

# Maryland Transit Administration Interactive Maintenance Kiosk 2012

created by ScreenPlay InterActive Corp. for the Maryland Transit Administration  
contact: Debra Benkler  
[debra@screenplayinteractive.com](mailto:debra@screenplayinteractive.com)  
<http://bit.ly/lnsIVa>

## DESCRIPTION

In a bold move, Maryland Transit Administration (MTA) envisioned how well-crafted interactive kiosks could motivate railroad maintenance mechanics to do their jobs in a more efficient and effective way. They recognized that to sustain user engagement, the units had to be special, so they commissioned ScreenPlay InterActive (SPIA) to build custom software that would transform the ruggedized 46" touch screen units into iPad-like devices.

Not so simple though: the kiosks were required to use circa 2001 Windows XP as the operating system – a far cry from iPad's iOS. Plus, mixing existing Windows based third party applications with new custom applications meant that XP itself had to be part of one fully blended OS/application UI!

## PURPOSE

The interactive kiosks would resolve inefficiencies in information utilization and internal communications, by:

- consolidating needed information into one device, accessible in the shop
- eliminating paper forms, personal notes/drawings, and undocumented repair methods
- easing communications between departments and locations
- simplifying the task of retrieving/reporting data
- integrating existing and new custom applications into one cohesive interface
- inspiring mechanics to use the information available to them

## USER GOALS

The interface would be predicated upon simplicity, intuitiveness, and a non-Windows experience, to encourage daily usage, by:

1. incorporating tablet-style touch actions
2. minimizing the number of touch interactions to access anything (2 max!)
3. unifying the O/S and custom applications to make the user feel it's one specialized tool and not a typical “computer” environment
4. eliminating user ID's and passwords while maintaining security

## USER GOALS : 1. incorporating tablet-style touch actions

### **The Challenge:**

The touch glass used on the kiosks was single-touch and incapable of multi-touch or gestures.

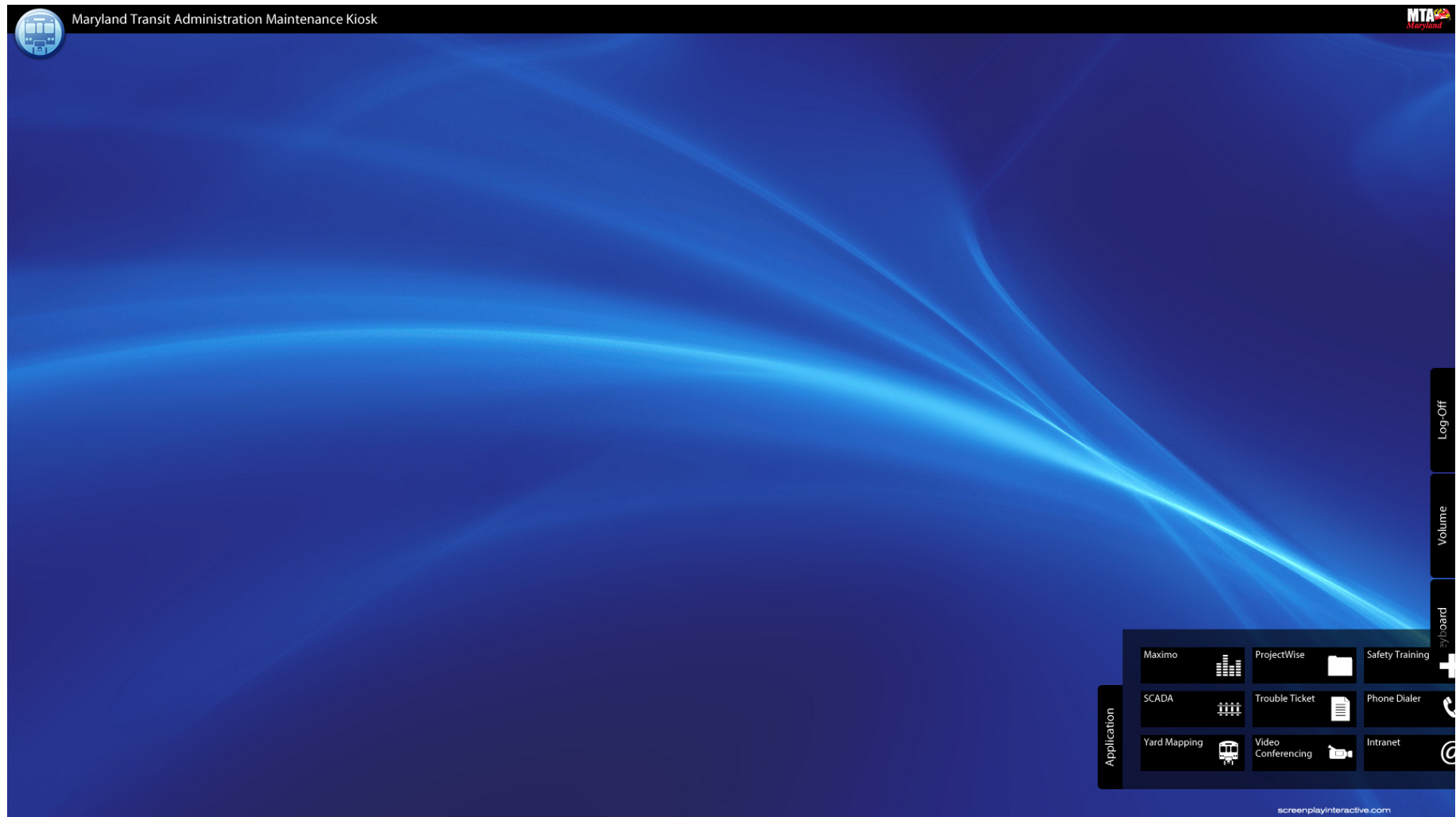
### **The Solution:**

Careful study of the iPad and mobile phones revealed that despite the perception of two fingered gesturing being a dominant action, most gestures were flick, drag, or button touch.

Part clean U/I design and part U/X “gesture impersonation”, a four tab main screen interface was conceived so that the tabs and their off-screen menus would be flicked out with a single digit, glide to center screen, and end with a little easing action.

*(See image following page)*

## USER GOALS : 1. incorporating tablet style-touch actions



Application menu gliding to center screen.

## USER GOALS : 2. minimizing the number of touch interactions

### **The Challenge:**

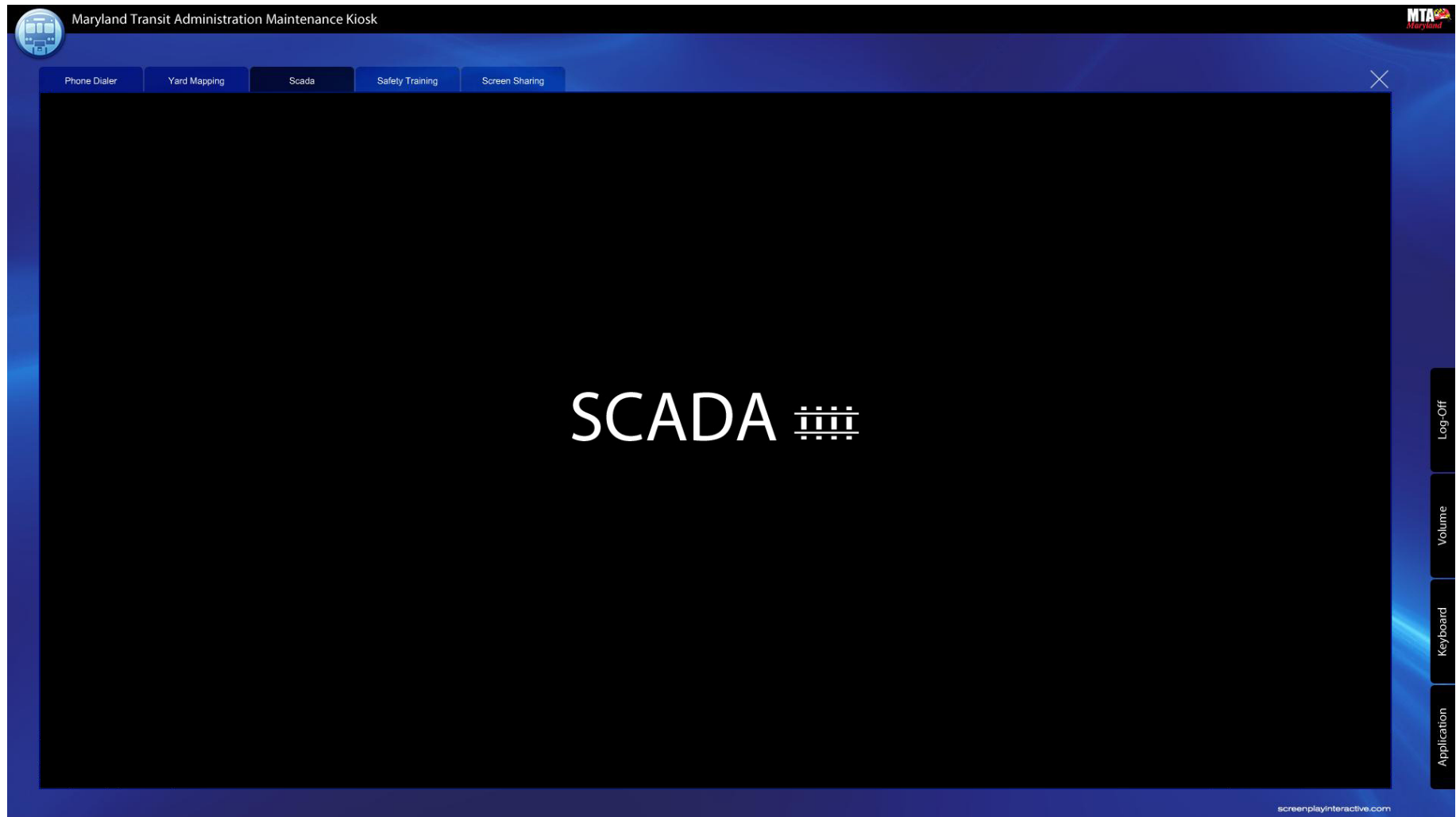
An effective UX/UI implementation is an effortless one that simplifies/minimizes interactions, but kiosks and Windows applications largely run counter to that thinking.

### **The Solution:**

Working through data gleaned at client meetings regarding needed applications, features, and functionality, the UX/UI requirements were honed and multiple design treatments created. With logic and simplicity as the driving factors, the concepts were paired down to a very open-screen, tab metaphor. The design would minimize touches in any state of use, such as launching applications (2 touches) and switching between/closing applications (1 touch), w/o sacrificing required screen real estate.

*(See image following page)*

## USER GOALS : 2. minimizing the number of touch interactions



One touch to switch between open applications.



## USER GOALS : 3. unifying the O/S and custom applications

### **The Challenge:**

Users would only care about ease-of-use. They wouldn't know and wouldn't care what part was OS and what part was application.

### **The Solution:**

From client feedback at discovery meetings, it was learned that the standard XP interface and the ability to get outside of user applications could never be exposed! That triggered a decision to extensively rebuild Windows XP with a SPIA designed UI, making the OS the primary user application. In that way, third party programs could seamlessly meld with the custom apps SPIA was building for the project, blurring the OS/application model.

Outside of the development team and a handful of client people, no one knows that they are using Windows XP or exactly where it or applications begin/end.

*(See image following page)*

## USER GOALS : 3. unifying the O/S and custom applications

The screenshot displays the 'Maryland Transit Administration Maintenance Kiosk' interface. At the top left is a bus icon, and at the top right is the 'MTA Maryland' logo. A 'Trouble Ticket' tab is active, with a close button (X) in the top right corner. Below the tab is a text box: 'To complete a Trouble Ticket, select the appropriate parameters from the menus. Upon pressing submit, an email will be distributed to all departments associated with the trouble issue. However, if this is an emergency, please contact responsible personnel directly.'

The main content area contains four vertical panels:

- Trouble Ticket Request**: A list of input fields with the following values: 5/1/2012, 3:35 pm, Ken Lonyal, Janitorial Services, Janitorial Services, Building Type, Location, and Specific Location.
- Department**: A list of departments with 'Janitorial Services' highlighted in red. The list includes: Access Control, Safety, Police, Facilities, Fare Collection, Systems, Janitorial Services, Administration, Maintenance of Way, Metro Transportation, Rail Car, Engineering, Elevator/ Escalator, and Radio/ Communication.
- Trouble**: A single selection box with 'Janitorial Services' highlighted in red.
- Building Type**: A list of building types with 'Station' highlighted in red. The list includes: Station, Annex Building, Administration Building, and Vent Shaft/ Pumping Station. A 'Back' button is located at the bottom of this panel.

On the right side of the screen, there is a vertical toolbar with four buttons: Log Off, Volume, Keyboard, and Application. At the bottom right corner, the text 'screenplayinteractive.com' is visible.

Trouble Ticket application being completed by a user.

## USER GOALS : 4. eliminating user ID's and passwords

### **The Challenge:**

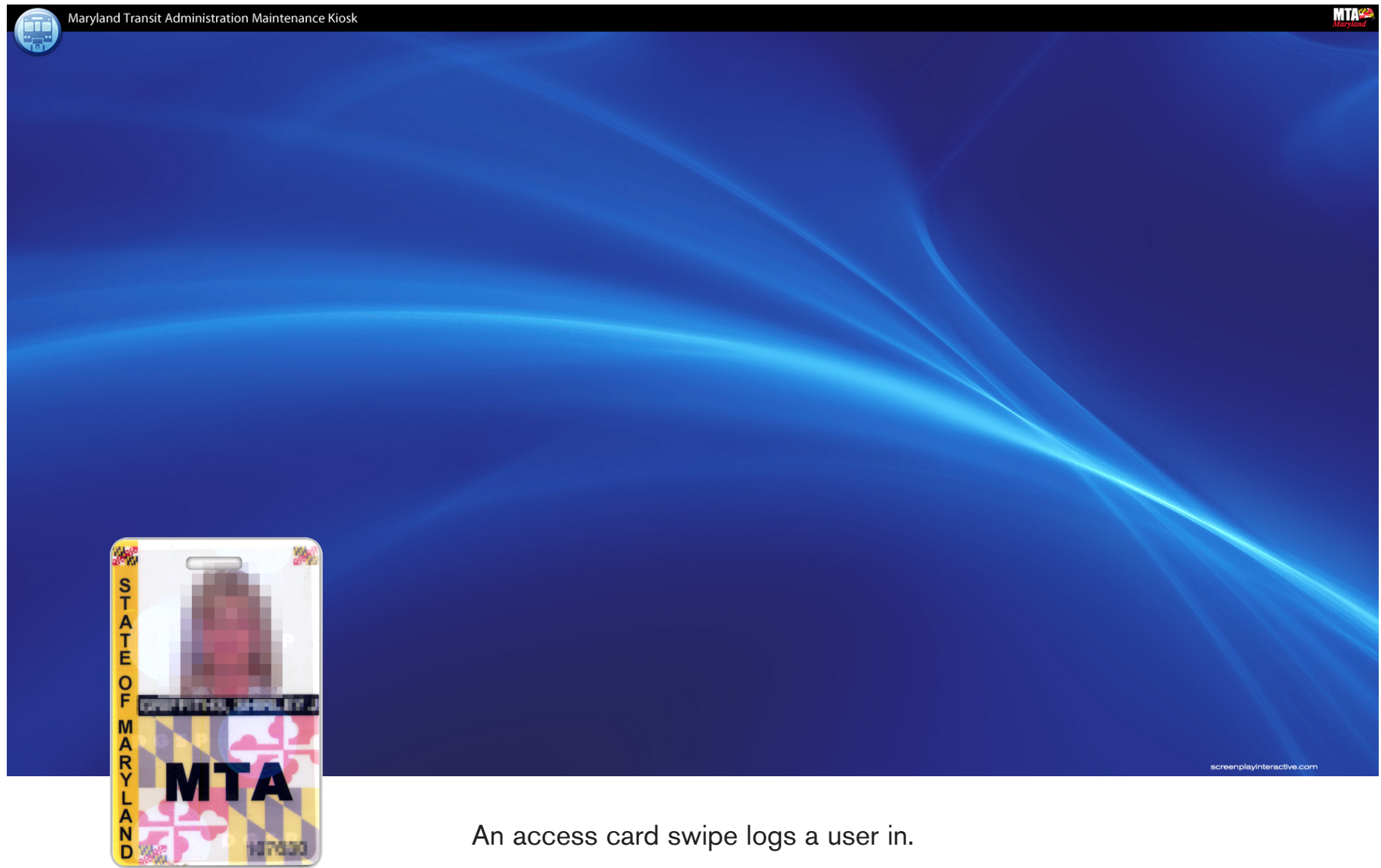
People find it difficult to remember log-in info and passwords for each unique application that they use.

### **The Solution:**

Everyone at the repair facility had an RFID building access card on them at all times. It became the perfect tool to relieve password memorization woes by installing matching readers on the kiosk and mapping their card ID to ID's and passwords in software applications. If they could get into the building they could access any application they wanted to, without even remembering their name. Simply walk up to a kiosk, present a card and you're in. A train horn sounds and the side tabs slide into standby position to let users know they're ready to work.

*(See image following page)*

## USER GOALS : 4. eliminating user ID's and passwords



An access card swipe logs a user in.

The key to the success of this project was mastering contextual design. Visual appeal was a given, as was ease-of-use. Understanding what information a mechanic needed to repair a train, the flaws in the existing paper/PC data methods, and objections to “yet another piece of software”, was the focus.

## Discovery

- on-site visit to a train yard and repair facility
- meetings with management and mechanics
- review of paper documents and existing software apps
- development of a wish list of features and new applications

## Preliminary Design

- study of iPad and mobile device touch gestures
- ideation of various UI treatments and usability cases
- refinement of UX/UI key factors
- integration of special features/considerations
- break-out of features: OS vs. custom applications
- client review/feedback iteration loop

### Development

- final touch actions determined/perfected
- final UI creation
- coding methodology determined from OS/API requirements/limitations
- hardware/software interaction finalized
- client approval

### Build-out/Testing

- detailed development specs written
- code, code, code
- segment by segment code debugging
- usability testing/tweaking by development team and random test subjects
- continued testing/monitoring

Creative license with peripheral devices allowed SPIA to push the envelope on the application's U/X feature set:

- video camera triples as a presence sensor and log-in device through facial detection and recognition, when not used for video conferencing
- RFID building access cards provide an alternative to facial recognition to activate the kiosks and eliminate ID's and passwords in applications

Additionally:

- even “back-end” OS functions like log-off and sound/mic controls never break from the UI
- upholding developer tradition, an Easter Egg is there to be found by the adventurous

## APPLICATIONS

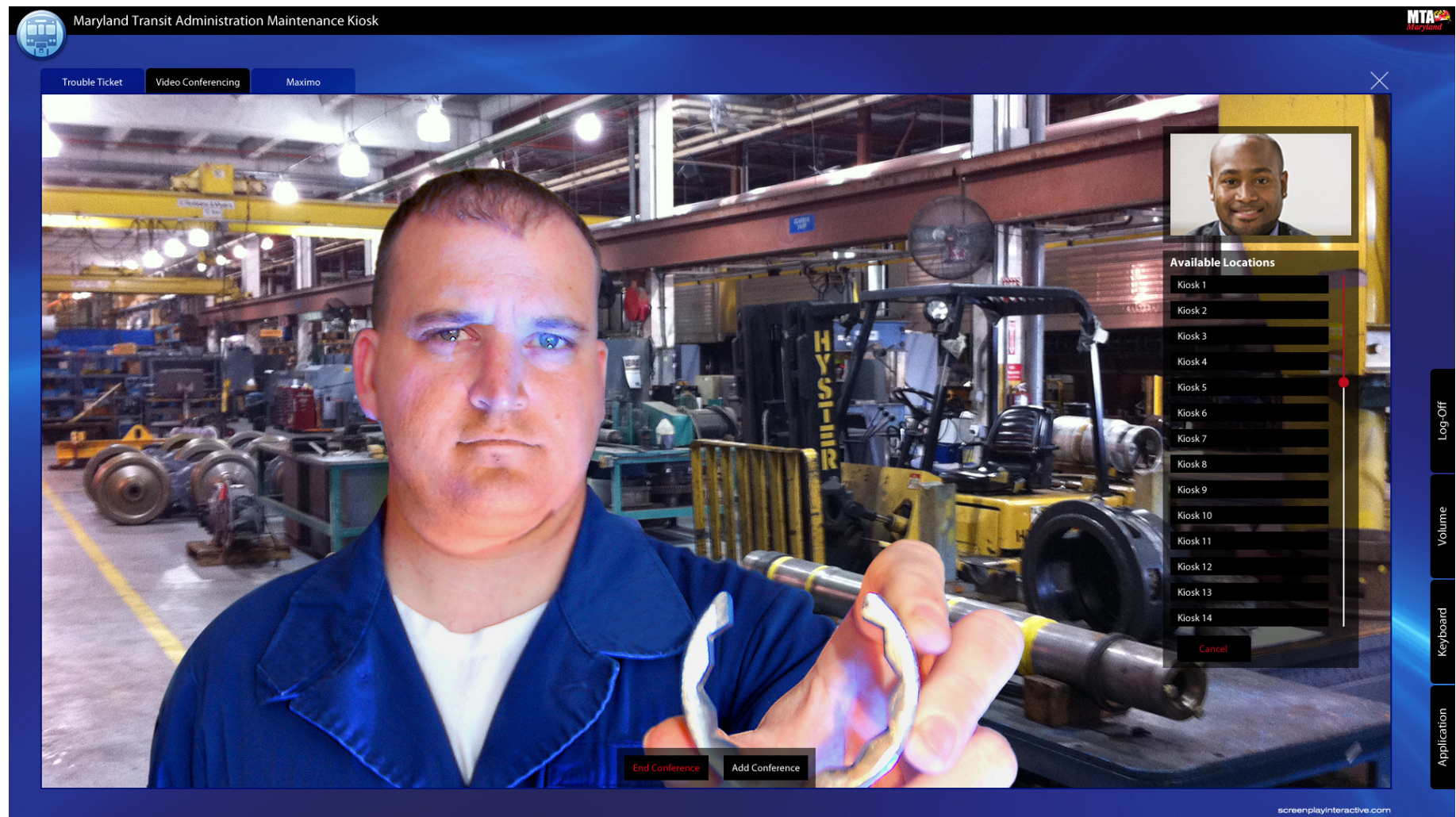
In addition to incorporating existing third party industry specific software, this project included the following applications custom built for the client by SPIA:

- **SPIA version of Windows XP:** functions as both OS and main UI
- **Facial Recognition:** a user optional method of logging into a kiosk
- **Presence Sensing:** determines if a user is still using a kiosk, else logs them out
- **Video Conferencing:** a custom LAN multi-user video conferencing system
- **Trouble Ticket:** an intelligent data submission application that uses no forms
- **Phone Dialer:** an on-screen dialer enabling hands free calls to pre-populated numbers
- **Screen Sharing:** access to other user's kiosk environment for training/support
- **Universal On-Screen Keyboard:** fully draggable, available to all applications

*(Selected application images following)*

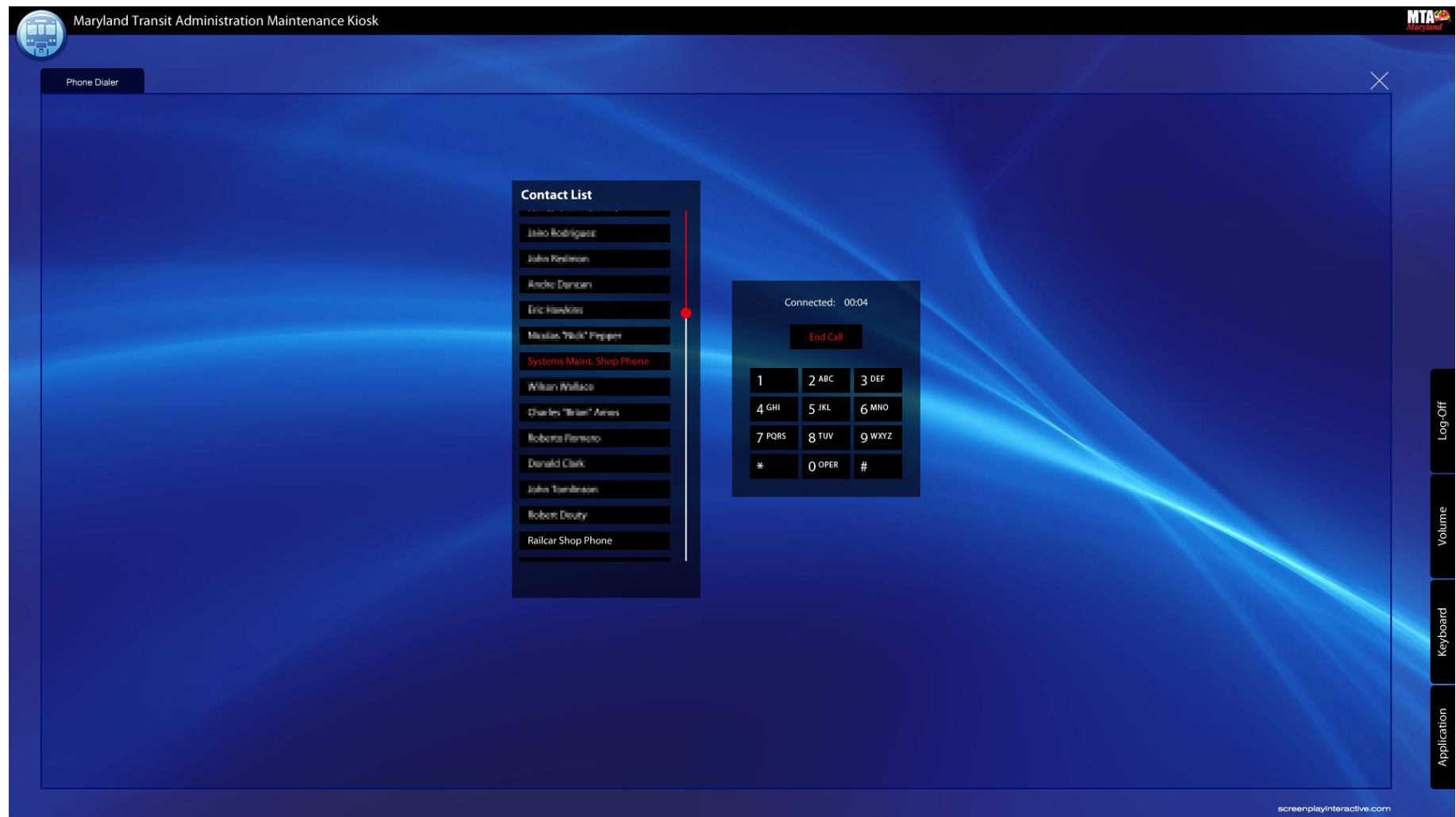


## APPLICATIONS continued



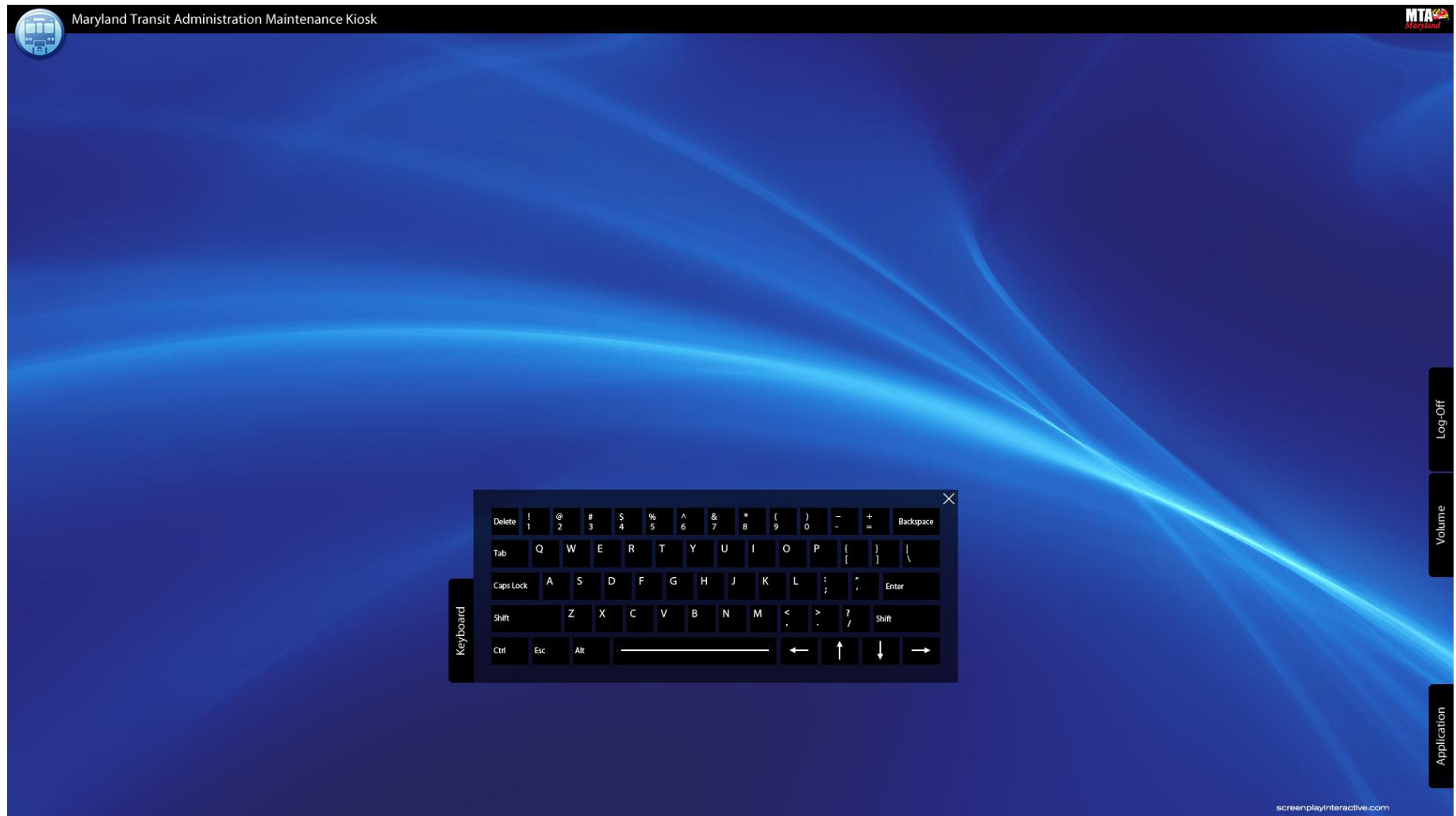
A two person video conference.

# APPLICATIONS continued



Phone Dialer application after connecting.

## APPLICATIONS continued



Draggable keyboard just launched.

## WHY THIS PROJECT IS A WINNER

This project evolved from a largely unsupported concept within the client's organization to one that more and more personnel have bought into, as they've been exposed to it and its usefulness. It alleviated user concerns that more software meant more training – none required, none provided. That's the best outcome a developer can wish for.

Additionally, it has shifted the kiosk paradigm in a number of significant ways:

- familiar, tablet style “gesture” interactions on a limited capability device
- seamless application and OS interaction without tampering/security concerns
- a utilitarian device and unsupported use case reinvigorated through good design/execution