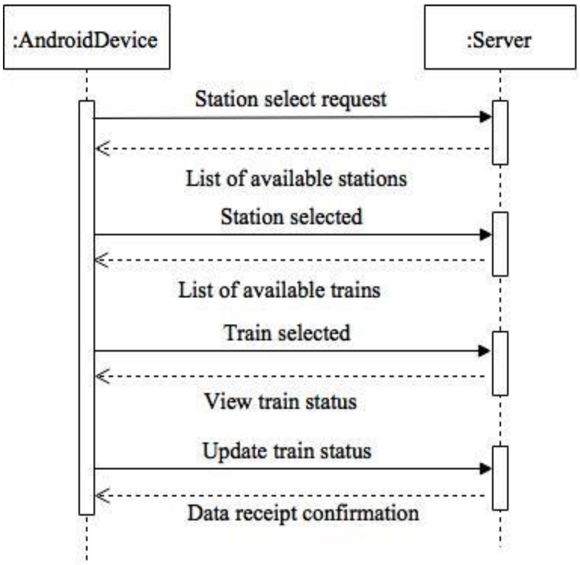

Fig. 2. Sequence diagram.

III. System Design

The system follows the client-server architecture. The users' mobile phones send and receive data to and from the server whereas the server processes the data gathered from the users and produces results to the client mobile phones. The overall architecture is described through the end of this section.

A. Use cases

The main two use cases that involves the user of the application are "Check train status" and "Update train status" (Fig. 1).

The "Update train status", briefly explained, is the situation where the crowdsourcing actually happens. Through an interface in the application, the users are given the ability to mark whether a given train has left a given station. (The station and the train selection happen as the user comes to the aforesaid interface.)

"Check train status" use case involves the part where the user gets the data about a particular train. From the data gathered from the users, the server end process tries to come to a conclusion whether a particular train has actually left the station which is mentioned by the user or not. Given that the server can provide a confident answer, it will send it back to the user and if not, it will respond saying that an answer is not capable of being given.

B. Logical View

The sequence diagram in Fig. 2 would give a better understanding on the above mentioned use cases. Here the "Update train status" and the "Check train status" are combined since they can be accessed in the given sequence in the diagram.

Fig. 3 illustrates the logical view of the system: the basic components of each client-side and the server-side of the system. The server side functionalities are handled incorporated with the Google App Engine Application Programming Interfaces (APIs).

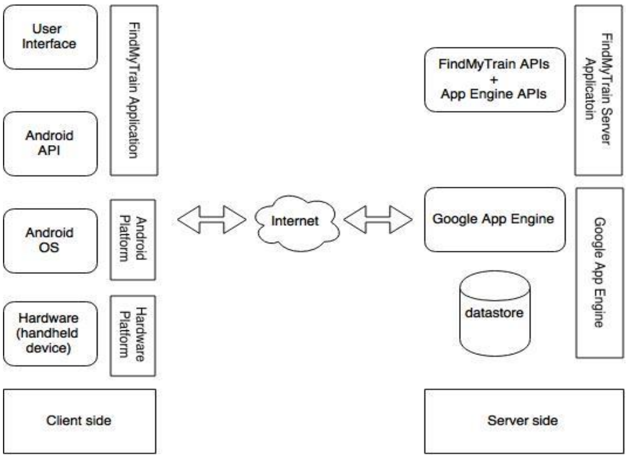

Fig. 3. Logical view.

IV. Implementation Details

A. Communication between the server and the application

The server application of this system is run on Google App Engine [10]. It has a set of API methods which gives the client application the ability to communicate with its data store. The data are stored as Java Document Object (JDO) s and the definitions of these objects are defined in the entity classes. When the entity classes are defined, the necessary APIs that can be used to store, retrieve, update and delete can be automatically generated. If needed, these and any other custom APIs that are needed, can be created manually as well.

B. Processing the gathered data

The data collected by the users are stored as Record objects. Once a particular station and a train has been selected, there is a set of related Record objects. Those Record objects contain information about the server timestamp, status of the train (whether it has left or not) among other details. Once we have filtered the records list, these two are going to be the factors that are used to determine the state of a train. (The "left" status of a train is denoted by '1' in a record whereas the "not left" status is denoted as '0'.) The status of the train will be then displayed with respect to a number of nearby stations from the user's station.

The first part of the analysis is straightforward. If too many people say that the train has left, then we can come into a conclusion that the train has actually "left" with a high confidence.

This "too many people" value is measured by a threshold value (which will be referred hereafter as "the upper limit") and it can be changed appropriately based on the usage statistics of the application. In the same way, if too little people say it has left (another threshold could be selected to determine the "too little people" which will be referred as "the lower limit") then the output would be that it has "not left".

The two threshold values, the upper limit and the lower limit will be chosen with a gap in between. This refers to the situation where the train status attribute of the records being somewhat equally distributed with '0's and '1's. Without proper reasoning this situation cannot be lead to a conclusion about the train's status. The following process handles these kind of situations.

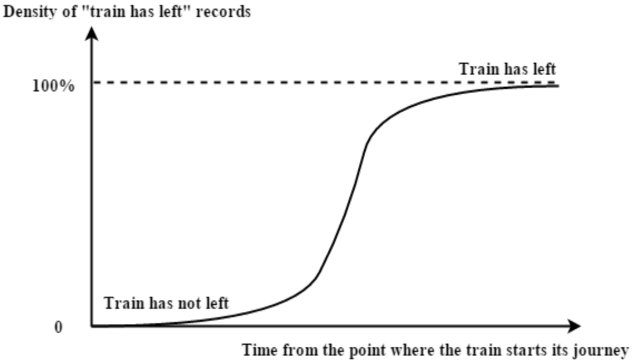

Fig. 4. The graph of the density of "train has left" records with respect to a particular railway station versus the time elapsed since the train has started its journey.

First step in this process is to sort the record objects based on the timestamp. After sorting, a stream of data from the statuses in the record objects can be extracted. This can be modelled into a discrete signal where the function value will be either '1' or '0'. This data stream (the '1's and '0's stream in the record objects) higher the number of transitions get mean that the accuracy of the received data is less. (A transition is the data stream being changing from '1' to '0' or '0' to '1'.) A higher number of transitions mean that within a small time period, the records received from the users change. This means that the user records are not consistent. Hence if the transition proportion exceeds a particular threshold, the produced output will be "maybe" since the states "left" or "not left" cannot be produced with certainty.

In the case where the number of transitions are much less, (which means that '1's and '0's are concentrated separately rather than being distributed through the stream) we can say, based on the pattern, whether the train has left. That is when the '0's are concentrated towards the beginning of the stream and the '1's are concentrated towards the end.

The graph in Fig. 4 represents how the data stream would be, in the above mentioned situation. (In perfect case scenario, i.e. no erroneous data are present, the inclined part of the graph would be replaced by a vertical line.) Therefore, if the data stream has more '0's in the front, less transitions and again more '1's in towards the end, that would mean that the train has passed that station. And whether the data stream has too many transitions is checked using the following formula.

$$f_1\,[n_i] = (f\,[n_{i+1}] - f\,[n_i]) \,/\, (n_{i+1} - n_i) \qquad (1)$$

Equation (1) is defined on a discrete valued domain and it is having n as the independent variable. It is the index of a record object which is in the sorted order based on the timestamp. The function f represents the status of the train of a given record object (which will be identified by n). The value $n_{i+1} - n_i$ will be 1 since consecutive record objects are used in (1). The function f_1 produces a non-zero value when the considered record objects have different values as the train status.

The input data stream is considered as a discrete digital signal when (1) is used on it. (The y-axis being the status of the train, '1' or '0' and the x-axis being the record number sorted according to the timestamp.) With this modelling, removing erroneous data could be seen as removing noise from the signal. Even though the formula is not the best way to remove noise, for this application, its performance was enough as it could be used to find the number of transitions in the data stream.

As an example use of the method used in the application, when the data stream is represented by the following stream of '1's and '0's: "0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1" the formula returns the following output: "0, 0, 0, 0, 1, -1, 1, -1, 1, 0, 0, 0" in which a '1', '-1' pair or a '-1', '1' pair corresponds to a transition in the input data stream.

As of now, the result provided by the server-end about a train with respect to a station could have three states; “left”, “not left” and “maybe”. Another possible result would be the “not sure” state. This state will be returned when the data that have been gathered is not enough (a case where the number of records collected would be less than a selected threshold).

V. Conclusion

A. Problems faced and potential solutions

One of the main issues is where the status of the trains with respect to stations being in the “not sure” state all the time. As it was explained in the previous section, this happens due to lack of information gathered from the users, which is highly likely the case at the early stages of the product launch. To build the necessary user base, promotions for the application can be targeted on selected groups of people (frequent train passengers whom is in touch with the technology) whom can be attracted using social media (groups of people gathered around pages and groups such as Train Simulator Sri Lanka [11] , Train LIFE [12] , Sri Lanka Railway Forum [13] etc.)

The other major issue would be to get people engage in this application. Since the application is powered by crowdsourcing, it is required for the users to “actively” participate in using the application rather than only using the application to view details of trains, the user should also support it by updating the data in the system with their inputs. For that, a level system for the users, when they tend to do update more information, their user profile’s level would get higher can be included in the system.

B. Future improvements

1) Using GPS to verify the location of a user.

The gathering of data from the users is done according to the sequence diagram in Fig 2. Here the stations selection is done manually (the user will select the station and there is no way of verifying whether the user is actually at that railway station when he does an update). The accuracy of this method can be improved by using GPS to verify the location of the user.

2) Web application for data entering

At the moment, data about the trains (the stations on which they stop and the respective times) are entered to the system using the default API provided by the Google App Engine. With a user friendly web application interface, the data entering process could be made easier and it can be given to the railway department personnel to be used.

3) Record considering time gaps based on the train type

When the records are used for processing to determine the status of a train with respect to stations only the records within a certain time gap is considered (that is to get the most recent records for the processing). In the current system, regardless of the train type whether it is express or slow the time gap remains the same. When observed carefully, it can be seen that based on the type of the train, this time gap should be changed. Slow trains could have a smaller time gap since they tend to stop at almost all the stations in the line they are travelling and it does not take much time to go from one station to the next one and express trains should have a larger time gap since once they stop at a station, the next stop could be after 10-15 stations in certain cases and it takes some time to get there.

4) Language support for users

The main operation language of the app is United State English at the moment. But since the application is intended to be used by the general public of Sri Lanka, it would be better if the application can provide support to Sinhala and Tamil as well.

References

[1] Department of Cencus and Statistics – Sri Lanka. “Statistical Pocket Book 2015.” Available: http://www.statistics.gov.lk/Pocket%20Book/chap08.pdf

[2] News 1st. (2015, Sep, 7) *Train Delay Archives* [Online]. Available: http://newsfirst.lk/english/tag/train-delay [Accessed: 10-Sep-2015].

[3] GSMA Intelligence. Country overview: Sri Lanka, 2013. [Online]. Available: https://gsmaintelligence.com/files/analysis/?file=131003-sri-lanka.pdf [Accessed: 20-Sep-2014].

[4] Lanka Business Online. (2015, May, 7) *Sri Lanka’s mobile internet usage grows 85.8-pct in 2014: CB* [Online]. Available: http://www.lankabusinessonline.com/sri-lankas-mobile-internet-usage-grows-85-8-pct-in-2014-cb/ [Accessed: 10-Sep-2015].

[5] Sri Lanka Railways, [Online]. Available: http://www.slrail.info/ [Accessed: 20-Sep-2014].

[6] “Indian Train Locator”. [Online]. Available: https://play.google.com/store/apps/details?id=com.logiclabz.indiantrainlocator&hl=en [Accessed: 11-Sep-2015]

[7] “Live Train Status”. [Online]. Available: https://play.google.com/store/apps/details?id=com.LiveIndianTrainStatus [Accessed: 11-Sep-2015]

[8] S. Kessler, (2010, May, 26). “5 Creative Uses for Crowdsourcing,” Mashable. [Online]. Available: http://mashable.com/2010/05/26/creative-crowdsourcing/ [Accessed: 20-Sep-2014].

[9] “road.lk”. [Online]. Available: https://twitter.com/road_lk [Accessed: 20-Sep-2014].

[10] “What Is Google App Engine?”, (2015, Sep, 9). [Online]. Available: https://cloud.google.com/appengine/docs/whatisgoogleappengine [Accessed: 10-Sep-2015]

[11] “Train Simulator Sri Lanka”. [Online]. Available: https://www.facebook.com/trainsimulatorsrilanka [Accessed: 20-Sep-2014].

[12] “Train LIFE”. [Online]. Available: https://www.facebook.com/trainlife9 [Accessed: 20-Sep-2014].

[13] “SriLankaRailwayForum”. [Online]. Available: https://twitter.com/SLRailwayForum [Accessed:20-Sep-2014]