



The Transit Rider's Guide to the Galaxy

**Submission for
Google Android Developer Challenge (ADC) I
by Todd Lee**

Introduction	3
System Overview	3
Application Overview	3
Transit Service Provider Selection	4
Service Schedule Details	6
Routes	6
Stops	6
Schedule	6
Additional Features	7
Map Stop	7
Set Alarm	7
Technical Details	8
Known Issues	8
Future Work	9
Project Motivation	9

Introduction

Tranzit is a mobile application designed to make existing data from Google Transit public data sources accessible in a convenient form for mobile users. Built on the Android platform, *Tranzit* takes advantage of a number of the features available in Android such as the location based API, integrated mapping capabilities, and real-time system notifications to provide a comprehensive mobile experience.

System Overview

Given that the existing Google Transit data for a given transit provider is only available in zip archive format, the mobile client is paired with a PHP/MySQL backend web service, providing access to mirrored transit data in a format more appropriate for mobile use. A high level system diagram is shown in Fig. 1.

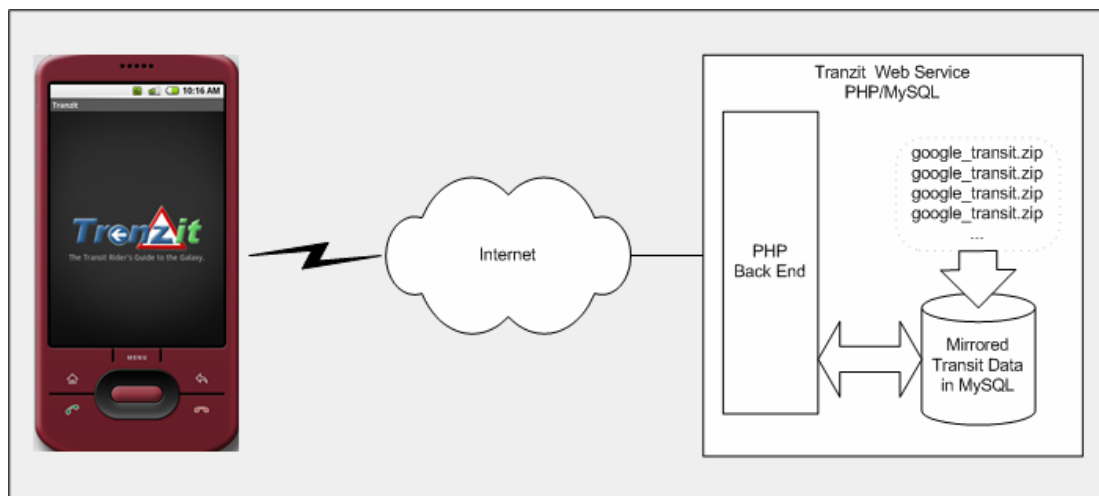


Figure 1 Client-Server System Overview for Tranzit Application

Application Overview

In the interest of providing an efficient usage model, the typical usage scenarios are optimized for mobile use, requiring minimal input/user intervention. The application's interface is designed with the mobile user in mind. The main application screen (Fig. 2), contains a simple options menu, providing access to the available features.

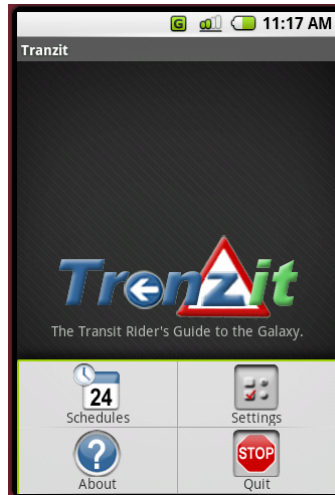


Figure 2 Application with Options Menu Displayed

The intuitive process used to find the desired transit information can be broken down into three simple steps:

- Select appropriate Transit Service Provider
- Select desired Route
- Select desired Stop

A full service schedule for the current day based on the user's selections will be displayed. Two additional features allow users to further interact with their selected service schedule. Using the options provided, users can:

- Create a notification to alert them of the arrival of the selected Route at the selected Stop at a given time.
- Display a Map View showing the current Transit Stop selection.

These usage scenarios and the supported features are explained in more detail below.

Transit Service Provider Selection

Naturally, given the scope of the application (serving a significant subset of available public data from the Google Transit) the first step is to select the appropriate service provider for the desired location. The application provides preference settings (Fig. 3) which enable automatic determination of the most appropriate Service Provider based on the current position of the device (determined by the Location Manager¹).

¹ Note: This feature requires the 'gps' provider from the Android Location API

If the use of the Location Manager is enabled (the checkbox is checked), *Tranzit* will use the range provided in the spinner control to determine the most suitable location/provider relative to the current position².

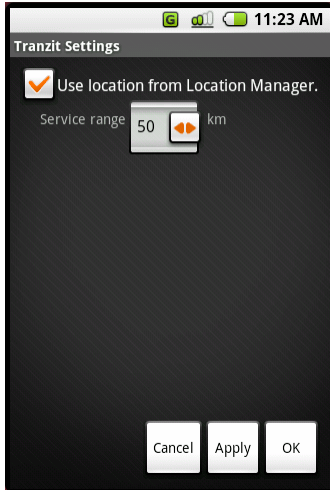


Figure 3 Pref. Settings

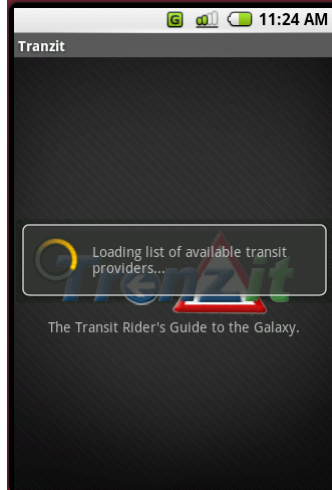


Figure 4 Location Query

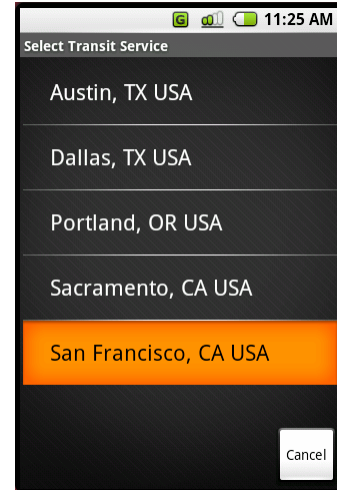


Figure 5 Location List

When the application is initially opened with the Location Manager enabled, available locations will automatically be queried from the *Tranzit* service. If the Location Manager is not enabled, or the user simply wants to re-query the locations manually, they can select the **Schedules** option (Fig. 2) from the application menu. A progress dialog is displayed while this operation runs in the background (Fig. 4). The successful location query will result in one of two outcomes:

- If the Location Manager is enabled, and a single Transit Service Provider is available within the given location criteria, then that Provider will be selected and the user will be forwarded directly to the Service Scheduling Activity where they can interact with the provider's service schedule.
- If the Location Manager is *not* enabled, or if there is more than one potential provider for the given location criteria, then the user will be presented with a list of available providers (Fig. 5) (in the case of the disabled Location Manager, this is *all* providers) from which they can select.

Either way, once a Transit Service Provider/Location has been selected, the user will be directed to the Service Details where they can interact with the provider's service schedule.

² Note that this feature is enabled by default.

Service Schedule Details

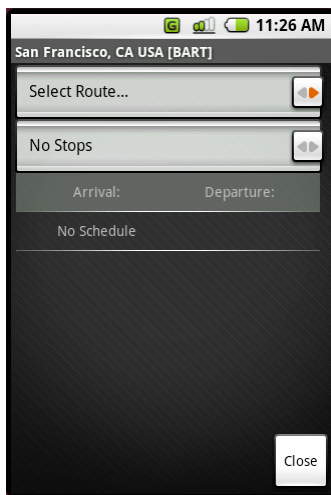


Figure 6 Service Details

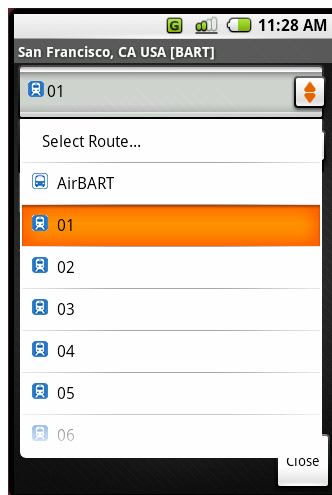


Figure 7 Select Route

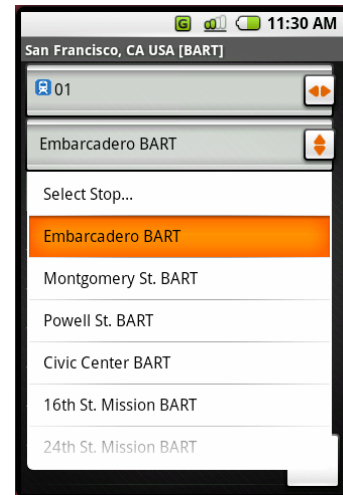


Figure 8 Select Stop

The *Service Details Activity* (Fig. 6) provides access to the transit provider's service information for the current day. The activity is composed to three primary controls used to access this information.

Routes

Routes are selected from the **Routes** spinner (Fig. 7). This list is automatically loaded with all of the transit routes in operation by the selected service provider. The list displays the name/id of the transit route along with an icon³ denoting the type of transit service available. Users can start by selecting the desired route from the available list.

Stops

Stops are selected from the **Stops** spinner (Fig. 8). This list is automatically loaded with the names of the available stops, based on the selection in the **Routes** spinner. This list is updated whenever a new route is selected.

Schedule

Once both a route and a stop have been specified, the **Schedule** List (Fig. 9) is updated automatically with the available service for the current day (note some routes don't provide service on certain days). The schedule is displayed with both arrivals and departures for the selected route at the specified stop.

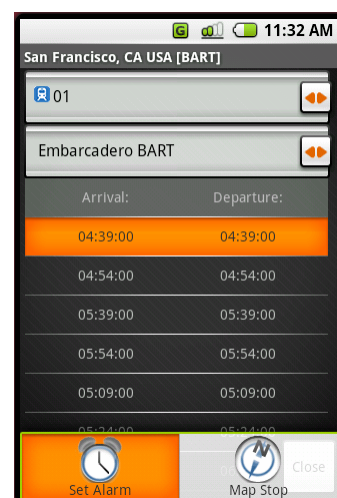


Figure 9 Service Details

³ Icons from Google Transit

Additional Features

The options menu in the *Service Details Activity* (Fig. 9) provides some additional features with which users can interact with the provider's service schedule.

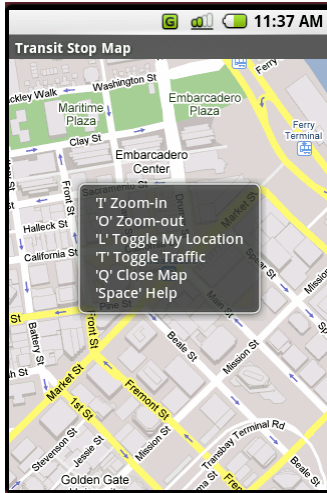


Figure 10 Mapped Stop

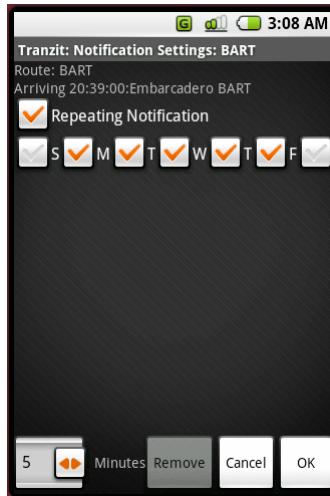


Figure 11 Setting Alarm



Figure 12 New Notification

Map Stop

If a stop is selected in the **Stops** spinner (Fig. 8), users can select the **Map Stop** option in the menu to open a map view centered on the coordinates of the selected stop (Fig. 10). This map provides basic functionality including a *help* dialog which can be accessed by pressing the *space* key on the device.

Set Alarm

If a specific schedule item (arrival/departure time) is selected (Fig. 9), users can select the **Set Alarm** option in the menu to create a new notification that will alert them prior to the arrival of the selected route at the specified stop.

New notifications are created using the *Notification Settings Activity* (Fig. 11), as either *repeating* or *one-shot*. Repeating notifications can be specified to occur on particular days to coincide with the desired schedule. The notice period determines the length of time prior to the scheduled arrival of the selected service that the user is to be alerted. The default is 5 minutes. Notifications are based on the real time clock and will wake the device if required.

The notification message (Fig. 12) displays the Route and Stop for the specified service as well as the scheduled time of arrival. Selecting the *Notification Details* will open the notification in the

Notification Settings Activity. Here, the user can simply acknowledge/update alerts (by clicking *OK*) or in the case of repeating notifications, choose to remove further alerts.

Note that in the case of repeating notifications, only the *next* notification is set at any given time. Subsequent notifications are not created until the current notification is acknowledged. If a given notification is not acknowledged, future notifications will not be set.

Technical Details

Tranzit was developed using the Android SDK version ***m5-rc15***. The application requires a network connection as it makes http requests to a web service to obtain information required for normal operation.

The following is just a few of the platform features employed by the *Tranzit* application:

- GPS location based services to determine current position and heuristically select the most appropriate service provider from all available locations.
- Remote Service (with custom aids) to manage background loading of requests for location information and updating available service providers.
- Activities (and plenty of them) to handle the main aspects of the application
- Integrated Map View to display transit stop coordinates.
- System Notifications to create RTC based alerts for service schedules.
- Intent Receiver to listen for system notifications and respond.
- SQLite Database to cache service provider location information.

Known Issues

- SQLite Database can occasionally lock up causing the application to fail when querying available locations. Presumably due to android issue #151. Prescribed work around generally seems to work.
- Routes/Stops spinners fire selection events on every selection (see android issues #607/#640) rather than just 'final' selection. Quick browsing in the dropdown lists can cause application errors.
- GPS location provider occasionally causes spurious errors (android issue #70). Sometimes it will report erroneous location

or not show up at all. If this happens, users must disable the Location Manager based *auto locate* feature⁴.

Future Work

- Expand back end service to go beyond 'proof of concept' stage.
- Lobby for possible evolution of Google Transit data into exposed service with complete API, rather than current mirroring.
- In the mean time, complete mirroring of ALL currently available Google Transit public data.
- Integrate trip planning heuristics.
- Improve Map integration for route mapping and overlay support.
- Improve Schedule/Notification interaction to synchronize notification enablement with actual service availability.
- Expand queries to include 'nearest stop' option.
- Query optimizations to enable even faster response time.
- Enable caching optimizations for common requests or specific location based information.
- UI optimizations corresponding to SDK updates.
- UI customizations with application of themes/style etc.
- Full project web site.

Project Motivation

Given that interest in sustainable living and eco-friendly practices are at an all time high, the motivation behind this application is to provide a means to encourage the use of public transit as a viable alternative to less sustainable modes of transport.

While many transit service providers are getting on the bandwagon with providing easy to use web-based trip planning software, there are few if any viable mobile alternatives that provide the same level of access.

Ultimately, *Tranzit* can serve not only as a tool for individuals for personal use, but as a *value add* to the service providers themselves who can in turn offer it to their ridership as a tool to promote their service.

⁴ Note for the purpose of the ADC judging, a static location of Mountain View, CA has been hard coded into the application.