STOL Crossing Request

# Saxton Transportation Operations Laboratory Crossing Request

## Tab 1: Summary

STOL Crossing Request is a pair of software components (Android application and server) that enables pedestrian to driver communication in the potentially hazardous environment of mid-block crossings. The smartphone app enables pedestrians to send alerts to approaching drivers to make them aware of their intent to cross the street. This software was used to conduct a human-factors study on driver awareness of pedestrians at mid-block crossings at the Turner-Fairbanks Highway Research Center.

## Tab 2: Description

STOL Crossing Request has the following functionalities:

* Android application:
  + Log user location via GPS for experimental analysis
  + Operates in either pedestrian or driver mode:
    - Pedestrian:
      * Display configured mid-block crossing geofence using Google Maps API
      * Once inside the geofence, the pedestrian can request the server to broadcast an alert to driver devices
      * Whether an alert is currently active or not the pedestrian UI will inform the user of the current alert status
      * The pedestrian device can command the server to create a new log file for easier data collection and analysis
    - Driver:
      * Displays a navigation system-like user interface
      * Displays a visual pop-down alert when an alert is received from the server
      * Plays an audio alert when an alert is received from the server
      * Only alerts the driver if they are within the bounds of the configured geofence and their heading is within the configured range.
* Server:
  + Receives alert requests from the pedestrian device and broadcasts out to the driver devices for 30 seconds
  + Provides the needed geofence data for the pedestrian and driver modes of the Android application
  + Stores and manages log files containing research data (latitude, longitude, timestamp, network latency, speed, heading, etc.) from all devices
  + Logs events such as devices entering geofences, leaving geofences, starting requests, registering and unregistering with the server, and receiving alert broadcasts

## Tab 2: Release Notes:

### License

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This software makes use of various third-party software components that have reuse licenses associated with them. These packages and their license information are shown below.

1. SPRING:

* Terms of use are provided at https://pivotal.io/terms-of-use

### Installation and Removal Instructions

* The Android application may be compiled into a .apk file for installation on a device through the Android Studio user interface. The default Android Studio build process will generate an unsigned-APK file but many devices require signed-APKs for successful installation. Instructions can be found here (<https://developer.android.com/studio/publish/app-signing.html>) for how to produce this signed APK. Installation can then be completed by connecting the device via USB for file-sharing and copying the signed-APK onto the device. The device must be configured to support side-loading of applications. Once it is copied, simply click on the copied file and select to install it.
* The server software may be compiled into a single JAR file using the “gradle bootRepackage” command in the root of the server project. This jar may be installed into a cloud hosting environment such as Amazon Web Services. The “data” folder and the geofences inside must also be copied into the same directory as the JAR file. A “datalogs” folder with appropriate permissions must exist in the same folder as the JAR file as well. The server’s firewall should be configured to allow incoming requests and outbound responses on port 5000
* Software removal – To remove the android application simply uninstall it via the operating system’s application manager. To remove the server software, simply delete the JAR file, the “data” folder, and the “datalogs” folder.

### Operating requirements

* Android device:
  + Android OS version 6.0 or higher
* Server:
  + At least equivalent to Amazon Web Services Nano-tier
* Connectivity: Cellular or Wi-Fi
* Operating systems supported: Linux and Android

### Related web sites

The software is distributed through the USDOT's JPO Open Source Application Development Portal (OSADP), <http://itsforge.net/>

## Tab 3: Documentation

* STOL Crossing Request User’s Manual version 1.0.0

## Tab 4: Discussion

* Main discussion (link)
* Issue discussion (link)

## Tab 5: Related Applications

* List of all applications with same categorization