Smart Work Zone Activity App (SWiZAPP)

Final Report and User Manual
June 2019

SWZDI
Smart Work Zone Deployment Initiative

Sponsored by
Smart Work Zone Deployment Initiative
(TPF-5(295))
Federal Highway Administration
(Part of InTrans Project 19-535)

The preparation of this report was financed in part through funds provided by the Iowa Department of Transportation through its "Second Revised Agreement for the Management of Research Conducted by Iowa State University for the Iowa Department of Transportation" and its amendments.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
InTrans Project 19-535			
4. Title Smart Work Zone Activity App (SWiZAPP)		**5. Report Date** June 2019	
		6. Performing Organization Code	
7. Author(s) Yaw Adu-Gyamfi, Carlos Sun, Praveen Edara, and Farzaneh Azadi		**8. Performing Organization Report No.** InTrans Project 19-535	
9. Performing Organization Name and Address University of Missouri-Columbia Department of Civil & Environmental Engineering E 2509 Lafferre Hall Columbia, MO 65211		**10. Work Unit No. (TRAIS)**	
		11. Contract or Grant No.	
12. Sponsoring Organization Name and Address Smart Work Zone Deployment Initiative Iowa Department of Transportation 800 Lincoln Way Ames, Iowa 50010	Federal Highway Administration U.S. Department of Transportation 1200 New Jersey Avenue SE Washington, DC 20590	**13. Type of Report and Period Covered** Final Report and User Manual	
		14. Sponsoring Agency Code TPF-5(295) Federal SPR Part II, CFDA 20.205	
15. Supplementary Notes Visit https://swzdi.intrans.iastate.edu/ for color pdfs of this and other Smart Work Zone Deployment Initiative research reports.			

16. Abstract

The objective of this research project was to design, develop, and deploy the Smart Work Zone Activity app (SWiZAPP), which is a cross-platform mobile application for collecting, reporting, and posting accurate, real-time, work-zone activity information and status. The app is enabled with functionalities for managing an unlimited number of construction work zones due to its scalable, cloud-based design architecture. It supports work-zone geolocation and mapping via on-board GPS sensors and Google Maps, respectively. Users of the app can post live activities from construction sites by taking snapshots and uploading images, utilizing buttons within the app's interface to indicate traffic conditions and lane activities, or text messaging via the app. SWiZAPP also enables its users to view both real-time and historical activities of all work zones in the SWZDI states.

This document includes a user manual, as an appendix, to access and use SWiZAPP for work-zone activity monitoring. The user-friendly interface includes standard work-zone procedures and is suitable for use by both department of transportation (DOT) staff and contractors.

The benefits from the successful deployment of SWiZAPP include more accurate and timely work-zone information for work-zone management, traveler information, inspections, contract monitoring, safety analysis, and project coordination.

17. Key Words		18. Distribution Statement	
data collection—mobile applications—traveler information and communication systems—work zones		No restrictions.	
19. Security Classification (of this report)	**20. Security Classification (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified.	Unclassified.	56	NA

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

SMART WORK ZONE ACTIVITY APP (SWIZAPP)

Final Report and User Manual
June 2019

Principal Investigator
Yaw Adu-Gyamfi, Assistant Professor
University of Missouri-Columbia

Co-Principal Investigators
Carlos Sun, Professor
Praveen Edara, Professor
University of Missouri-Columbia

Research Assistant
Farzaneh Azadi

Authors
Yaw Adu-Gyamfi, Carlos Sun, Praveen Edara, and Farzaneh Azadi

Sponsored by the Smart Work Zone Deployment Initiative and
the Federal Highway Administration (FHWA) Pooled Fund Study TPF-5(295):
Iowa (lead state), Illinois, Kansas, Missouri, Nebraska, Wisconsin

Preparation of this report was financed in part
through funds provided by the Iowa Department of Transportation
through its Research Management Agreement
with the Institute for Transportation
(InTrans Project 06-277)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This research was conducted under the Smart Work Zone Deployment Initiative (SWZDI) and Federal Highway Administration (FHWA) Pooled Fund Study TPF-5(295) with the following state departments of transportation (DOTs):

- Illinois
- Iowa (lead state)
- Kansas
- Missouri
- Nebraska
- Wisconsin

The authors would like to thank the FHWA, the Iowa DOT, which also used Federal state planning and research (SPR) Part II, Catalog of Federal Domestic Assistance (CFDA) 20.205 Highway Planning and Construction funding for this publication, and the other pooled fund state partners for their financial support and technical assistance.

EXECUTIVE SUMMARY

Background

Accurate and timely communication of work-zone activities improves work-zone safety by alerting department of transportation (DOT) staff, traffic management centers (TMCs), contractors, and the traveling public that a work zone has become active or inactive. Such information also facilitates on-going work-zone safety analysis by enabling the synchronization of work-zone and incident data. With the increase in computing power of portable electronic devices, such as smartphones, smart work-zone information exchange can be accomplished simply and cost-effectively.

Objectives

The objective of this research project was to design, develop, and deploy the Smart Work Zone Activity app (SWiZAPP), a cross-platform mobile application for collecting and reporting real-time work-zone activity information. This included development of requirement specifications for the app, prototype design, and field testing.

Prototype design was central to this project, as the goal of this project was to provide a tangible software deliverable that is close to being market ready. The field testing was important for demonstrating the app in a live and demanding environment.

Methodology

SWiZAPP was developed using React Native, currently the most popular open-source, mobile application development framework. The research team followed a modular design approach to enable seamless future expansion of the app by other agencies. SWiZAPP consists of three main modules, as shown in Figure ES1: frontend, backend, and middleware.

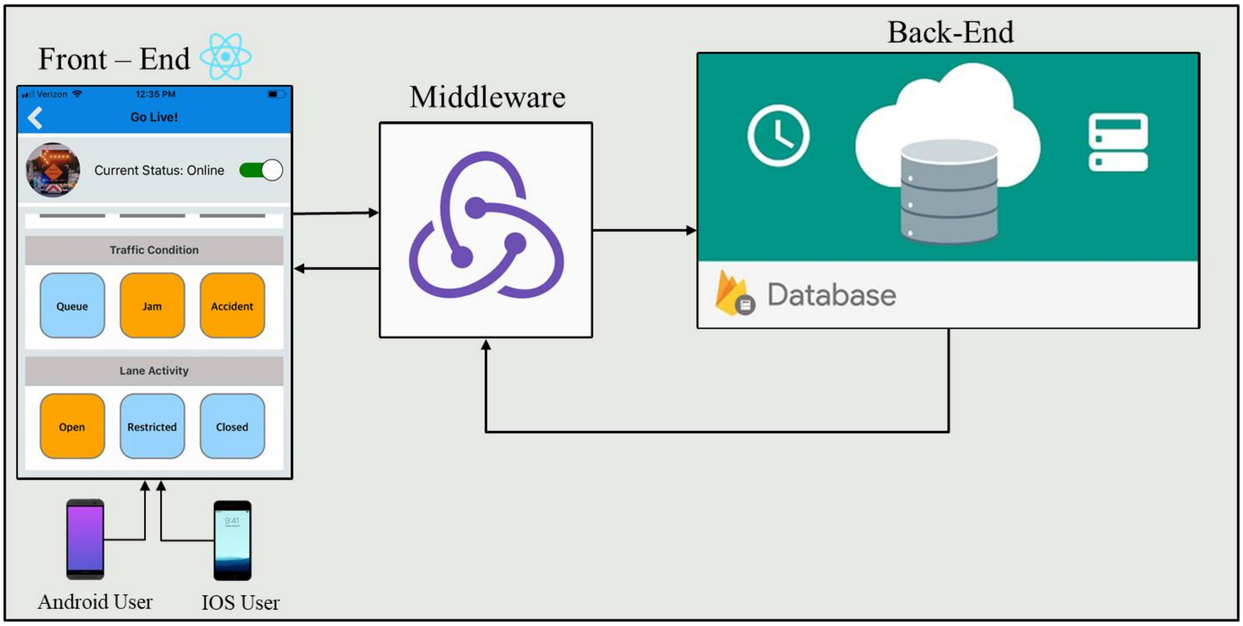


Figure ES1. Frontend, middleware, and backend of SWiZAPP

The frontend module allows users to connect and interact with the app via a graphical user interface (GUI). A variety of layouts and user interface elements, including flat lists, grid and scroll views, buttons, switches, dialog boxes, and notifications, were used to improve the user friendliness of the frontend module.

The backend module is responsible for storing all work-zone activity information posted by users in a scalable manner. Google's Firebase engine was used to design and deploy a real-time, cloud-based database to support the backend.

The middleware module is managed by Redux, which translates user actions from the frontend to the backend and vice-versa.

Key Outcomes

The outcome of this project is a fully-functioning mobile application for work-zone activity monitoring. The app is enabled with functionalities for managing an unlimited number of construction work zones.

SWiZAPP currently supports automatic work-zone geolocation and mapping via on-board global positioning system (GPS) sensors and Google Maps, respectively. Users of the app can post live activities from construction sites by taking snapshots and uploading images, utilizing buttons within the app's interface to indicate traffic conditions and lane activities, and/or text messaging via the app. The app also enables users to view both real-time and historical activities of all work zones in Smart Work Zone Deployment Initiative (SWZDI) states. The key components of SWiZAPP are shown in Figure ES2.

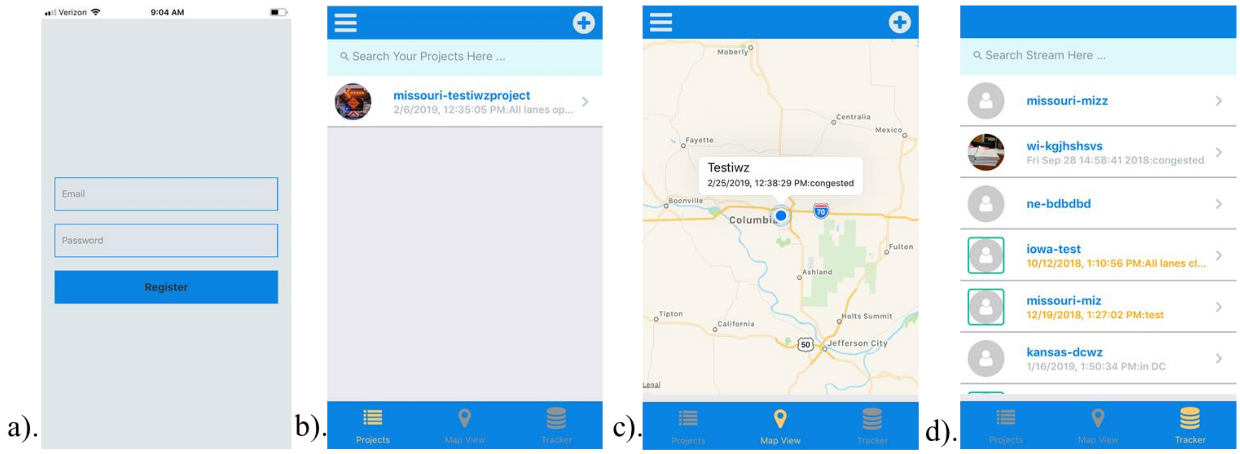


Figure ES2. Key components of SWiZAPP: (a) log-in, (b) Projects, (c) Map View, and (d) Tracker

A summary of SWiZAPP's use cases is provided in Table ES1.

Table ES1. SWiZAPP use case summary

Applications	Use cases	Users
Alert Management System	Sends real-time alerts during lane closures and crashes and when a new work zone is activated	1. Traffic management centers 2. 511 3. Third-party service providers such as Waze, Google, etc. 4. Possible integration with CAVs
Information Management and Tracking	Work zone data collection, tracking, and archiving	1. Contractors 2. Work zone managers 3. Road users

The app could serve as a work-zone-alert management system for TMCs or third-party service providers, such as Waze, Google, and INRIX. Lane closures, crashes, and new work-zone location information will be communicated to users in real time. Contractors and work-zone managers could also use the app for data collection, tracking, and archiving purposes. Application programming interfaces (APIs) could be provided for open data platforms to utilize stored work-zone data.

Recommendations and Limitations

One of the key bottlenecks of SWiZAPP is its reliance on internet access. A sizeable number of construction work zones in many states are in dead zones where internet access is restricted. SWiZAPP will not function in such environments.

Ideally, the app could be re-designed to store work-zone activity information on the phone's storage system when there is no internet access. The locally stored data could later be pushed to the frontend and backend after the user re-enters an area with Wi-Fi access.

We also recommend that concerted efforts be made to integrate SWiZAPP with key transportation data management systems, traveler information systems (e.g., 511), and other transportation data archival systems.

CHAPTER 1. INTRODUCTION

Background

Work-zone environments can be very dynamic in nature, depending on the characteristics of a particular site. Work intensity, lane closure (if any), traffic demand, and time of day are just some of the factors that affect the safety and efficiency of facilities with work zones. Under certain circumstances, work-zone sites can change rapidly once traffic control is set up and work has begun.

For example, if a lane is closed, capacity is restricted and traffic congestion can develop quickly, leading to queues. This is especially true in urban environments with high demand where a queue can grow several miles within minutes (King et al. 2004). Therefore, timely information about work zones is needed to properly document and describe work-zone operations on a particular day.

In addition, work-zone schedules, even week-ahead schedules, are constantly being adjusted due to staff/contractor resource allocation, work progress, weather, and other circumstances. What is planned in advance is often not realized in the same way. Finding an easy way to track the dynamic start and end times of work-zone activities helps with both the real-time and the long-term management of work zones.

Project Objectives

The objective of this research project was to design, develop, and deploy the Smart Work Zone Activity app (SWiZAPP), a cross-platform mobile application for collecting and reporting real-time work-zone activity information. This included development of requirement specifications for the app, prototype design, and field testing.

The requirements needed to be carefully specified because the app is intended to be used by multiple state departments of transportation (DOTs), and DOTs could differ in their desired work-zone information content, format, and delivery medium.

Prototype design was central to this project, as the goal of this project was to provide a tangible software deliverable that is close to being market ready. Prototype design was accomplished through hardware platform specification, software interface design, software sub-component design, and backend service development.

Field testing was important to demonstrate the app in a live and demanding environment.

Benefits and Opportunities

One benefit of using SWiZAPP is for work-zone management and operations. The accuracy of the information stems from the fact that the information is originating from a staff member who is at the actual work-zone site and has first-hand knowledge of the work-zone conditions. The information is broadcast instantaneously to multiple parties that have an interest in the information, including the DOT area office, project inspectors, traffic management centers (TMCs), contractor’s offices, and various media venues, if appropriate. This broadcast also results in more accurate and timely information being passed to third parties, such as travelers who use this information to plan trips or to navigate through the work-zone areas.

From the project management and contracting perspectives, SWiZAPP documents the times when the work zone is active. This is especially important for certain types of projects or contracts involving incentives/disincentives and lane rentals used in high-impact work zones.

SWiZAPP can also help improve coordination among different parties or even within organizations. For example, contractors can better coordinate with DOT staff and other contractors using the app.

After the SWiZAPP data are downloaded and archived, they become a valuable source of dynamic work-zone information that is currently lacking. For example, work-zone safety analysis, such as work-zone risk modeling, requires the integration of work-zone information with crash information. However, this integration is often difficult because dynamic work-zone status is not available (Sun et al. 2014). And, despite efforts in improving crash reporting, the work-zone field is often marked incorrectly in crash reports, as the reporting officer lacks knowledge of downstream work zones. Therefore, SWiZAPP provides both real-time and long-term benefits.

Modern smartphones are remarkable in terms of their portability and their enormous capabilities. The miniaturization of electronics has enabled the integration of multiple components into a single device instead of needing multiple devices that consume space and power. Even more impressive is the fact that the integration is also logical, meaning that information can be shared by all components via the software kernel.

The dizzying array of technology and sensors available on smartphones include high-resolution displays, touchscreens, microprocessors, accelerometers, gyroscopes, global positioning systems (GPS), cameras, infrared detectors, environmental sensors, switches, radio-frequency identification (RFID), Bluetooth, Wi-Fi, and, of course, cellular voice and data communications (Fedkiw 2012). By further processing raw sensor data, many more types of information, such as heart rate, quick response (QR) codes, and fingerprints can also be recognized by smartphones.

Even though electronic data systems have always been part of intelligent transportation systems (ITS), they are sometimes less visible since they function in the background of many ITS applications. The Missouri DOT (MoDOT), for example, has been using e-alerts to communicate work-zone and incident information via messaging and email to multiple parties. Similar to e-alerts, SWiZAPP will communicate dynamic work-zone information, but with more detail and more timely information.

This project focused on using ITS to communicate valuable work-zone information to ITS TMCs, traveler information systems, and archival data systems.

Report Organization

The remainder of this report is organized as follows:

Chapter 2 provides a review of key mobile applications developed to support various transportation agencies in the areas of planning, safety, operations, and general data collection. The chapter also contains information on commonly used frameworks for mobile application development, including their strengths, limitations, and popularity.

Chapter 3 includes a detailed description of the methodology used to build the components of SWiZAPP.

Chapter 4 discusses the deployment and results of field testing SWiZAPP.

Chapter 5 summarizes the key outcomes, recommendations, and limitations of the project.

Finally, a manual documenting step-by-step use of SWiZAPP is included as an appendix.

CHAPTER 2. LITERATURE REVIEW: MOBILE PHONE APPS IN TRANSPORTATION

Work zones are one of the critical conflict areas leading to crashes. Thousands of crashes involving motor vehicles, pedestrians, bicycles, and construction workers are recorded every year in the United States, many of which occur within highway construction work zones. The main reason reported for these crashes are the insufficient traffic controls and driver misjudgment (Li and Qiao 2016). Achieving smarter work zones is one of the main goals of the Federal Highway Administration (FHWA). For this purpose, many ITS tools and applications have been developed and implemented to effectively mitigate traffic impacts caused by construction (Liao and Donath 2016).

Mobile Applications for Transportation

Smart mobile applications are software applications that are designed to run on smartphones, tablets, and other mobile electronic devices. In this era of rapid technological advances, these applications have become one of the primary tools we use daily both in our personal and professional lives (Siuhi and Mwakalonge 2016). Smartphone applications have many uses related to transportation, including route planning, ridesharing/carpooling, traffic safety, parking information, transportation data collection, fuel emissions and consumption monitoring, and travel information.

In the transportation sector, the early use of mobile applications was mainly for navigation and location-based services. As of 2016, mobile applications are used for many transportation-related applications including engineering education, traffic data collection, travel information, route planning, and ridesharing (Siuhi and Mwakalonge 2016). In the literature, there are numerous applications available for traffic safety, transportation emission quantification, transportation data collection, and navigation (Dutzik et al. 2013). Realizing the potential of the applications in the transportation area, several state DOTs have already implemented mobile applications that show real-time travel information to the traveling public.

In a study by Siuhi and Mwakalonge (2016), smart mobile applications were categorized according to their possible utilization in the transportation industry, such as for data collection, route planning, ride sharing, and conducting travel surveys. Tables 1 through 5 show some of these applications.

Table 1. Route planning applications

Acronym used	Key features
Easy Route Finder	Find a distance and time to drive/walk to reach a destination from a given start point.
Runtastic Road Bike Tracker	Search bike routes and tracks bike rides via GPS: distance, duration, speed, elevation gain, pace, calories burned, etc.
Voyager: Route Planner	Find an optimal (fastest) route to travel to multiple locations.
Journey Pro	Provide the best route using various combinations of available modes and real time journey planning around some major cities in UK.
GPS Route Finder	Find the easiest and fastest driving/walking route from a starting to a destination point.
Driving Route Finder	Find the driving routes between any start and end locations.
ROUTE 66 Navigate	Provide lane guidance and speed limits for optimal guidance for car driving navigation.
MySmartRoute	Optimize routes for multiple locations from a start point.
Smart Ride	Find the best travel connection based on multimodal travel information service in Austria.
BestRoute	Provide an optimal multi-destination route planning.
Route Planner	Find the easiest and fastest travel route to a destination from a start point.
Road Tripper	Provide trip planning, easily find places, save places, and send them to your favorite navigation app.
ESPO Free–Route Planner	Optimize many destinations by car or on foot.
Route4Me	Gives the most optimal multi-destination routes making trips quick and easy.
RouteOptimizer	Optimize the route from a start to a destination.
Bike Route Planner (& Tracker)	Assist route planning and navigation to cycling and walking.
Bike Hub Cycle Journey Planner	Find the quickest or quietest cycling routes.
Trip Planner	Help a traveler to organize everything needed for a trip.

Table 2. Traffic safety applications

Acronym used	Key features
ZoomSafer	Automatically detect driving and provides reminders to use your device safely while driving to reduce distractions.
Text'nDrive	Read your emails and let you reply to them all using voice.
IGuardianTeen	Keep teen drivers distraction-free and focus on the road to prevent texting and phone calls while driving, keeping a log of any communication and send report to the parent.
TextArrest	Limit distracted driving by prevents the temptation of teen drivers to use their cell phones while driving.
Drivesafe.ly Pro	Feature a one tap operation and auto-on functionality that allow a user to seamlessly interact with a phone without texting or emailing while driving.
Steer Clear Mobile	Help young drivers to reinforce positive driving behavior and stay aware of hazards on the road by giving them tips on how to drive and what to look out for in a variety of weather conditions.
SafeCell 360	Provide parents with oversight on the use of their driver's access to smart device while driving.
Cellcontrol	Determine when you are driving and eliminate the temptation of a driver to talk, text, email and surf the web while driving to stop distracted driving.
tXtBlocker	Disable cell phone and smartphone features when a younger driver is driving.
SaferCar	Provide important information and functions that will help the user to make informed safety decisions involving their vehicle.
Speed Watcher Free	Provide an analog speedometer and odometer that gives audible and visible warning/alert when a set speed limit is exceeded.
AT&T DriveMode	Silence incoming text message alerts, and helps avoid distractions allowing a driver to stay focused while driving.
No Texting While Driving App	Block all incoming calls and puts an auto-reply to all texts coming in.

Table 3. Transportation data collection applications

Acronym used	Key features
TurnCount	Count vehicles and pedestrians entering an intersection.
R&S Counts	Allow counting cars for a company, receiving count information, and submitting traffic count electronically when you are done.
SpeedClock	Use the camera to detect and analyze motion and a reference distance to calculate the speed of an object.
Speed Tracker Free	Help to gather all the necessary trip statistics: speed, time, distance, heading, elevation, etc.
DigiHUD Speedometer	Show useful speed and distance information for your journey.
Car Dashboard Pro	Replaces speedometer and a car home dock.
SpeedView: GPS Speedometer	Show your current, maximum and average speed, direction, total distance, and time traveled.
Speedometer Free Speed Box	Track your speed and distance.
Vehicle Speed Logger	Track, monitor, and keep a minute-by-minute log of speed, date, time and location.
Speed Gun	Measure the speed of a moving object.
Speed Limiter	Give the user five different speeds to choose and give an alert if you outside your current active speed limit range.
Speed limit	Check vehicle speed and warn if you drive faster than the speed limit to avoid speed tickets.
Trip logger	Detect device movement and automatically track mileage.
Mileage Log+	Track and log vehicle mileage for tax deduction or reimbursement.
Saga Traffic; Trip Tracker and Travel Surveys	Collect location data about the user's movement and travel.
Traffic Survey	Manual traffic volume count at a road intersection.

Table 4. Travel information applications

Acronym used	Key features
Beat the Traffic Plus	Provide real time traffic specific to your route.
Sigalert.com	Feature real time updates on traffic and road speeds in the U.S.
ATC Delays	Allow the user to view airport ground stops and delays, arrival and departure delays, and airport closures.
Georgia 511	The official traffic app of the Georgia DOT, provides real time access to traffic and travel information.
VDOT 511	The official traffic information from the Virginia DOT, provides access to current and future traffic and roadway information.
NMRoads	Provide New Mexico and interstate motorists with mobile access to up-to-date travel and traffic information.
UK Bus Checker	Show when buses arrival and destination for 300,000 bus stops in the mainland UK.
CDOT Mobile	The official app from the Colorado DOT, provides real time travel data on its highways.
MoDOT Traveler Information	Show current work zones, incidents, and weather-related road conditions on state-maintained routes in Missouri.
Traffic Cameras + Toll and Travel Information	Allow to monitor traffic cameras around the world (including Australia, New Zealand, England, USA and Europe) and the user can also monitor toll and travel cards.
TDOT Smartway	Provide up-to-the-minute traffic information in Memphis, Nashville, Chattanooga, and Knoxville in Tennessee.

Table 5. Smart mobile application challenges

Acronym used	Key features
Trapster	Help drivers by reporting traps and hazards such as accidents, construction zones, traffic jams, live police, when spotted.
Cobra iRadar	Allow users to access live radar/laser detection and allow users to share and receive alerts in real time to avoid traffic enforcement such as speeding and red light cameras.
AES Malaysia	Automated enforcement system avoidance app in Malaysia.
PhantomAlert	Give audible and visual alerts while driving on the surrounding such as speed traps, red light cameras, speed cameras, and DUI check points.
Glob	Warn while driving close to speed cameras, speed traps, accidents or road works.
Police Radar	Show police traps, police radar, speed cams, speed traps, parking attendants, speed cameras, accidents and other traffic disturbances in your area added by application users.
Waze	Give road alerts along your route and find the cheapest fuel prices around you shared by the community.
Traffic Light Changer	Trigger a preemptive sensor on a traffic light that causes it to change from red to green.
CamSam	Give alerts in real time and warns against all fixed speed cameras worldwide.

Efforts to reduce worker injuries have focused on improving traffic control devices and minimizing confusion of motorists passing through work zones. For example, to supplement conventional traffic controls and enhance the safety of workers in work zones, a smartphone-based audio warning message (AWM) was proposed and tested in driving simulators by Li and Qiao (2016). Results indicated the AWM was able to effectively increase driver awareness of traffic signs and dynamic traffic situations, in particular, in hazardous situations.

The National Highway Traffic Safety Administration (NHTSA) and state transportation authorities have implemented many safety countermeasures to reduce forward collisions in work zones. However, due to the complexity of traffic in work zones, traditional countermeasures often fail to prevent crashes.

A study by Craig et al. (2017) examined the potential effects of in-vehicle messages to communicate work-zone events to drivers. The researchers conducted a work-zone safety survey in Minnesota to clarify drivers' attitudes toward work zones, along with smartphone use and in-vehicle messages through smartphones.

The researchers considered three different messaging interfaces: a roadside, portable changeable message sign; a smartphone presenting only auditory messages; and a smartphone presenting audio-visual messages. This research found that driver performance on speed deviation and lane deviation is better with in-vehicle messages versus that with roadside signs. Furthermore, drivers reported significantly less mental workload and eye gaze behavior for the in-vehicle conditions relative to the roadside sign condition.

Researchers with the Innovative Transportation Research Institute at Texas Southern University developed a smartphone-based warning system application that provides different types of warning messages, including sound, visual, and voice, to alert drivers to hazardous traffic situations. They measured driving behaviors in terms of headway distance, headway time, speed, and acceleration/deceleration. Results of this research showed that voice messaging is the most effective approach to instruct drivers to control their speed smoothly and keep sufficient headway time and braking distance while driving through a work zone. These driving simulator-based studies also found that smartphone-based warning systems could reduce both vehicle-to-vehicle crashes and worker fatalities (Rahman et al. 2016).

The safety of pedestrians is also an important concern in work zones. The FHWA reports that 17% of all work-zone fatalities, annually, are pedestrians. In addition, people who are visually impaired often confront physical and informational barriers that limit their accessibility and mobility. (The Manual on Uniform Traffic Control Devices [MUTCD] for Streets and Highways mandates temporary traffic control [TTC] to provide audible information for visually impaired people in work zones.)

Researchers with the University of Minnesota developed a smartphone app that works in conjunction with embedded sensors on the smartphone and Bluetooth technology to provide routing instructions to pedestrians as it detects upcoming work zones. When a work zone is detected, the smartphone vibrates to alert users, and the app then announces a corresponding audible message to users (Liao 2014).

Mobile devices and mobile applications have also been utilized in asset management projects. Research by Khan et al. (2016) developed a transportation asset management system (TAMS) as a comprehensive asset management tool to collect and manage the location and attributes of transportation assets and associated geographical features, such as road design, bridges, and the locations of horizontal curves, in the field.

Mobile App Development Frameworks

Mobile app development is increasing rapidly due to the popularity of smartphones. There are currently about 3,300,000 apps on Apple's App Store. (Pocket Gamer.biz 2018) and 2,900,000 apps on the Android market (AppBrain 2018). In general, mobile apps are categorized into three groups: native, web-based, and hybrid.

Native applications run on a device's operating system and are required to be adapted for different devices. Web-based apps require a web browser on a mobile device. Hybrid apps are "native-wrapped" web apps (Ferrari and Gerla 2010). A recent survey (Appcelerator / IDC 2012) revealed that developers are mainly interested in building native apps because they can utilize the device's native features, such as cameras, sensors, accelerometers, and geolocation.

One of the big problems with mobile application development is that the mobile market is divided among many platforms and, as a result, development time increases and more skills are needed. Cross-platform frameworks, which let programmers develop a single application for several platforms, is a solution to this problem. Some of the advantages of cross-platform development are cost-effectiveness, a single technology stack, reusable code, and easy maintainability. Xamarin, Ionic, and React Native are three popular cross-platform frameworks in the mobile application industry.

Since September 2015, the cross-platform framework React Native, which was created by Facebook, has been available for public use. This framework uses native scripting to create actual native components. The React Native framework promises the concept "Learn once, write everywhere," which means that once developers have learned the React Native framework, they will be able to apply it to multiple platforms. It allows development of mobile applications using concepts derived from the web framework ReactJS and allows their creation in a similar way to how web applications are developed using ReactJS. In addition, React Native is an open source framework that allows programmers to contribute to its further development (Hansson and Vidhall 2016).

Compared to other existing cross-platform frameworks, such as Cordova or Ionic, React Native uses a flexible structure for rendering mobile applications. Traditionally, combinations of JavaScript, hypertext markup language (HTML), and cascading style sheets (CSS) are usually used to write mobile web applications. Apps developed using these technologies usually do not have access to the host platform's set of native user interface (UI) elements. React Native translates the markup to real, native UI elements. In addition, React Native performs separately from the main UI thread, so applications can have high performance without decreasing capability (Eisenman 2017).

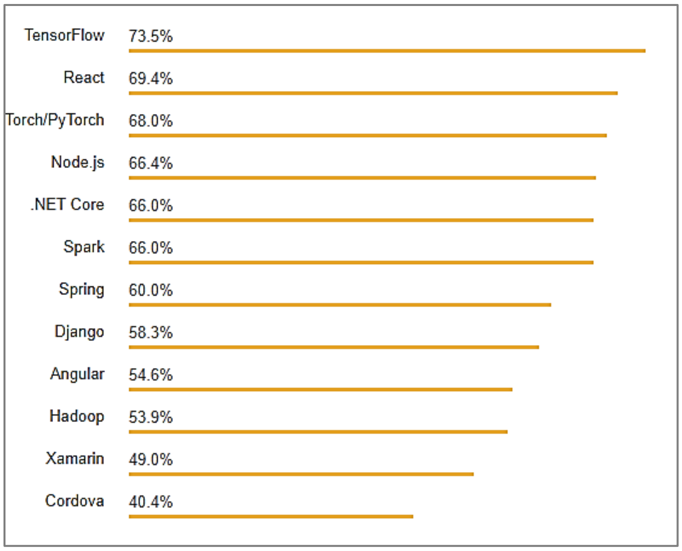
Another advantage of React Native is that it lets the programmer take advantage of intelligent debugging tools and error reporting and use any text editor for JavaScript editing; therefore, it does not force the programmer to work in Xcode to develop for iOS or in Android Studio for Android development (Eisenman 2017). Table 6 provides a comparison between three cross-platform frameworks: Xamarin, Ionic, and React Native.

Table 6. Comparison of Xamarin, Ionic, and React Native

		Xamarin		Ionic	React Native
Code		C#		HTML, CSS, TypeScript, JavaScript	JavaScript^{+}, Java, Objective-C, Swift
Compilation/ Interpretation	iOS	AOT		JIT with WKWebView	Interpreter
	Android	JIT/AOT		JIT	JIT
UI rendering		Native UI controllers		HTML, CSS	Native UI controller
GitHub stars		3.6K		33.3K	59.6K
Price		Open source		Open source	Open source
Additional costs		Visual Studio IDE* $539–$2,999 For commercial use		Ionic Pro $29–$199 For additional features	-
Cross-platform probability		Xamarin iOS Android	Xamarin Forms	Up to 98% of source code reuse	Adapting code to each platform
Performance		Close to native	Moderate-low	Moderate-low	Close to native
Use cases		All apps	Simple apps, corporate apps	Simple apps, corporate apps	All apps

Source: AltexSoft 2018

As shown in Table 6, based on GitHub statistics, React Native is the most “starred” framework among these three, with 59.6 thousand stars. According to the results of the 2018 Stack Overflow developer survey (Stack Overflow 2018), the React Native framework is not only the most used cross-platform tool, but also the second most “loved” one among developers in general (see Figure 1).



Stack Overflow 2018

Figure 1. Most "loved" frameworks, libraries, and tools

There are different open source frameworks available on the market. Table 7 provides a summary comparison between ReactJS and three other open source frameworks: AngularJS, Ember.js, and Aurelia.

Table 7. Pros and cons of mobile app development frameworks

Framework	Favorites* /Fans	Pros	Cons
AngularJS	454 /9.95K	▪ Quick to develop ▪ Great mvc ▪ Powerful ▪ Restful ▪ Backed by Google ▪ Two-way data binding ▪ Javascript ▪ Open source ▪ Dependency injection ▪ Readable	▪ Complex
Ember.js	73 /534	▪ Elegant ▪ Quick to develop ▪ Great mvc ▪ Great community ▪ Great router ▪ Values conventions, there is one true way to organize ▪ Open source ▪ Mvc framework ▪ Handlebars.js	▪ Too much convention, too little configuration ▪ Hard to use if the API is not Restful ▪ Very little flexibility
Aurelia	61 /153	▪ Simple with conventions ▪ Modern architecture ▪ Makes sense and is mostly javascript not framework ▪ Extensible ▪ Integrates well with other components ▪ Easy to use ▪ Dependency Injection ▪ Modular ▪ Great router ▪ Adaptive Data Binding	▪ Difficult to learn ▪ mandatory to use Typescript ▪ Two-way data binding
React	687 /8.89K	▪ Components ▪ Virtual dom ▪ Performance ▪ Simplicity ▪ Composable ▪ Data flow ▪ Declarative ▪ Not an MVC framework ▪ Reactive updates ▪ Explicit app state	▪ Requires discipline to keep architecture organized

Source: StackShare 2018

Commercial frameworks are expensive but relatively easy to use, while open-source frameworks may require developers to do more fundamental development and troubleshooting.

The React Native framework was selected for SWiZAPP for two main reasons: it is free and open-source and it is the most popular one among the frameworks. Therefore, the updating process would be expected to be unproblematic.

CHAPTER 3. SWIZAPP DEVELOPMENT

SWiZAPP consists of four main components (shown in Figure 2): log-in screen, Projects, Map View, and Tracker.

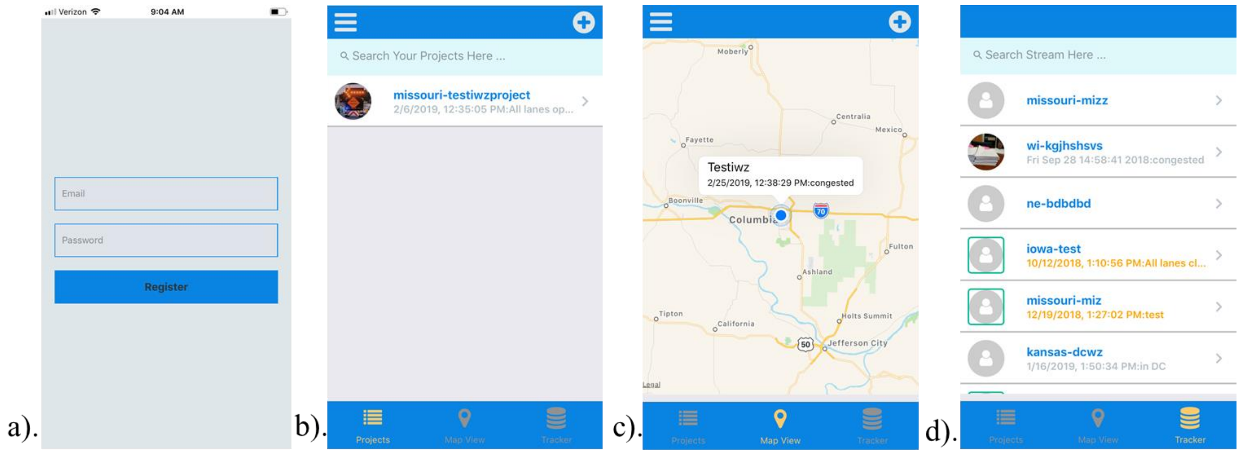


Figure 2. Key components of SWiZAPP: (a) log-in, (b) Projects, (c) Map View, (d) Tracker

All first time users of the mobile application are required to register with an email address and password via a log-in screen. Once a user is registered, SWiZAPP is designed to automatically keep the user logged in on the same device. The credentials of each user are stored in a database to manage user activities.

After a user successfully logs into SWiZAPP, they are granted permission to use the full functionalities of the app, including adding and updating work-zone information, geolocating the start and end of the work zone, and posting real-time work-zone-related activities.

The SWiZAPP Projects screen displays all work zones managed by the user. From this screen, users can add new work zones, update existing work zones with new information, and perform geolocation. This view also hosts a search engine for filtering work zones based on the state (such as Iowa) or the work-zone identification number.

The Map View displays the locations of all geolocated work zones on a map. Users can also access the most recent work-zone activities from this screen.

The Tracker screen monitors and relays work-zone activity information to all SWiZAPP users. It displays a list of all work zones being monitored by the app, their recent or past activity status, the last time the information was updated, etc. It also has a search component for filtering work zones based on their unique identification numbers, states, or names of contact people.

Design Approach

To allow for future expansion and integration of the app by individual agencies, SWiZAPP was designed using a modular design process. The current design consists of four main modules: a frontend UI module, a sensor module, a middleware communication module, and a backend module. Figure 3 shows the interactions and flow of information between these modules.

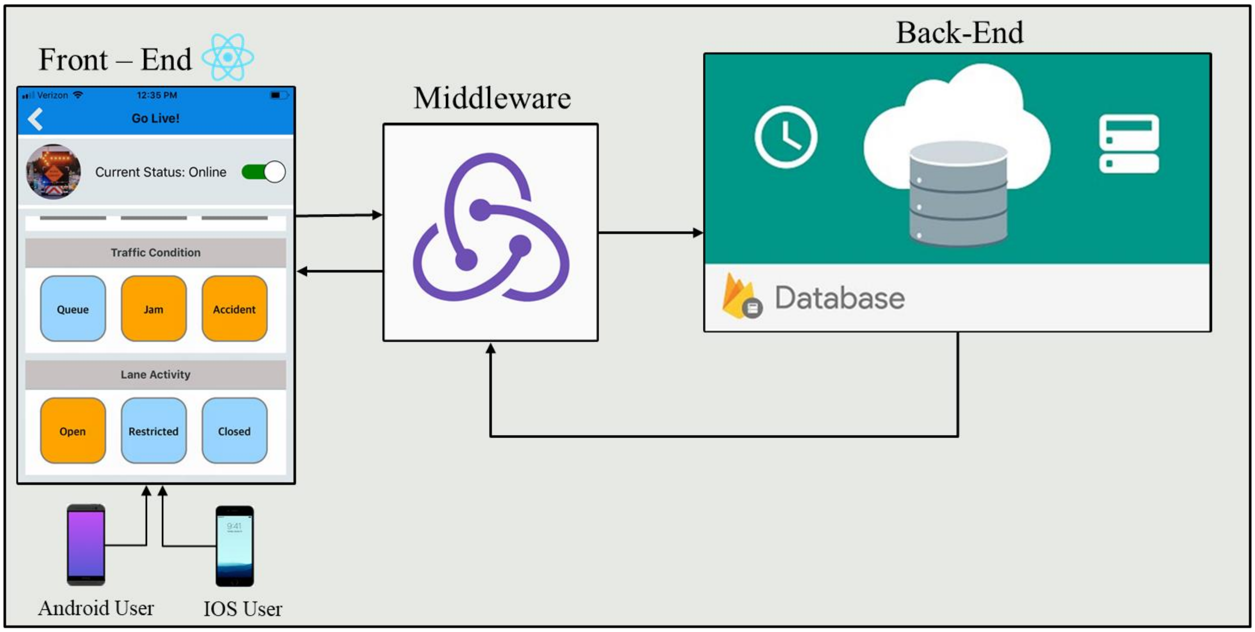


Figure 3. Frontend, middleware, and backend modules of the app

A user makes requests and updates by connecting to the frontend user interface, and the middleware receives all actions from the user and passes them to the backend, which responds to the user's request and sends an appropriate response through the middleware back to the user on the frontend. The following sections provide details on each module.

Frontend Module

The goal of the app's frontend is to mask all requirements related to software specifications by using a variety of layouts and UI elements. Examples of layouts used in the development of SWiZAPP included flat lists and grid and scroll views. Also, key UI elements such as drawer menus, buttons, switches, dialog boxes, and notifications were used to improve the user friendliness of the app.

Navigation

Three main types of navigation were used in the design of SWiZAPP: tab navigation, stacked navigation, and drawer menus.

Tab Navigation

Tab navigation, possibly the most common style of navigation in mobile applications, enables the user to switch between all the key features of an app from one screen. Tabs are very flexible and make the app user-friendly. After log-in, SWiZAPP consists of three key components: Projects, which lists all the work zones added by the user; Map View, which diplays the locations of the work zones on a map; and Tracker, which reports all work zone-related activities from all users of SWiZAPP. We used React Native’s tab navigation module to enable the user to switch between these key components. Figure 4 shows how tab navigation is used to navigate between the three components of SWiZAPP.

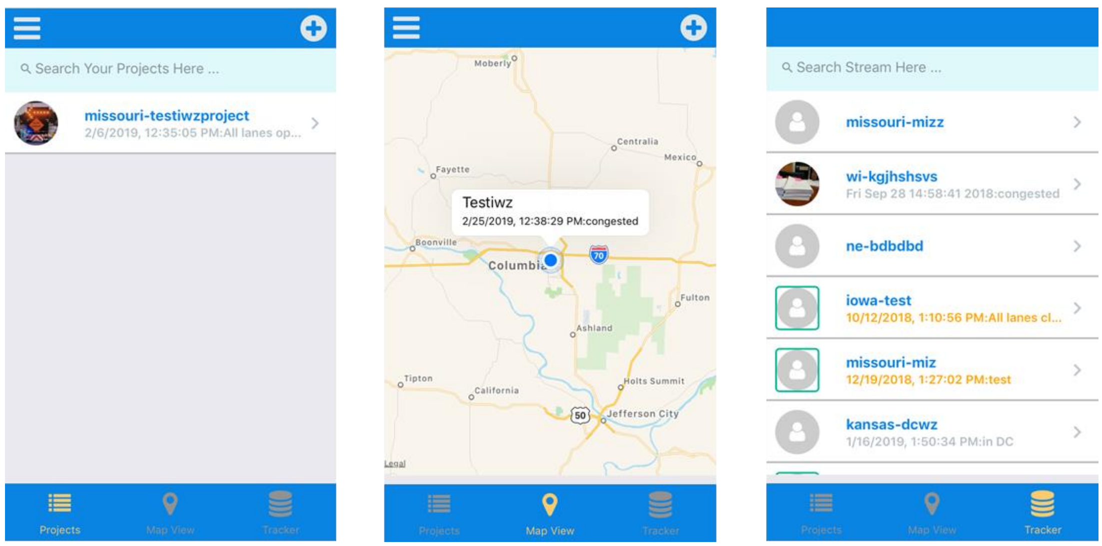


Figure 4. SWiZAPP tab navigation example

Stacked Navigation

Compared to tab navigation, a stacked navigation provides a way for SWiZAPP to transition between screens such that each new screen is placed on top of a stack. In tab navigation, the user can navigate from the first screen to the last screen without going through the intermediate screens. Stacked navigation however, requires the user to go through all the screens in the stack sequentially. In Figure 5, we show an example that necessitates using stacked navigation.

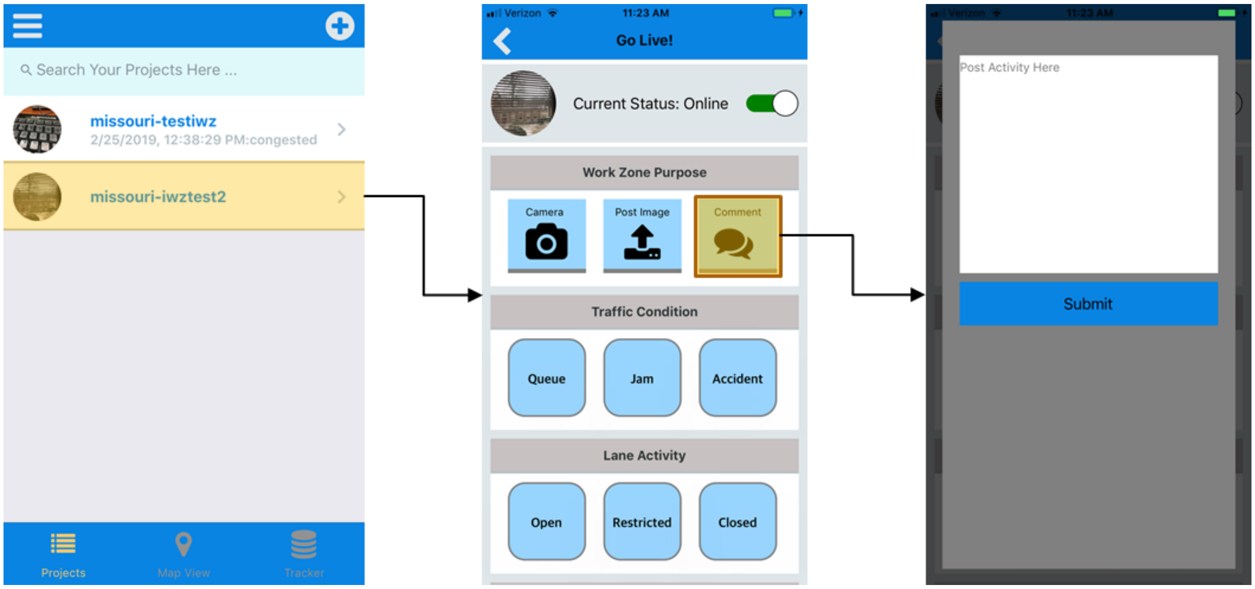


Figure 5. SWiZAPP stacked navigation example for posting activities to a work zone

In this example, the user needs to post an activity to a specific work zone. They must first select the specific work zone to which the activity is being posted. The app then navigates to a second screen, where the user must select the type of activity, and then a final screen for submitting the activity.

Drawer Menus

Mobile screen space is a precious commodity. Drawer menus help save screen space while offering an intuitive navigation. An example of how drawer menus are used in SWiZAPP is shown in Figure 6.

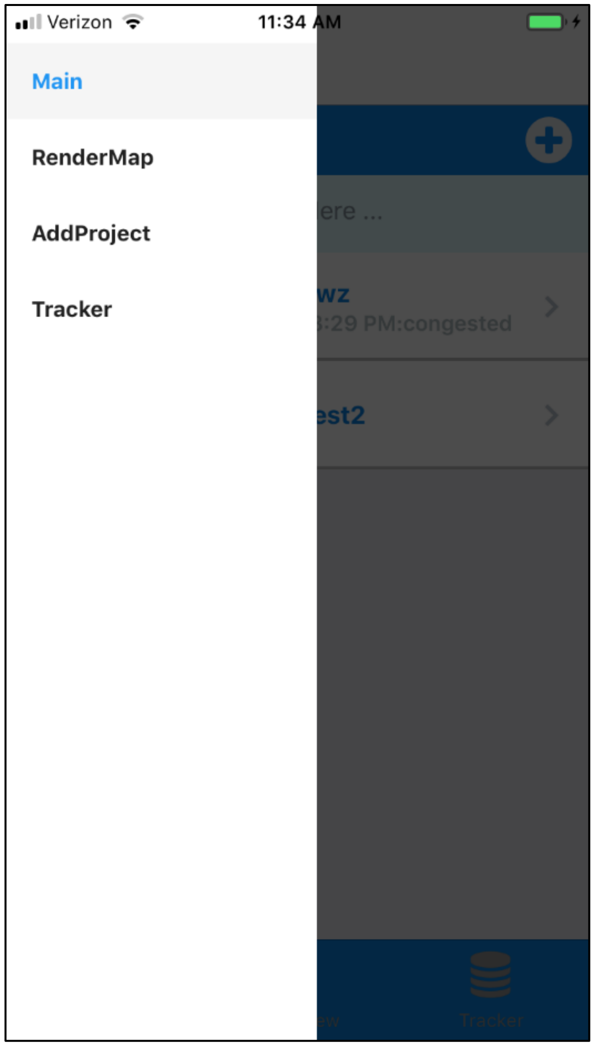


Figure 6. Drawer menu

Buttons

A number of customized buttons of different shapes were designed and integrated into the app. These buttons (shown in Figure 7) include a switch button used to activate and deactivate the work zone, a circularly shaped button for updating and displaying work-zone profile pictures, and square-shaped buttons for posting work-zone activities.

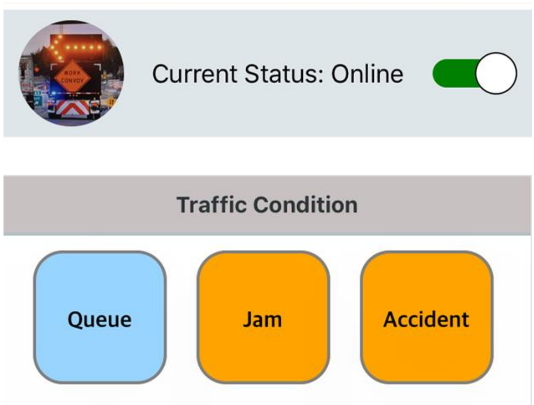


Figure 7. Customized buttons used in SWiZAPP: circularly shaped (top left image in a circle), switch (top right), and rounded square buttons (across the bottom)

Layouts

SWiZAPP was designed using popular modern and user-friendly layouts such as swipeable flat lists, grid views, and scroll views.

Flat Lists

Flat lists are useful for displaying a list of items on a one-dimensional grid. They summarize key information for each list, giving the user a preview of what the item contains. For this app, each item in a flat list is uniquely designed to allow swiping gestures that enable the user to make changes specific to a particular item by swiping left or right.

Grid Views

Grid views are used to display items in a two-dimensional grid. For this app, we implemented a responsive grid layout that adapts itself to various screen resolutions. A grid view layout is used in one instance (Figure 8) to display the different types of work-zone activity posts that are supported by SWiZAPP.

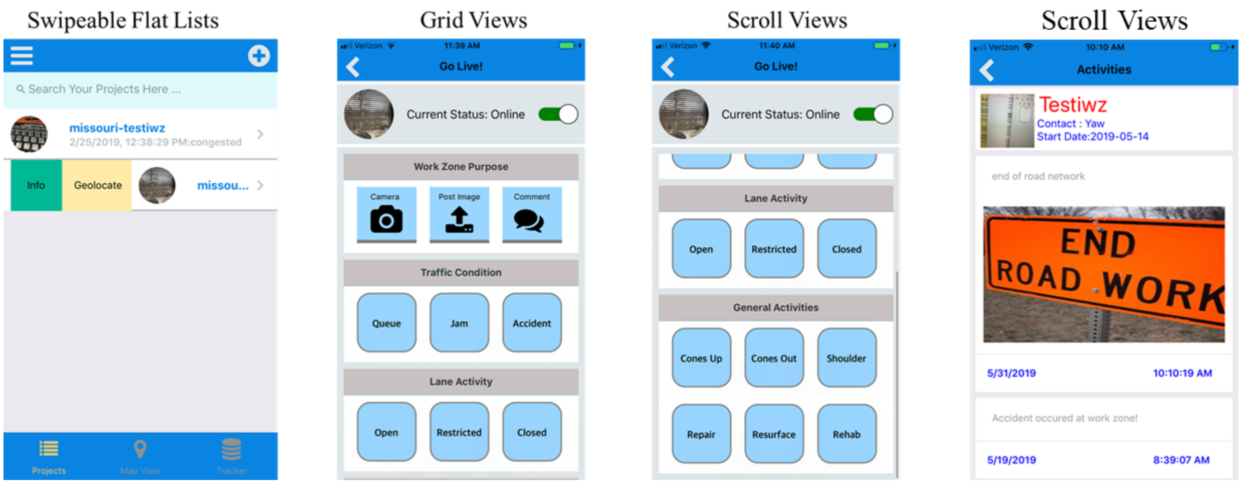


Figure 8. SWiZAPP layout examples: swipeable flat list, grid view, and two scroll views

Scroll Views

Flat lists and grid views, in general, tend to exceed the default height of the mobile device's screen. To enhance the usefulness of these layouts, scroll views enable the user to view the contents of a page exceeding the height of the screen by scrolling up and down. Examples of scroll views for SWiZAPP can be seen on the Projects page, the Tracker page, and the activity pages for specific work zones (see Figure 8).

Alerts, Prompts, and Notifications

A number of prompts and alerts were used to enforce requirements for data entry and posting by users of the app. For example, in Figure 9, a prompt is used to request further information from the user about the location of the work zone.

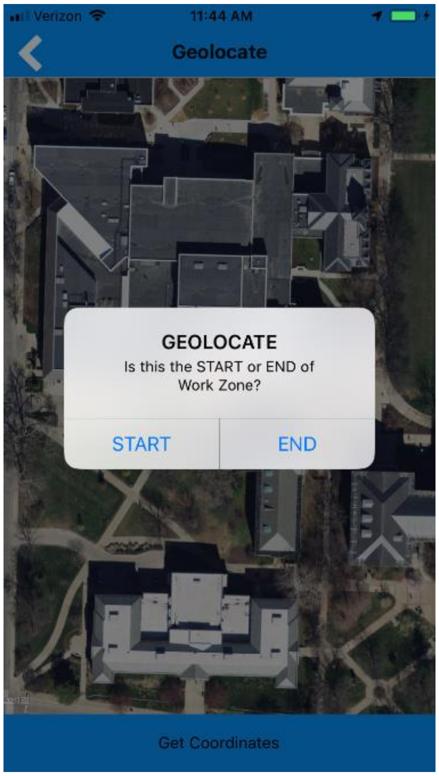


Figure 9. SWiZAPP's geolocation prompt example

Sensor Module

The sensor module involves the management of two key technologies: GPS and camera. Users must grant SWiZAPP access to these sensors before they can be used. The coordinates received from the GPS allow the geolocating of the work zone. These coordinates can complement any linear referencing system (LRS) information that is entered manually, e.g., route and log mile. The coordinates will enable the work-zone activity information to be easily located on maps and geographical information systems (GIS). SWiZAPP requests both the start and end location of a work zone. To geolocate a work zone, the user must be on-site. A satellite view of the work-zone location is shown to the user as a form of verification before a request to get coordinates is made. Upon work-zone location request, the app will ping the current location of the user to represent either the start or end of the work zone. SWiZAPP can also access the phone/device camera to take snapshots of the events occurring at the work zone and broadcast them to users.

Backend Module

The core of SWiZAPP is its backend module. It is responsible for storing all user or work-zone information and supports user requests from the frontend. To enable real-time monitoring of work-zone activities, the backend had to be designed such that it can respond to requests and post information in real-time. We deployed a cloud-based, real-time database and storage system using Google's Firebase (see Figure 10).

Figure 10. Real-time posting of user requests

Firebase is a robust mobile and web application development platform, which was developed by Firebase, Inc. in 2011, then acquired by Google in 2014. It provides an application programming interface (API) that enables data to be synced across all clients in real-time and remain available when the app goes offline. One of the key reasons for choosing Firebase to drive the app's backend was due to its ability to store and manage unstructured datasets, such as those consisting of images, text, and numbers. The SWiZAPP data feeds are unstructured; hence, Firebase was a good fit.

SWiZAPP's backend has three main functionalities: authenticate users, provide storage for user data, and manage a real-time database. Google's Firebase serves as the platform for carrying out all these functionalities. Before a user can use the app, they must be authenticated. Firebase Authentication provides the ability to authenticate users via email and password, phone number, Facebook, Google, Twitter, or Github. The flow of authentication is illustrated in Figure 11.

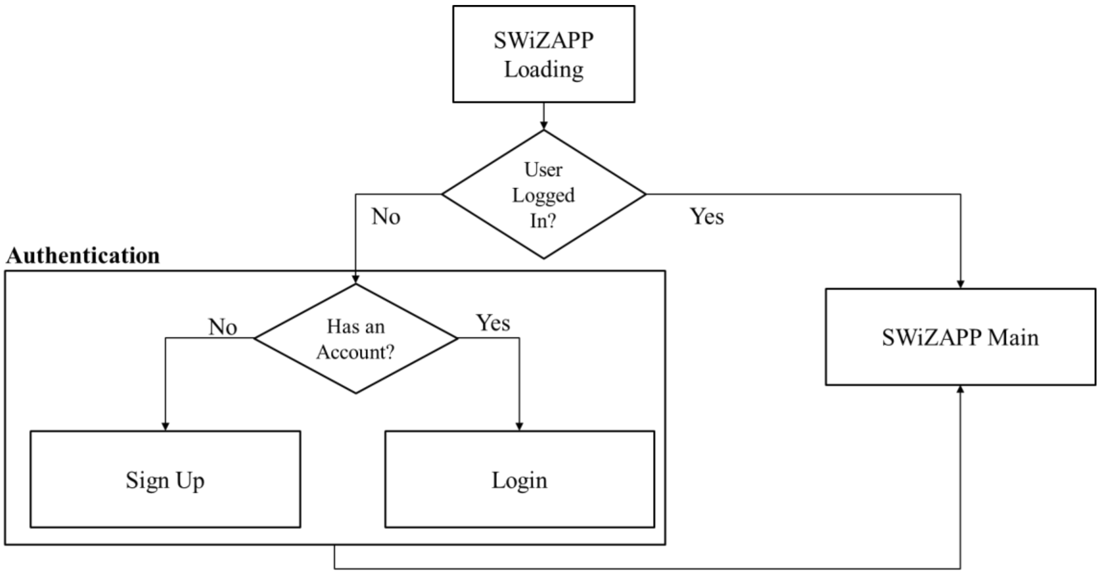


Figure 11. SWiZAPP authentication framework

To simplify SWiZAPP development, users are authenticated only by email and password. The first time a user downloads the app, they are required to register using their email and password. The Firebase authenticator verifies that the information entered is correct before navigating to the main app screen. Any errors encountered during the authentication process are communicated to the user via alerts and prompts.

The structure of the backend real-time database is shown in Figure 12.

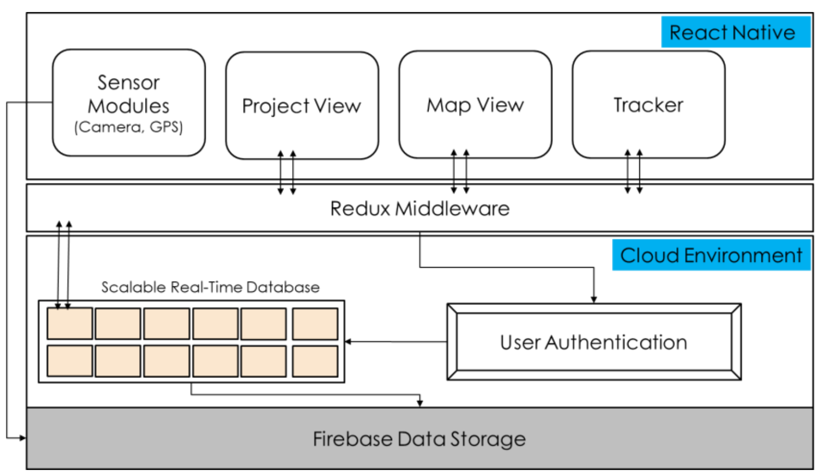


Figure 12. Backend integration into SWiZAPP framework

With the exception of snapshot images, all user data are pushed directly into Firebase's real-time database. Images are stored as blobs inside Firebase storage. Once an image is stored, a unique reference path to the location of the image is generated and kept in the real-time database.

Middleware and Communication Module

The middleware connects the frontend to the backend and vice-versa. The state of the mobile application data is designed to constantly change as different users simultaneously post work-zone activities and request information. The backend is expected to generate an appropriate response for all requests within reasonable time. Although middleware is not necessary for the intended purpose of this app, handling such state changes can be difficult as the number of users increases.

For this project, Redux was used as the middleware for SWiZAPP. It is used to fetch data from the Firebase backend via an API, update app states, and render different things to the user interface. Redux was set up to constantly monitor the app for state changes and requests from the frontend, relay changes to the backend, and fetch responses from the backend to deliver to users on the frontend. Figure 13 shows how the Redux middleware manages the app's actions and state changes.

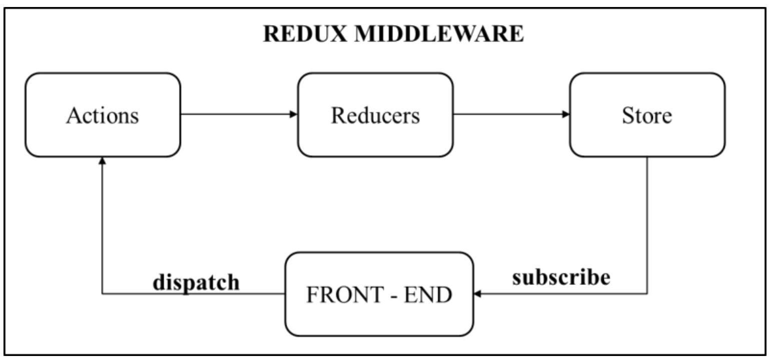


Figure 13. Flow chart of Redux middleware

The application sends an action from the frontend with some data. These actions go through a reducer, which takes the current state of the application and transforms it to a new state. The complete state of the application is stored in the Redux store.

CHAPTER 4. FIELD TESTING AND DATA COLLECTION

In order to use the app for field testing, we first compiled and uploaded SWiZAPP on both Google Play and the Apple App Store. Four work-zone sites within the City of Columbia, Missouri, were surveyed thereafter. Locations of the sites selected are shown from the SWiZAPP Map View in Figure 14.

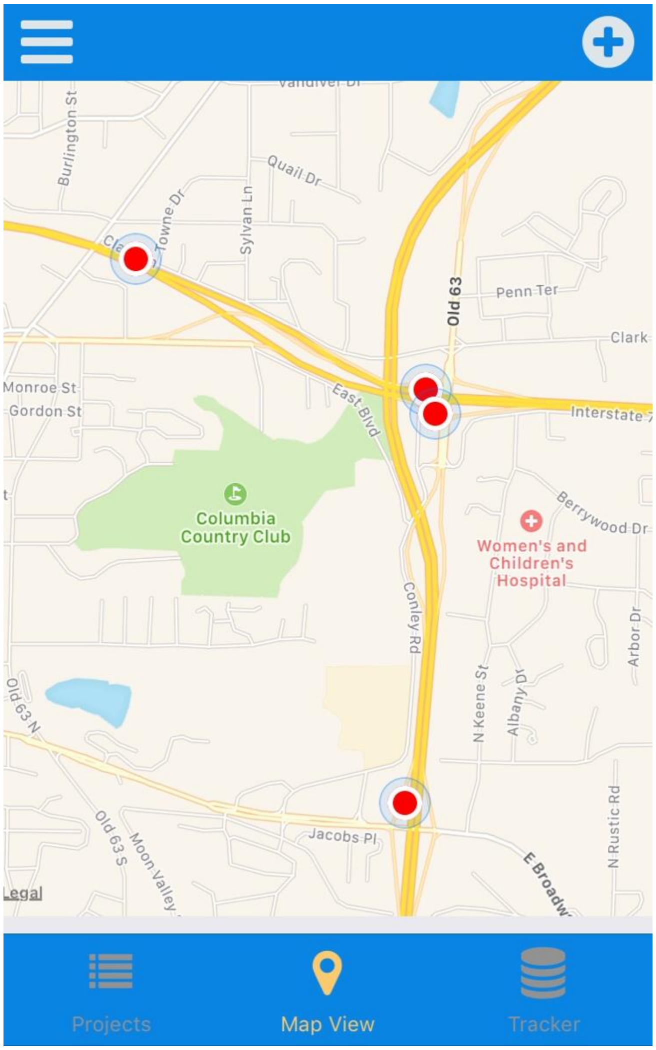


Figure 14. Selected work-zone sites for field testing

The key metrics used to evaluate SWiZAPP during field testing are covered in the remainder of this chapter:

- Geolocation accuracy
- User-friendliness
- Scalability

Geolocation Accuracy

Although SWiZAPP uses satelitte imagery maps to enable users to verify the accuracy of the work-zone locations provided, it was necessary to evaluate the expected range of accuracies depending on the location characteristics of the work zone. The accuracy of GPS coordinates provided by the app was therefore compared with ground truth coordinates manually extracted from Google Maps. The influence of four conditions (outlined in Table 8) on geolocation accuracy were evaluated: the first and second conditions are when the user acquires the work-zone location from inside and outside a vehicle, respectively, in a central business district (CBD) area. The third and fourth conditions look at geolocation inside and outside a vehicle on freeways.

Table 8. Range of geolocation accuracies by condition of the work-zone site

	Geolocation Accuracy Range	
Condition	**iOS**	**Android**
In-vehicle geolocation on local roads in central business district area	12–25 feet	6–15 feet
Out-vehicle geolocation on local roads in central business district area	2–15 feet	3–10 feet
In-vehicle geolocation on freeway	< 6 feet	< 6 feet
Out-vehicle geolocation on freeway	< 6 feet	< 6 feet

Overall, the difference between the extracted Google Map coordinates and those obtained from SWiZAPP was insignificant, with the exception of in-vehicle geolocation in CBD areas. This is expected, as high-rise buildings may affect the phone GPS signals. We recommend that, in such conditions, users should use the satellite preview to verify the proposed location suggested by the app before requesting the geolocation.

It was also observed that the type of phone (Android or iPhone) influenced the geolocation accuracies, especially in CBD areas. Android phones generally had higher accuracies and the GPS values recorded were more stable compared to those from iPhones.

User-Friendliness

To evaluate SWiZAPP's user-friendliness, the researchers surveyed a number of professionals and civil engineering students with some knowledge about the activities conducted in a work zone. Each user was first briefed on the purpose of the app and then allowed to navigate the app without any further assistance from the development team. A user-friendliness score was assigned based on the components the user was able to use. Figure 15 shows which of the components were easy to use and intuitive.

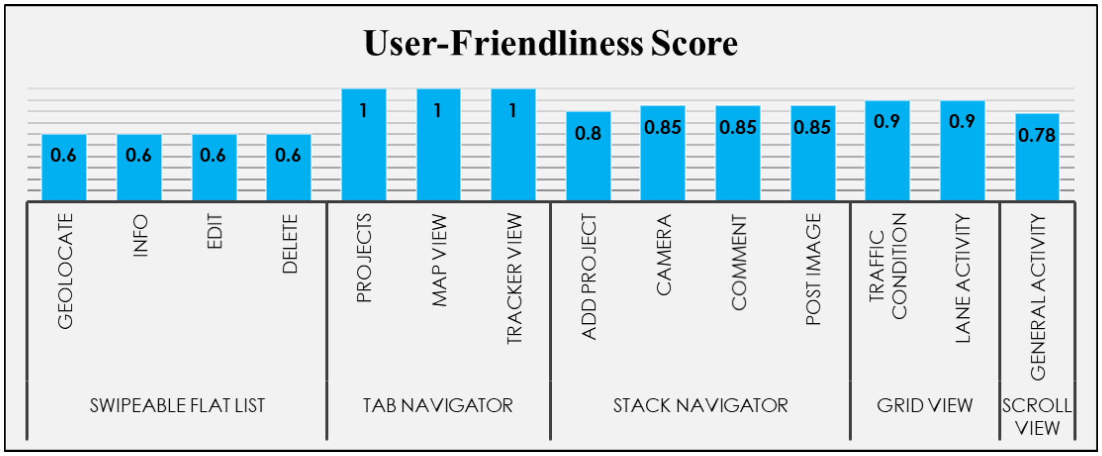


Figure 15. User-friendliness chart

Overall, the most challenging aspect of the interface for users to navigate was swiping left or right on a flat list to geolocate or edit a work zone. The remaining components were fairly straightforward to navigate.

Scalability

To evaluate the scalability of the app, the researchers monitored the latency between user activity post time and the activity broadcast time as the size of the SWiZAPP database was gradually increased by the addition of image snapshots. With the exception of image uploads, all other activities posted by users could be seen by other users of the app within a second (see Table 9).

Table 9. Scalability of SWiZAPP

Database size	Activity	Latency (in seconds)
10–50 MB	Upload image	3
	Upload text message	< 1
	Push alert button/Geolocate	< 1
50–100 MB	Upload image	6
	Upload text message	< 1
	Push alert button/Geolocate	< 1
100–500 MB	Upload image	7
	Upload text message	< 1
	Push alert button/Geolocate	< 1
500 MB–1 GB	Upload image	7
	Upload text message	< 1
	Push alert button/Geolocate	< 1
1–5 GB	Upload image	12
	Upload text message	< 1
	Push alert button/Geolocate	< 1

The latencies observed for image uploads increased as the size of the database increased. It is important to note that a fraction of these latencies could be due to network speeds, which SWiZAPP had no control over. A high-performance cluster could be used to reduce latencies and thereby increase the scalability of the app.

CHAPTER 5. CONCLUDING REMARKS

The current project designed, developed, and deployed a fully functional cross-platform mobile application, SWiZAPP, for collecting and communicating work zone-related activities. The app is enabled with functionalities for managing an unlimited number of construction work zones due to its scalable, cloud-based design architecture. It supports work-zone geolocation and mapping via on-board GPS sensors and Google Maps, respectively.

Users of the app can post live activities from construction sites by taking snapshots and uploading images, utilizing buttons within the app's interface to indicate traffic conditions and lane activities, or text messaging via the app. SWiZAPP also enables its users to view both real-time and historical activities of all work zones in the SWZDI states.

A modular design process was followed in the development of SWiZAPP. The current design framework consists of four main modules: a frontend UI module, a sensor module, a middleware communication module, and a backend module. A user makes requests and updates by connecting to the frontend user interface, and the middleware receives all actions from the user and passes those to the backend, which sends an appropriate response through the middleware back to the user on the frontend.

The key components of the mobile app were evaluated through a series of tests at four construction work-zone sites. The metrics for the evaluation included geolocation accuracy, user-friendliness, and scalability. Overall, the accuracy of the app's geolocation module was fairly high, although external factors such as proximity to high-rise buildings and the type of mobile device operating system (iOS or Android) could affect the overall geolocation accuracy.

Satellite imagery maps were used to guide users to verify the app's suggested location before confirmation. The ease of use of SWiZAPP was evaluated based on a user-friendliness score, which was dependent on how many UI components a user was able to navigate without assistance. Swipeable flat lists were found to be the most challenging to navigate.

Lastly, the developers tested the scalability of the app by increasing the size of the app's database over time and measuring the resulting latency. The researchers observed notable latencies in image upload speeds as the database size increased over 1 GB. This could be resolved by deploying a cluster of machines instead of a single node to support the backend.

SWiZAPP Limitations and Recommendations for Future Development

Future updates to and developments for SWiZAPP will need to address the following limitations of the current application.

The first and most obvious limitation of the app is its over-reliance on internet access. A sizeable number of construction work zones in many states may be in dead zones where internet access is limited. SWiZAPP will not function in such environments. Ideally, the app could be re-designed to store work-zone activity information on the device's storage system when there is no internet access. The locally stored data could later be pushed to the frontend and backend after the user re-enters an area with Wi-Fi access.

A second development that could be relevant in subsequent releases is live video streaming and/or archiving of work-zone activities. The current app does not have streaming capabilities, neither does it store video information. However, video feeds could increase the size of the SWiZAPP database at an exponential rate. This will increase the cost of cloud storage and require new database designs to enable the app to scale as the size of the data uploads increase. With the modular design approach adopted to develop SWiZAPP, such new extentions could be carried out seamlessly.

In addition, the app could be enabled with a chat area, where work-zone workers and DOT workers could communicate and seek approval for unplanned activities. All chat messages could be stored to enrich the SWiZAPP database for documenting work-zone activities on a particular day.

Finally, we recommend that concerted efforts be made to integrate SWiZAPP with key transportation data management systems, 511, etc. A restless API could be developed to allow agencies and developers to query and explore the SWiZAPP database in real-time.

REFERENCES

AltexSoft. 2018. Xamarin vs React Native vs Ionic: Cross-Platform Mobile Frameworks Comparison. https://www.altexsoft.com/blog/engineering/xamarin-vs-react-native-vs-ionic-vs-nativescript-cross-platform-mobile-frameworks-comparison/. Information retrieved September 6, 2018.

AppBrain. 2018. Android and Google Play Staistics. https://www.appbrain.com/stats/. Information retrieved August 31, 2018.

Appcelerator / IDC. 2013. *Voice of the Next-Generation Mobile Developer: IDC Q3 2012 Mobile Developer Report*. Appcelerator, Inc. and IDC. http://www.appcelerator.com.s3.amazonaws.com/pdf/Appcelerator-Report-Q3-2012-final.pdf.

Craig, C. M., J. Achtemeier, N. L. Morris, D. Tian, and B. Patzer. 2017. *In-Vehicle Work Zone Messages*. Minnesota Department of Transportation, Minnesota Local Road Research Board, St. Paul, MN. https://conservancy.umn.edu/bitstream/handle/11299/189538/MnDOT2017-19.pdf?sequence=1&isAllowed=y.

Dutzik, T., T. Madsen, and P. Baxandall. 2013. *A New Way to Go: The Transportation Apps and Vehicle-Sharing Tools That Are Giving More Americans the Freedom to Drive Less*. U.S. PIRG and Frontier Group, Denver, CO.

Eisenman, B. 2017. *Learning React Native, 2nd Edition*. O'Reilly Media, Sebastapol, CA. https://www.oreilly.com/library/view/learning-react-native/9781491989135/?ar.

Fedkiw, R. 2012. Sensors and Cellphones. Lecture Presentation for CS75N: Cell Phones, Sensors, and You. Stanford University. https://web.stanford.edu/class/cs75n/Sensors.pdf.

Ferrari, D. and M. Gerla. 2010. Mobile Computing, Applications, and Services. MobiCASE 2nd International Conference on Mobile Computing, Applications, and Services, October 25–28, Santa Clara, CA.

Hansson, N. and T. Vidhall. 2016. Effects on Performance and Usability for Cross-Platform Application Development Using React Native. MS Thesis, Linköpings Universitet, Linköping, Sweden. http://www.diva-portal.org/smash/get/diva2:946127/FULLTEXT01.pdf.

Khan, G., A. Bueff, I. Mihov, N. Tessema, J. Garrido, and A. Parnia. 2016. Development of Transportation Asset Management and Data Collection System (TAMS) Using Mobile Applications. *Procedia Engineering*, Vol. 161, pp. 1180–86.

King, T, C. Sun, and M. Virkler. 2004. Evaluation of a Freeway Work Zone Advance Speed Advisory System Using Multiple Measures. Transportation Research Board Annual Meeting, January 11–15, Washington, DC.

Li, Q. and F. Qiao. 2016. Performance Measures of Smartphone Warning Messages in Work Zones and Intersections. 23rd ITS World Conference, October 10–14, Melbourne, Australia.

Liao, C. F. 2014. *Development of a Navigation System Using Smartphone and Bluetooth Technologies to Help the Visually Impaired Navigate Work Zones Safely*. Minnesota Department of Transportation, St. Paul, MN.

Liao, C. F. and M. Donath. 2016. *Investigating the Effectiveness of Using Bluetooth Low-Energy Technology to Trigger In-Vehicle Messages in Work Zones*. Minnesota Department of Transportation, St. Paul, MN.

Pocket Gamer.Biz. 2018. App Store Metrics. https://www.pocketgamer.biz/metrics/app-store/ Information retrieved August 31, 2018.

Rahman, R., F. Qiao, Q. Li, and L. Yu. 2016. *Developing a Smartphone Based Warning System Application to Enhance the Safety at Work Zones*. TranLIVE, Houston, TX.

Siuhi, S. and J. Mwakalonge. 2016. Opportunities and Challenges of Smart Mobile Applications in Transportation. *Journal of Traffic and Transportation Engineering (English Edition)*, Vol. 3, No. 6, pp. 582–592. http://dx.doi.org/10.1016/j.jtte.2016.11.001.

Stack Overflow. 2018. Developer Survey Results 2018. https://insights.stackoverflow.com/survey/2018/. Information retrieved September 6, 2018.

StackShare. 2018. Top Comparisons. https://stackshare.io/stackups. Information retrieved September 10, 2018.

Sun, C., P. Edara, C. Brown, Z. Zhu, and R. Rahmani. 2014. *Calibration of Highway Safety Manual Work Zone Crash Modification Factors*. Smart Work Zone Deployment Initiative, Ames, IA.

APPENDIX: SWIZAPP MANUAL

This appendix is a user manual for the Smart Work Zone Activity app (SWiZAPP), which is a mobile application used for work-zone activity monitoring and providing accurate and timely work-zone information. SWiZAPP allows the timely communication of work-zone status.

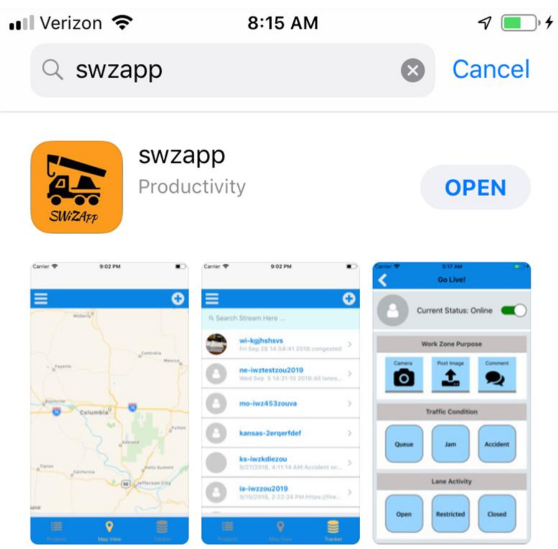
This app is capable of geolocating the work-zone location automatically, connecting to work-zone cameras (via Wi-Fi, 3G, 4G) to monitor work-zone activities in real-time, and alerting interested parties that permission is sought for commencing work or that work has commenced or ended.

The user-friendly interface includes standard work-zone procedures. The app design is suitable for use by both DOT staff and contractors. The benefits from the successful deployment of SWiZAPP include more accurate and timely work-zone information for work-zone management, traveler information, inspections, contract monitoring, safety analysis, and project coordination.

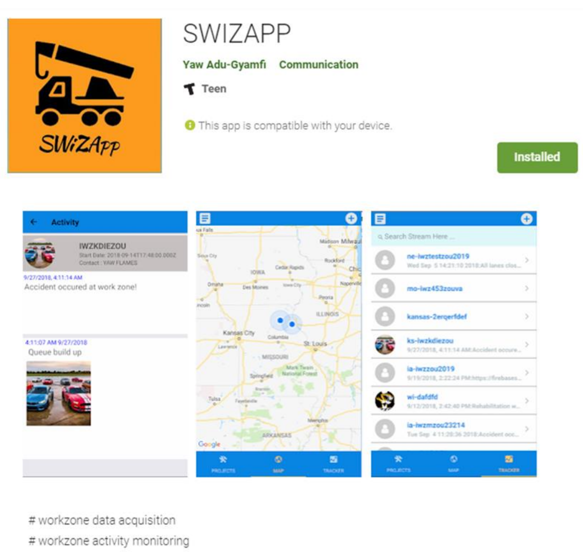
Download and Install the App

SWiZAPP is available on both the Apple App Store and Google Play.

APP STORE



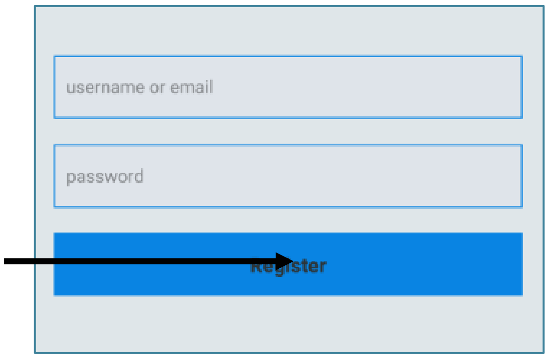
GOOGLE PLAY STORE

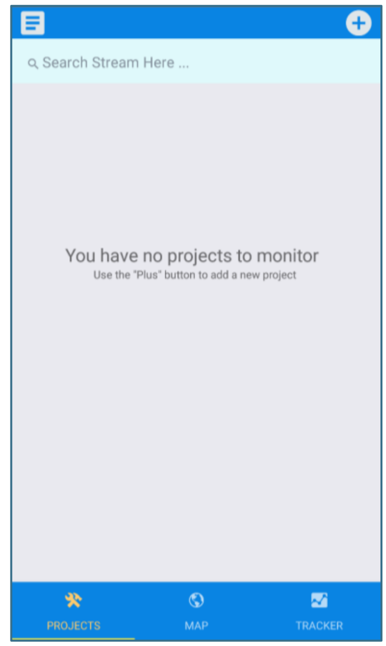


Register as a User

All users are required to register using a valid email address and password.

User Registration



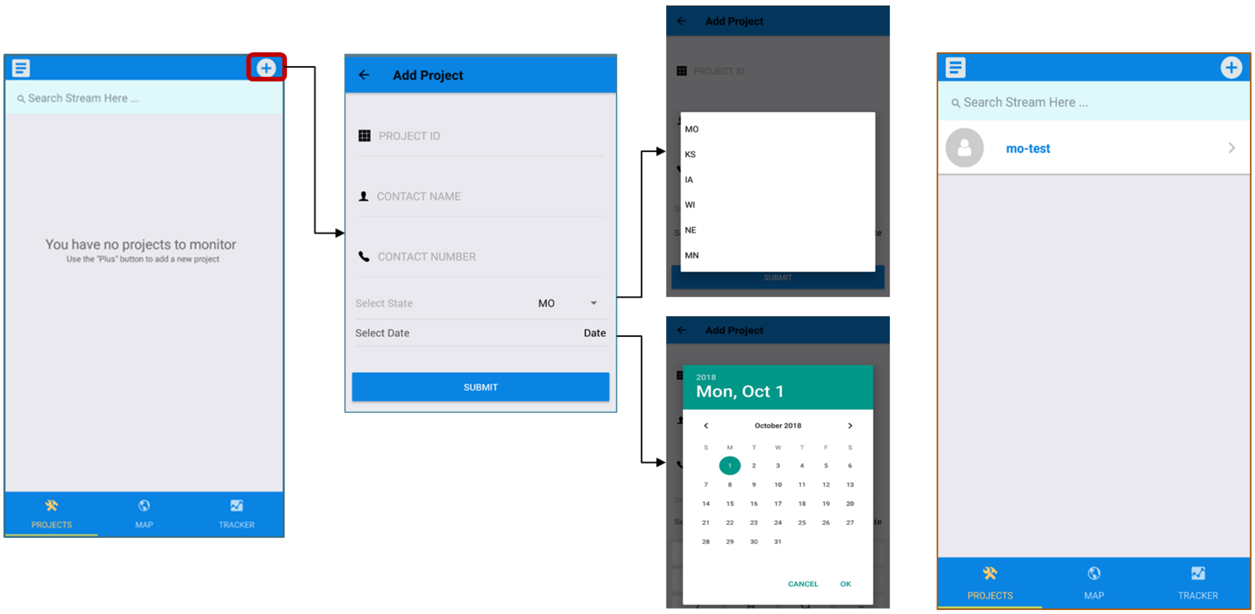


Once registered, users will be automatically signed in every time they open the app. After successful registration, the user is granted privileges to add new projects and track other work-zone project updates within Illinois, Iowa (lead state), Kansas, Missouri, Nebraska, and Wisconsin. SWiZAPP has three main components:

- **Projects** – Renders a profile of all projects added by the user
- **Map View** – Provides a map visualization of all active and inactive work-zone projects
- **Tracker** – Tracks work-zone activity for all work zones by all users

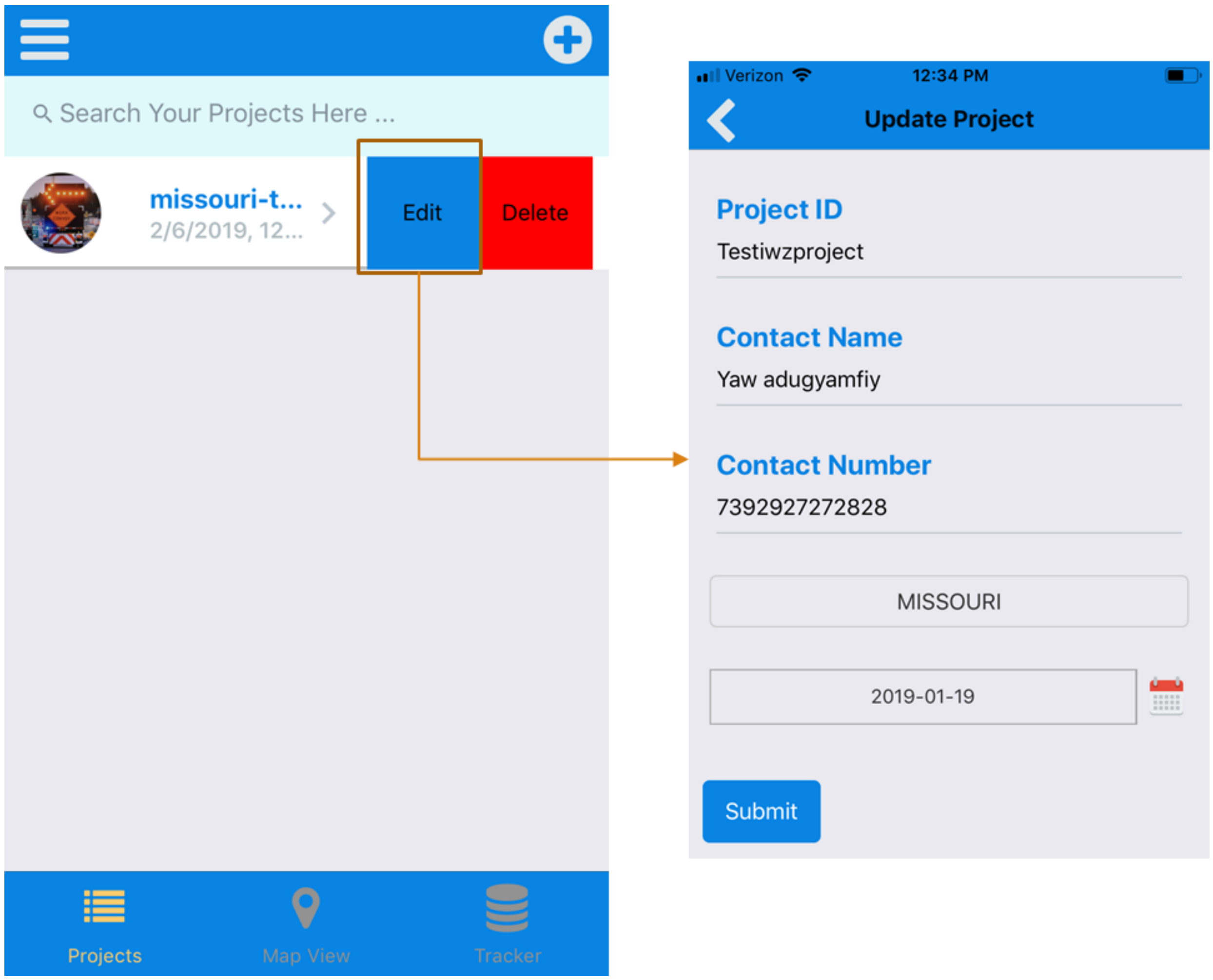
Create a Project

Use the plus-sign icon (in the top right corner) on any of the pages to add a new work-zone project for activity monitoring. The following information is requested to add a project: a project id, contact name, contact number, state, and the start date for work-zone activities. After submission, a profile is created for the project and made available on the Project page for activity monitoring.



Edit/Update Project Information

To edit or delete a project, swipe left on the project profile on the Projects page.



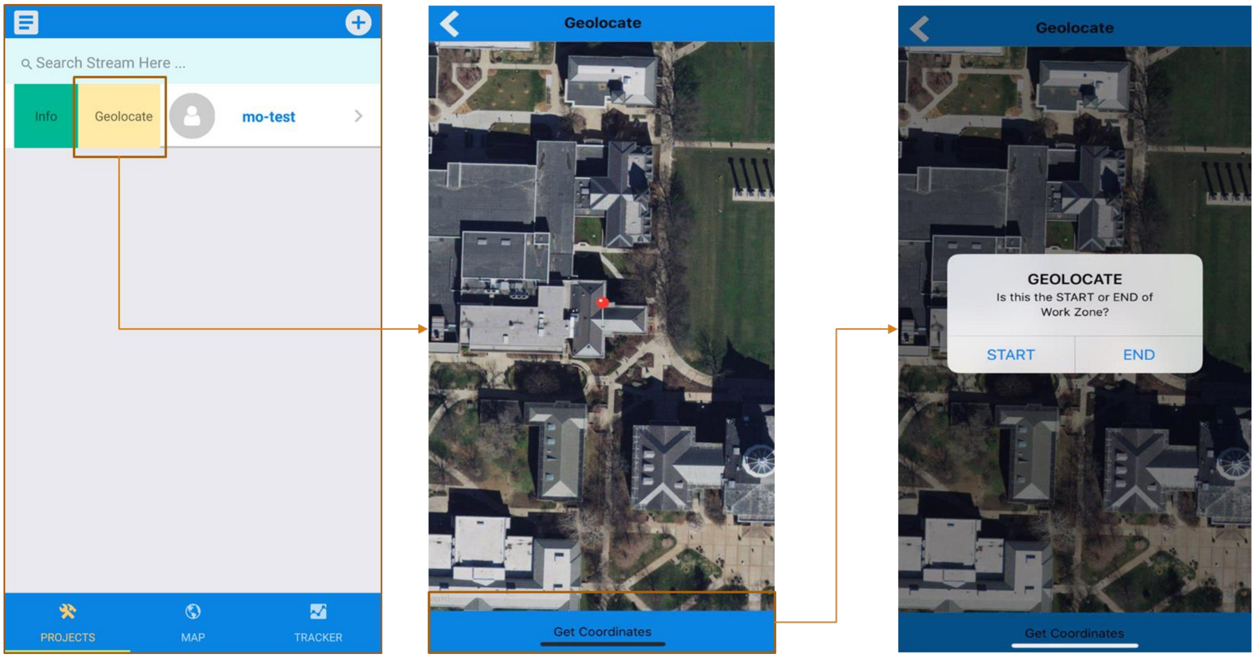
Press the Edit button to fetch all information related to the selected project and display it in a window for editing.

Set Default Project Information

To set default work-zone information, swipe right on the project profile and press the Info button. The user will be navigated to a page to update pertinent information such as number of lanes closed, speed limit, average time on site, etc.

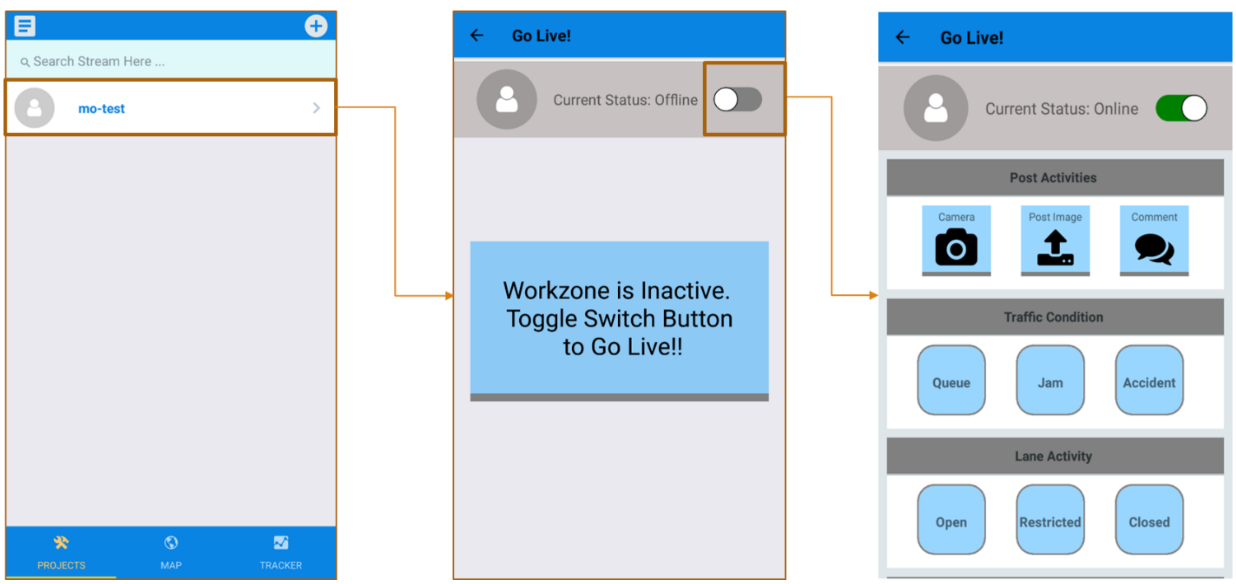
Geolocate a Project

SWiZAPP can automatically geolocate a project by using the phone's on-board GPS. The user must grant access for the app to use this capability. By swiping right on the project profile and selecting the Geolocate icon, one can mark the location of the work zone on the map. Select the Get Coordinates icon on the bottom of the Geolocate page. The user will then be asked to define whether they are entering the start of the work zone or the end. Afterwards, the app will automatically geolocate the work zone based on the location of the user.



Activate/Deactivate a Project

After a project is created, users can activate the project when work-zone activities are being carried out or deactivate it when there is no activity. The screenshots below show how a test project (called “mo-test”) can be activated or deactivated.



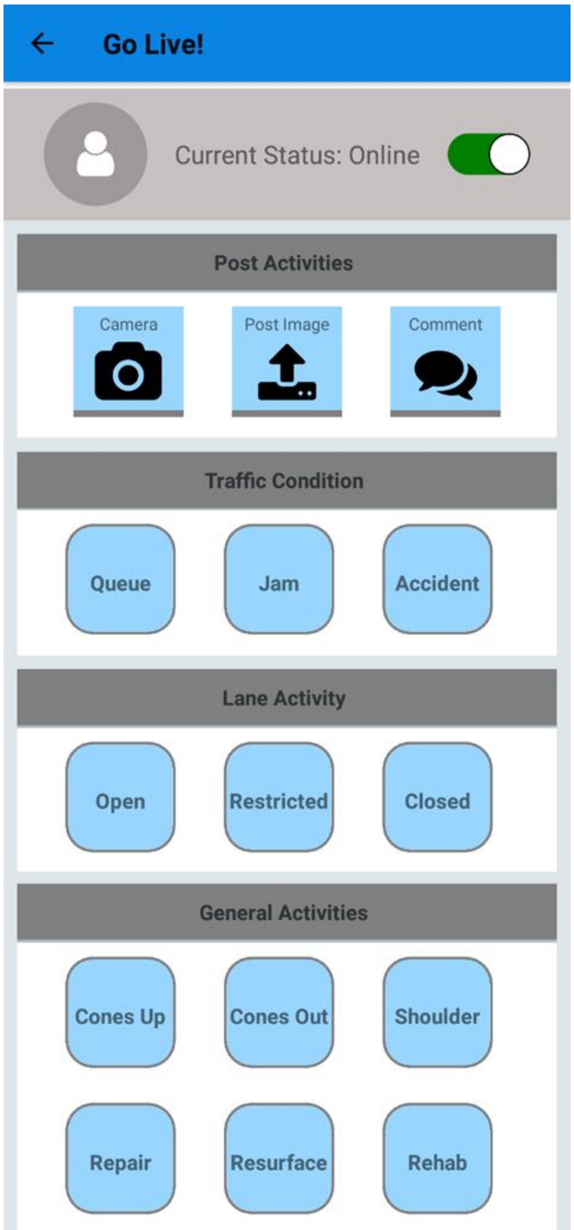
Tap the project name to view options for activating and deactivating the work zone.

The default work zone status is Inactive. A gray toggle button at the top right corner indicates the work zone is inactive.

To activate, press the toggle button. A green toggle button means the work zone is active.

Post Activities: Traffic Condition/Lane Activity

After activating a project, the user is given access to a number of app functionalities that enable them to post real-time work-zone activities such as traffic conditions, lane activities, etc. by the click of a button. The user is also given the ability to take live pictures of incidents, upload the images, and type comments about activities that are not pre-defined in the app.



In order to provide a user-friendly app, SWiZAPP is designed such that most of the information can be added to the project only by selecting one of the predefined categories. Three categories have been defined for traffic conditions:

- Queue
- Jam
- Accident

A user can easily define the lane condition by selecting one of the three lane activity conditions:

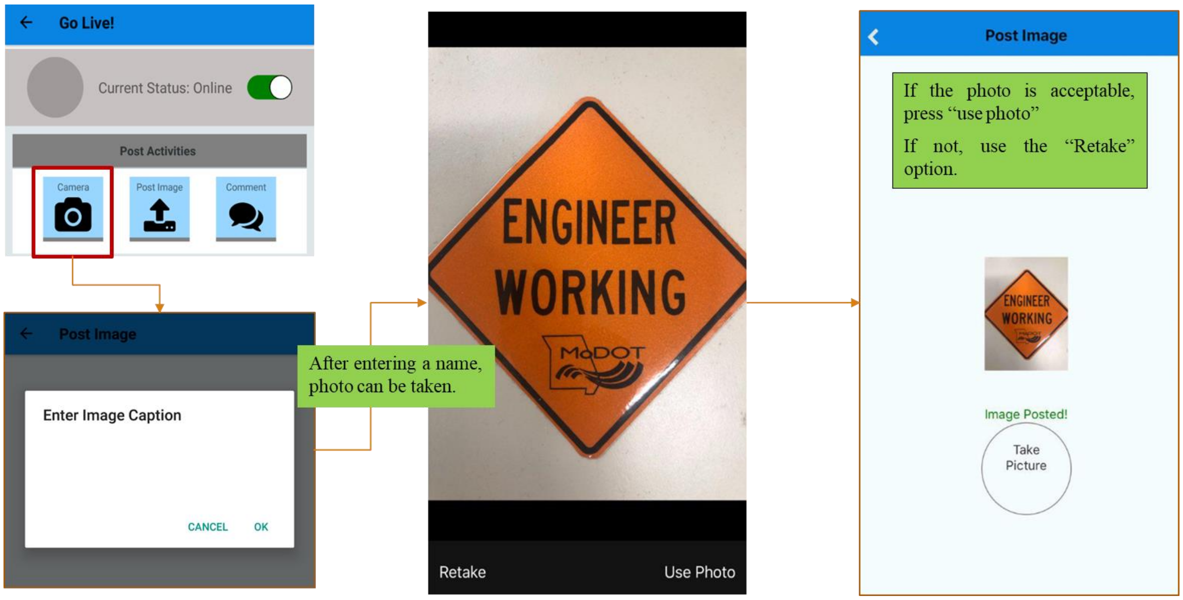
- Open
- Restricted
- Closed

The general activities include six categories:

- Cones Up
- Cones Out
- Shoulder
- Repair
- Resurface
- Rehab

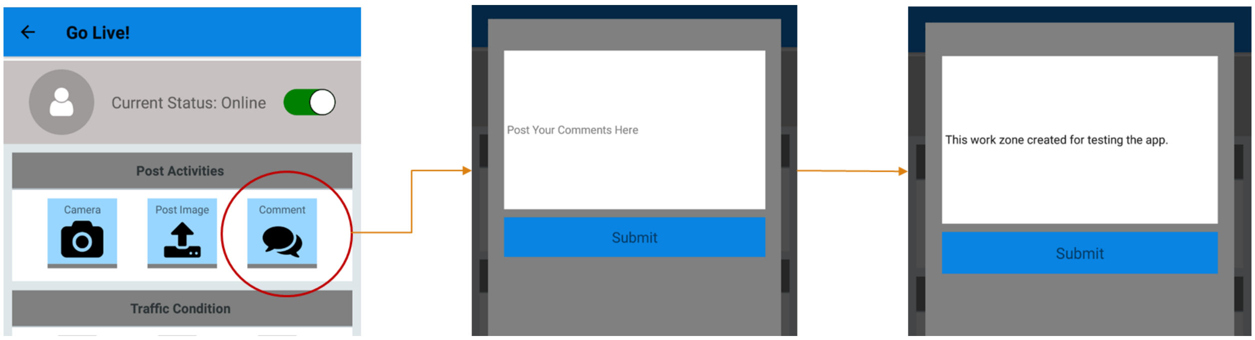
Take and Post Images

The camera icon is used to take a live digital image and post it.



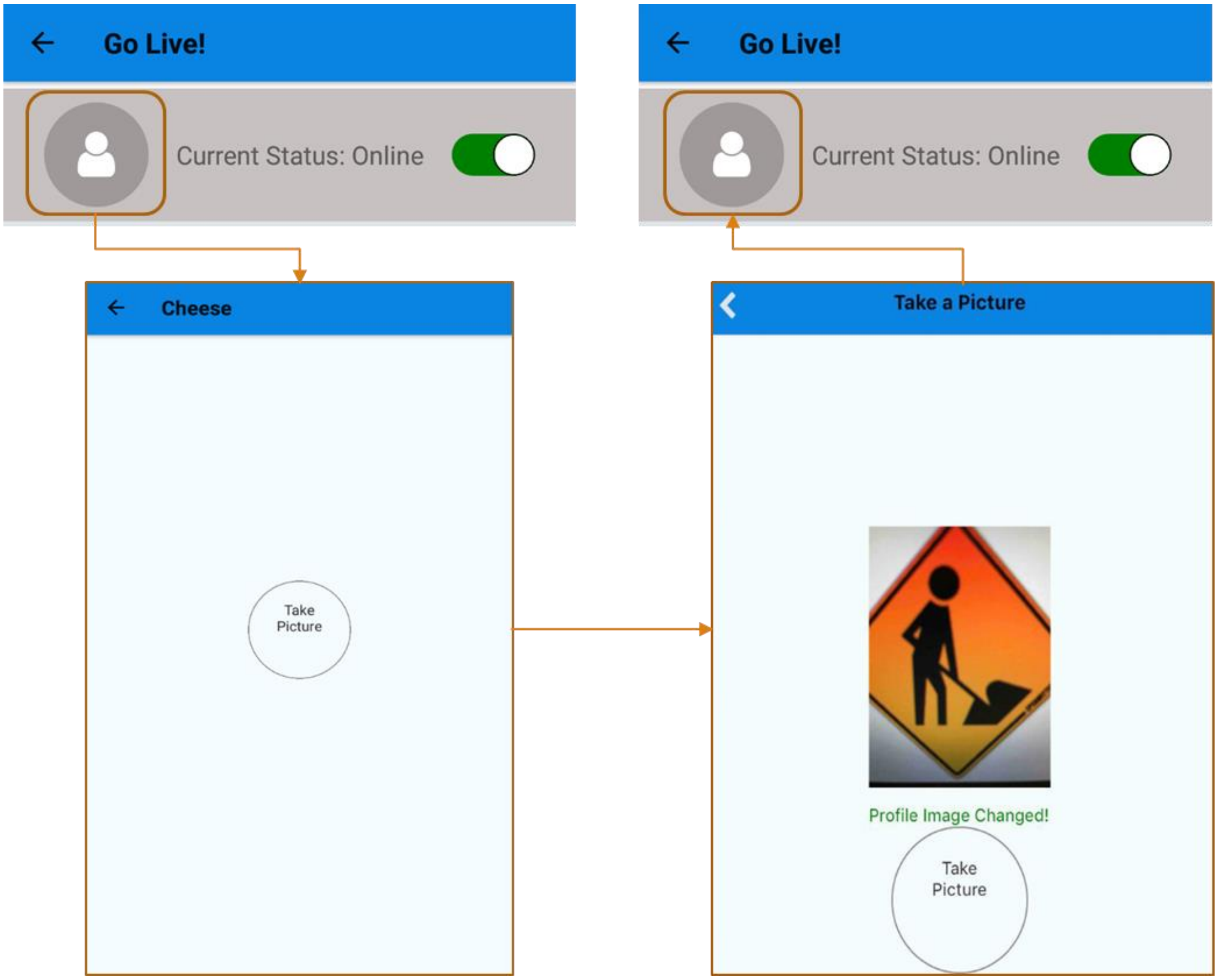
Post Comments

Users can post additional information using the comment icon.



Designate Project Profile Image

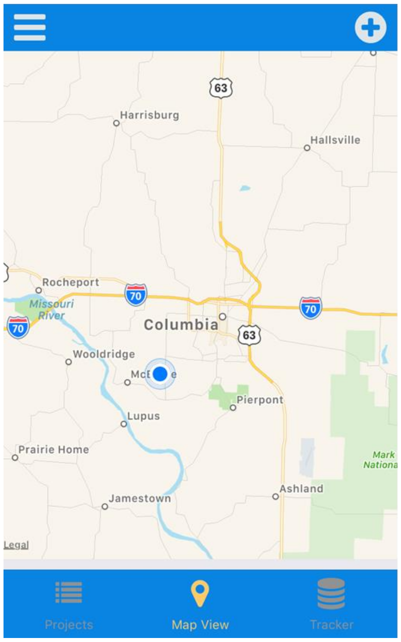
An image can be assigned to the project's profile by clicking on the icon to the left of Current Status.



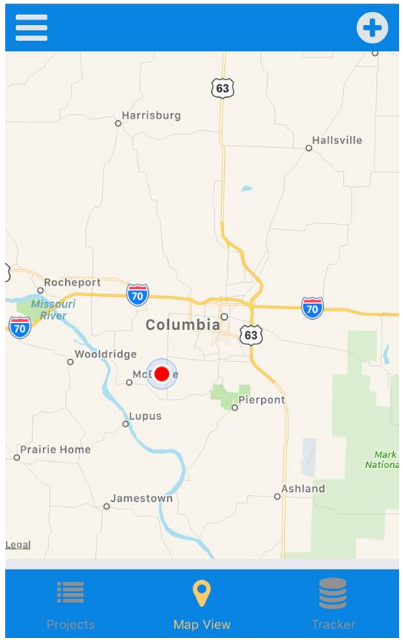
Use the Map Page

The Map page displays the location of all work zones being monitored by the user. To view a project on the map, the project must first be geolocated. The map also displays the most recent activity for each project. The user must click on the respective projects to view this information. Different colors are used to distinguish between active (red) and inactive (blue) work-zone projects.

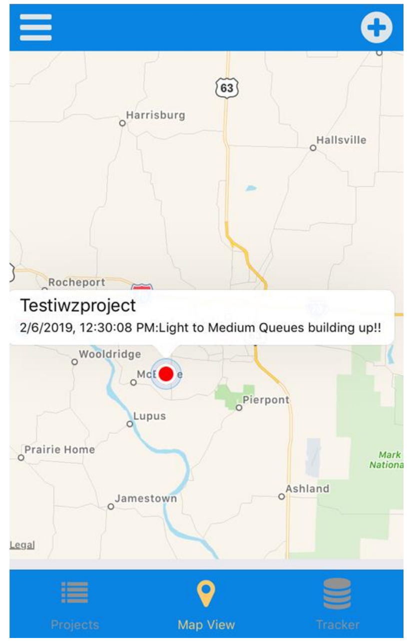
Inactive Work Zone



Active Work Zone



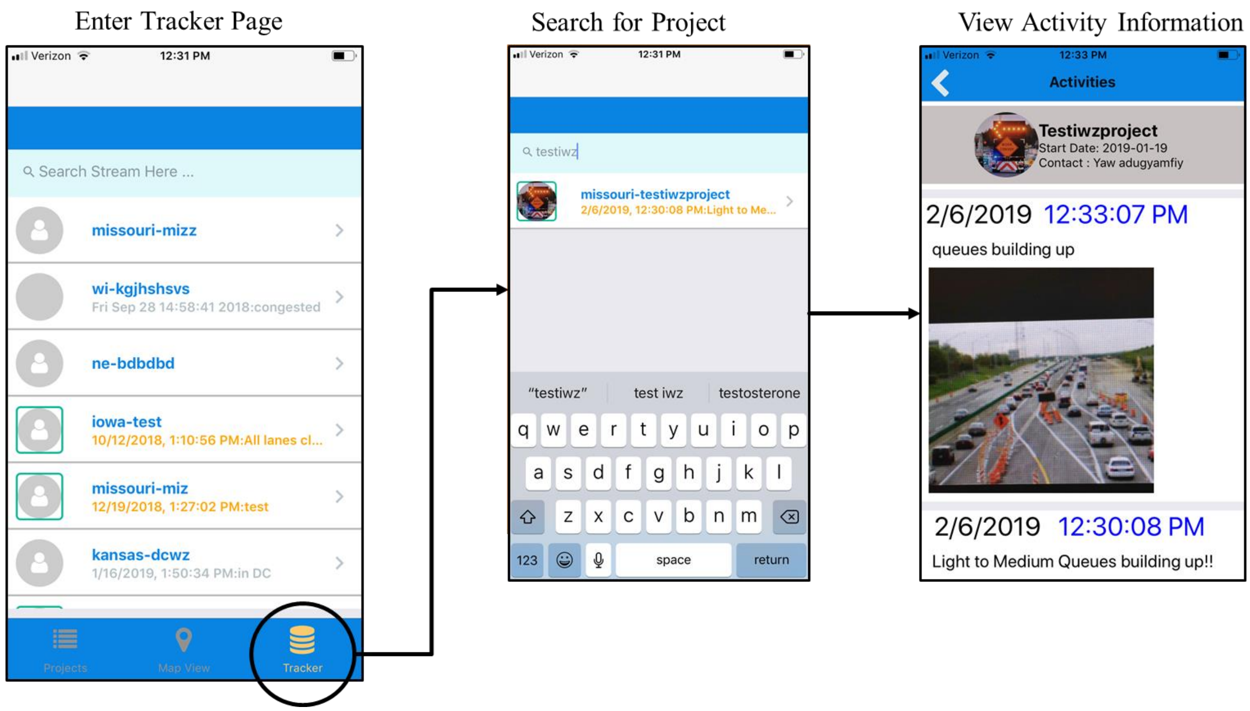
Activity



Use Tracker

With Tracker, the user can view or monitor activities on work zones projects added by other users of SWiZAPP. It is important to note that Tracker can only be used to view real-time and historical work-zone activities. Activities cannot be edited on this page unless the user is the primary owner of the project. To use Tracker:

- Click on Tracker to navigate to a profile page of all work zones stored in SWiZAPP.



- Search for a particular project by typing into the search box either the name, state, or project id. The list of profiles will update to show projects corresponding to the search.

- Click on the project to view all activities tagged by date and time.