Team Quicken Loans: Mobile RFID Inventory Tracking System

Administrator Manual

TABLE OF CONTENTS

WHAT'S THE MOBILE RFID INVENTORY TRACKING SYSTEM?

ABOUT

The RFID mobile device tracking system automates the process of checking mobile devices in and out of a secure lockbox. The mobile device testing group at Quicken Loans had been using manual logging to keep track of who had what mobile device checked out at any given time. This caused issues such as devices missing for extended periods of time, thus the team decided that an automated process would be more desirable.

The RFID mobile device tracking system utilizes RFID tagging of the mobile devices in conjunction with in-place RFID scanners and employee authentication to automate the process. Each device will have a unique RFID tag affixed to it that will identify the device. The in-place RFID scanner is responsible for detecting which tagged devices are present in the cabinet. Lastly, employees may access the storage area by authenticating themselves with their company assigned badge.

A front-end web application replaces the manual logging. This application allows team members to view current inventory and check-out history. The front-end also has administrator interfaces to notify employees in possession of checked-out devices, as well as providing management functionality for devices and information pertaining to the devices.

YOU SHOULD KNOW

The RFID Mobile Inventory Tracking System is configured to be simple to use. For most users, the process should be highly automated. Scanning your badge at the cabinet before and after exchanging devices with the cabinet will automatically go through the entire check-in/check-out process. Users never need to touch the bundled web application, but it is available for checking device statuses beforehand.

For administrators, the experience should be similar. The model for handling devices isn't too complicated, and has a simple UI in the web application.

How do I use the system?

Example Use Case #1 For Administrator

Sally Admin is in charge of managing the devices in the Mobile Device Cabinet. She receives a shipment of new devices for testing purposes. Sally logs into the front-end web application and selects the option to add a new device. She enters the information for each new device on the web application and confirms that each device was added. She then tags each device with a new RFID tag, scans her ID to open the cabinet, and places the devices inside. When the cabinet closes, the RFID reader reads the new devices and marks them as checked in.

Example Use Case #2 for Administrator

Tom Boss is also an administrator in charge of tracking the mobile devices in the cabinet. He receives a complaint from Norah Smith in the mobile device testing team that the mobile device she needs to test on has been missing for a while. When Tom logs into the front-end application to view the device inventory, he confirms that the Samsung Galaxy S3 running Android version 4.3 that Norah Smith is requesting has been checked out by Bob Smith for a year. Tom chooses the administrator option to notify and selects the option to notify the current possessor. The front-end notifies the back-end web service, and the back-end web service sends out an email to Bob Smith notifying him that the device is needed and that he has had it for too long. Bob Smith receives the email and returns the phone the next day.

Tour – Web Application

The tracking system comes bundled with a web application to make viewing device statuses easy. The application should be hosted internally, and should be available where a user works with company-owned devices. In the following figures, note that the application host will vary.

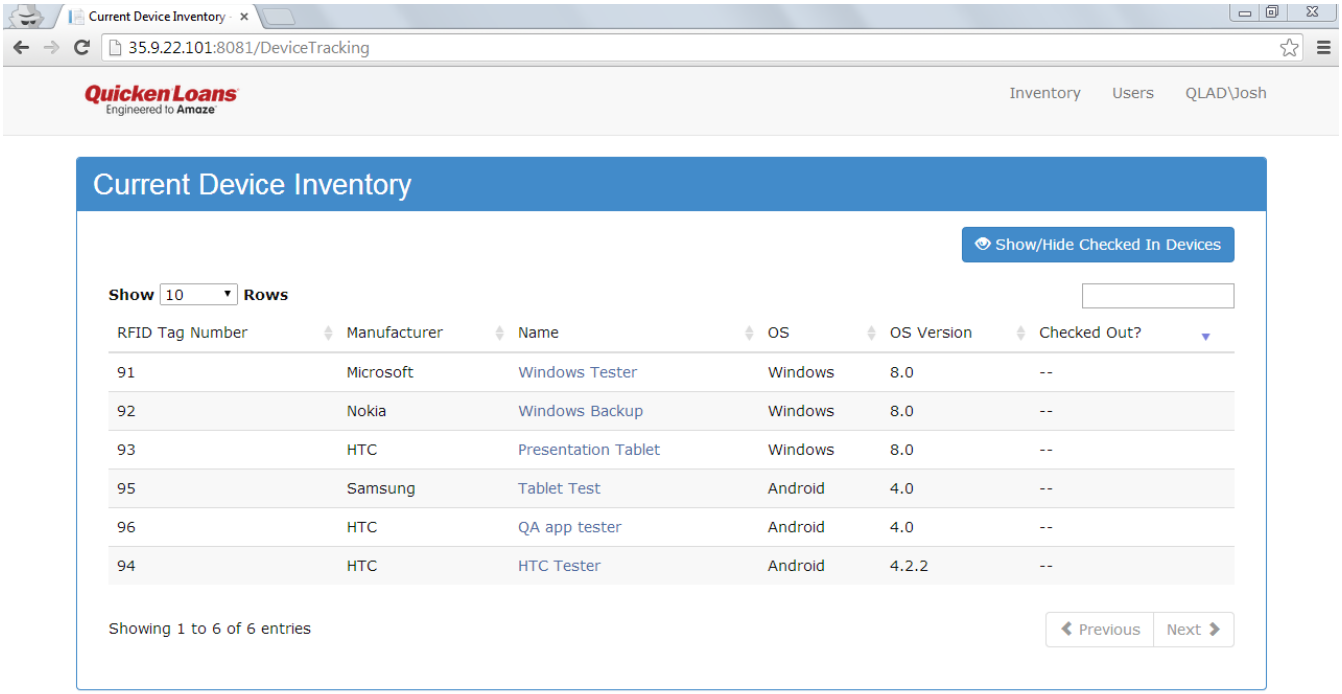

Figure 1: Web Application Device Inventory Page

As shown in figure 1, the web application displays the list of all registered devices with their current status and information. Administrators can delete devices off of this main page. Delete devices when you're no longer interested in tracking them with the system.

When devices are checked out, the user to whom the device is checked out and the time they checked that device out are shown. This can be used to know the whereabouts of a device.

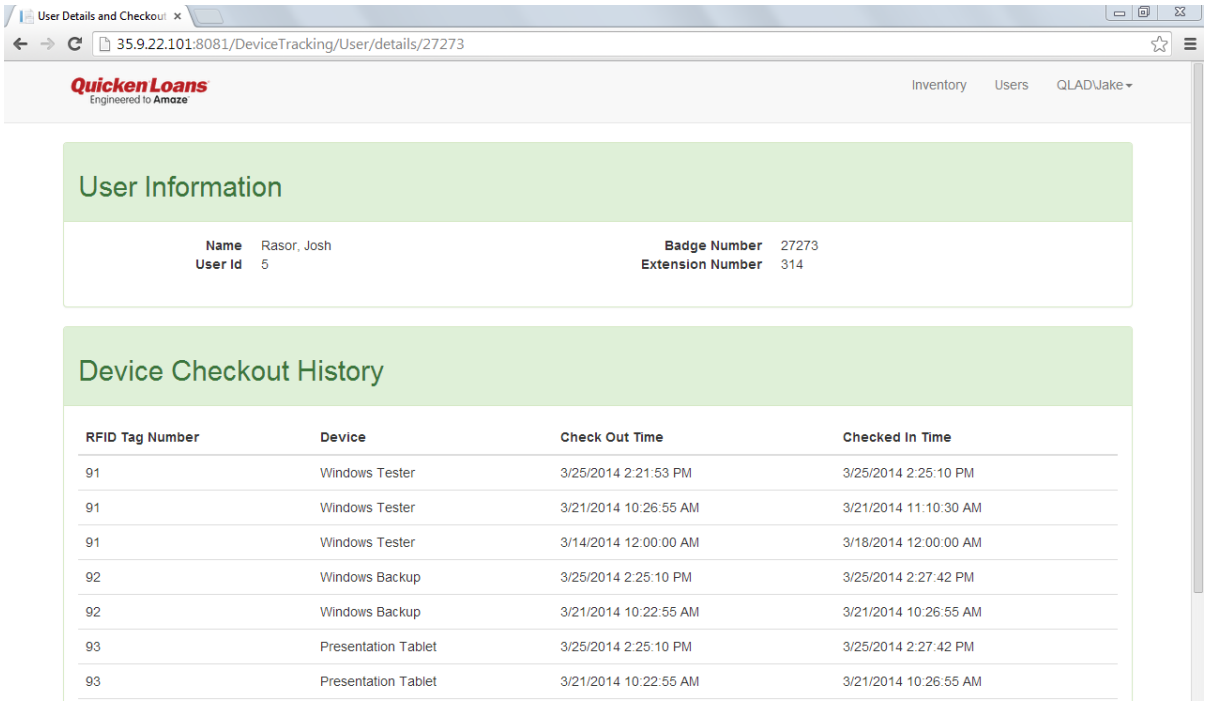

Figure 2: User History Page

Figure 2 shows the web application's view of all known sessions in which that user checked a device out and checked it back in. Each user's history is accessible and recorded. The web service provides the list of all known users and their histories. This may be useful to see if a device will likely be inaccessible when a user has historically used that device frequently.

As an administrator, you'll be able to notify users of all devices they have checked out. The web service will send an email containing information about all the devices a user has, and request they be returned to the cabinet.

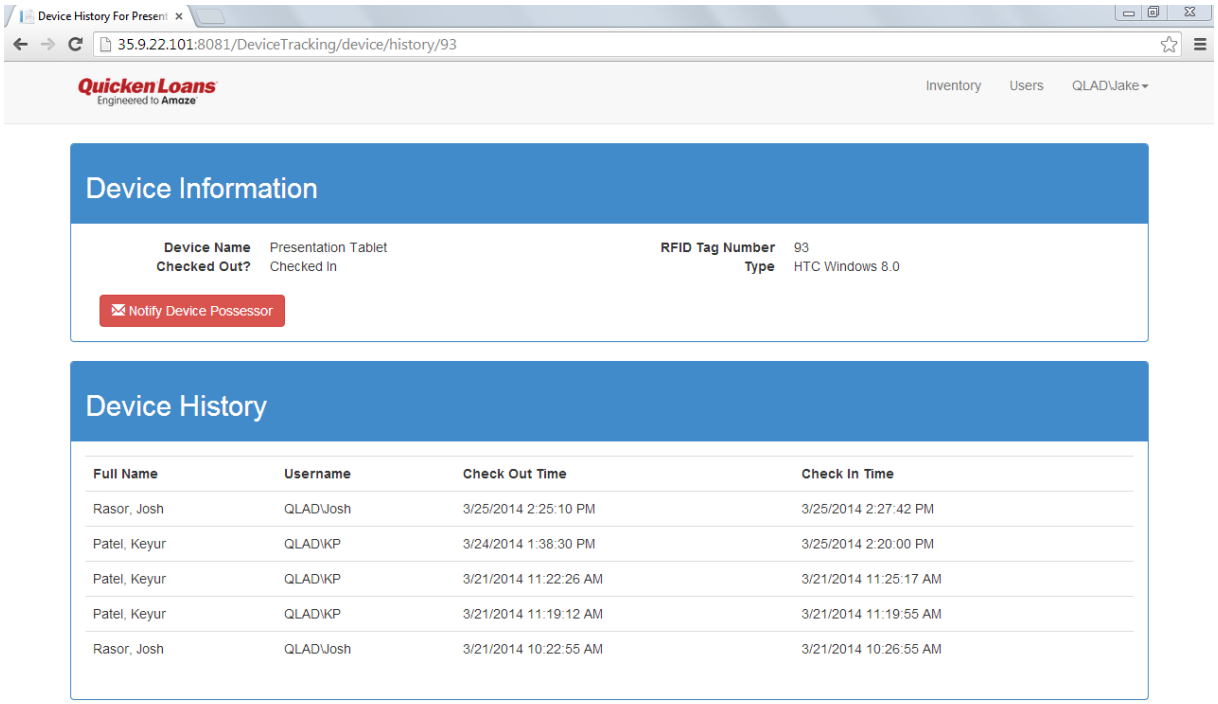

Figure 3: Device History Page

Figure 3 displays the web application's device history view. A device's history consists of all known sessions in which any user checked that device out and later checked it back in. Each device has its corresponding history tracked and can be viewed via the web application. This may be useful to see a typical availability schedule of a device.

As an administrator, you can notify by device as well. By clicking the button on the web application, the web service will send an email to the current device's possessor, requesting the device be returned to the cabinet.

To manipulate devices and their information, the web application comes with administrator functionality restricted with administrator access.

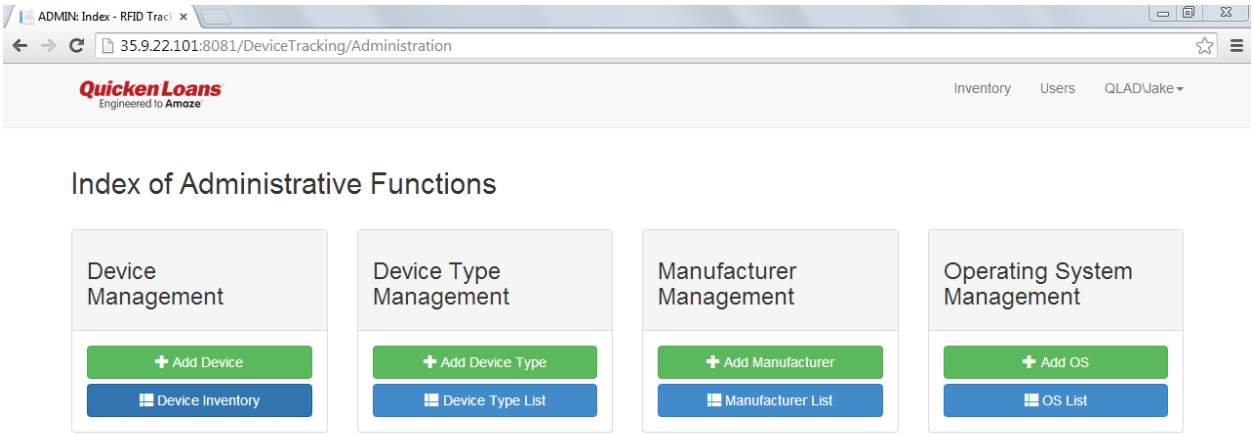

Figure 4: The landing page for administrator functions

Figure 4 displays the landing page for administrator functions. From this page, you may manipulate information of devices you wish to track. Devices require a device type, RFID tag number, and name. Administrators can manage devices, device types, manufacturers, and operating systems. A device has a device type; in turn, a device type has a manufacturer and operating system.

STEP-BY-STEP

The following section will detail all the steps needed to check devices in and out of the cabinet in a similar case to the sample user case. Note that the below process encompasses all steps you'll need to both take devices from and return them to the cabinet.

1. Locate the badge scanner, which should be attached to a computer nearby your device cabinet.
2. Scan your badge over the scanner – using RFID technology, the scanner should easily be able to pick up your badge.
3. Depending on the snapshot length configured in the Cabinet Guard application, you may have to wait a moment before grabbing devices. Usually, this delay should be very short or non-existent.
4. Grab from the cabinet any devices you want to work with. If you have any devices you want to return, replace them back into the cabinet.
5. Scan your badge on the badge scanner. This will let the system know you're done, and that it can figure out what devices have moved.

SETUP AND DEVELOPMENT

SETUP

This system is intended to be deployed to a Microsoft Server with .NET framework 4.5.1 or above.

Refer to the <appsettings> area of the web.config files of the API and MVC projects to change the following key/value pairs:

1. Smtpserver/port – the server and port from which notification emails will be sent
2. Sender Email/password – the credentials the system will use to send emails out
3. Admin email – the email to which additional administrator notifications will be sent
4. API password – the shared secret to be used by the API and other applications with which it will communicate.

The cabinet guard runs on Windows machines. The machine running cabinet guard needs to be able to access the RFID scanner over the network, and can be targeted via the XML file in cabinet guard's own UI. The badge scanner should be connected via USB to the computer running cabinet guard to receive its input.

The cabinet guard can be configured by hand through an XML file named config.xml that it expects to be in the same directory as the application itself. This file also contains credentials for authenticating the application to the API.

To generate the database model used by our system, run the SQL script bundled with the source code to create the model.

Next, change the web.config file in the MVC application to target the location at which the API is hosted, and target this url in the config.xml file as well.

API AUTHENTICATION

The API uses basic HTTP authentication to secure sensitive API calls. The web.config for the API contains a field for an API shared secret, shared between the API and the front-end web application. The authentication is also notable in the config.xml file for the Cabinet Guard, which contains fields for authentication to the API.

For basic HTTP authentication to be secure, the server should communicate over SSL.

To remove or alter API authentication, alter the custom message handler in the API project.

DEVELOPMENT

To change membership over to Active Directory, change the membership provider in both the MVC and API projects to the Active Directory provider, which is configured in each web.config file. To change the database model correspondingly, remove the ASP.NET membership table in the database. The SQL role provider is bundled with the membership provider, which can be

removed in web.config of the API and MVC projects; the permissions table in the database can be deleted correspondingly.