



**Team Quicken Loans: Mobile RFID Inventory Tracking
System**

Project Plan

Spring 2014

Michigan State University Student Team

Josh Rasor
Jacob Riesser
Keyur Patel

Quicken Loans Professional Team

Jim Brzuch
Patrick Hartford
Sarah Kindinger
Christina Mathes
Bryan Wisner
Breanna Perrett
Channing Pritchett
Xulu Zhang

TABLE OF CONTENTS

Executive Summary	3
Functional Specifications	4
Example Use Case For Employee.....	4
Example Use Case #1 For Administrator.....	4
Example Use Case #2 for Administrator.....	4
Specifications	5
Design Specifications.....	6
Web Application	6
RFID Reader	8
Sequencing	9
Technical Specifications.....	10
System Architecture	10
Badge Scanner.....	10
RFID Scanner	11
Badge Scanner.....	11
Cabinet Guard.....	11
Web Application	11
Web Service.....	11
Database.....	11
Locking Mechanism.....	11
RISKS.....	12
Badge Scanner.....	12
Physical Locking Device.....	12
RFID Reader	12
Web API 2.1	12
Timetable.....	13
Additional Information	15
Point of Contact.....	15
Acknowledgements	15

EXECUTIVE SUMMARY

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 employees 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.

FUNCTIONAL SPECIFICATIONS

The following information regards the functional aspects of the RFID mobile device tracking system.

EXAMPLE USE CASE FOR EMPLOYEE

Bob Jones needs to test some new code on a mobile test device. Bob logs into the web application in order to determine if there is a device that meets his needs in the Mobile Device Cabinet.

He approaches the Mobile Device Cabinet and swipes his security badge using the badge scanner on the cabinet. The system authenticates Bob's security badge, immediately takes a snapshot of all the devices currently in the cabinet using RFID technology, and records this data. The cabinet door then unlocks, and he is free to take any device he chooses. Bob grabs a Samsung Galaxy S3 running Android version 4.3, closes the cabinet door, and scans his badge. The system then takes another snapshot of all devices in the cabinet, and determines which device has been removed. The data is recorded in a database and available to be viewed and reported on via an internal webpage.

When Bob finishes his testing, he returns to the cabinet and swipes his badge to open the cabinet. The system takes a snapshot using RFID technology of all devices currently in the cabinet. Bob places the device back into the cabinet and closes the door. The system takes another snapshot and determines what devices have been placed back into the cabinet, and records the data. The system sends emails to both Bob and the Mobile Development QA team when a device is checked in or out.

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.

SPECIFICATIONS

The Mobile RFID Inventory Tracking System conforms to the functional specifications described herein.

1. The system automatically checks devices in or out for employees; employees interact with the system only by scanning their identification badge.
2. The system authenticates users attempting to access the Mobile Device Cabinet.
3. The system can determine the states of all registered devices at any time: to whom devices have been checked out or if devices are physically present in the Mobile Device Cabinet.
4. A web application is available to review current inventory information, device history, and user history.
5. Furthermore, the web application can manage inventory information, such as registering new mobile devices.
6. For mobile devices, the system maintains the RFID tag ID, device type, device name, device manufacturer, device operating system, and device operating system version.
7. For users, the system maintains that user's security badge number, common ID, first name, last name, and extension number.
8. When users check devices in or out, the system will maintain the device check in and out date and time.
9. The system will send email to both the user and the Mobile Development Quality Assurance team providing information regarding the date and time the device was checked in or out, and details about which devices were checked in or out.

DESIGN SPECIFICATIONS

The following are the design specifications. These include front-end mock-ups and a systems diagram.

WEB APPLICATION

The web application is built on the .NET MVC framework and is styled using the Twitter Bootstrap. The Quicken Loans styling guide provides visual consistency with other Quicken Loans products and solutions.

The web application contains pages for device inventory, device history, and user history, as shown in the below three figures.

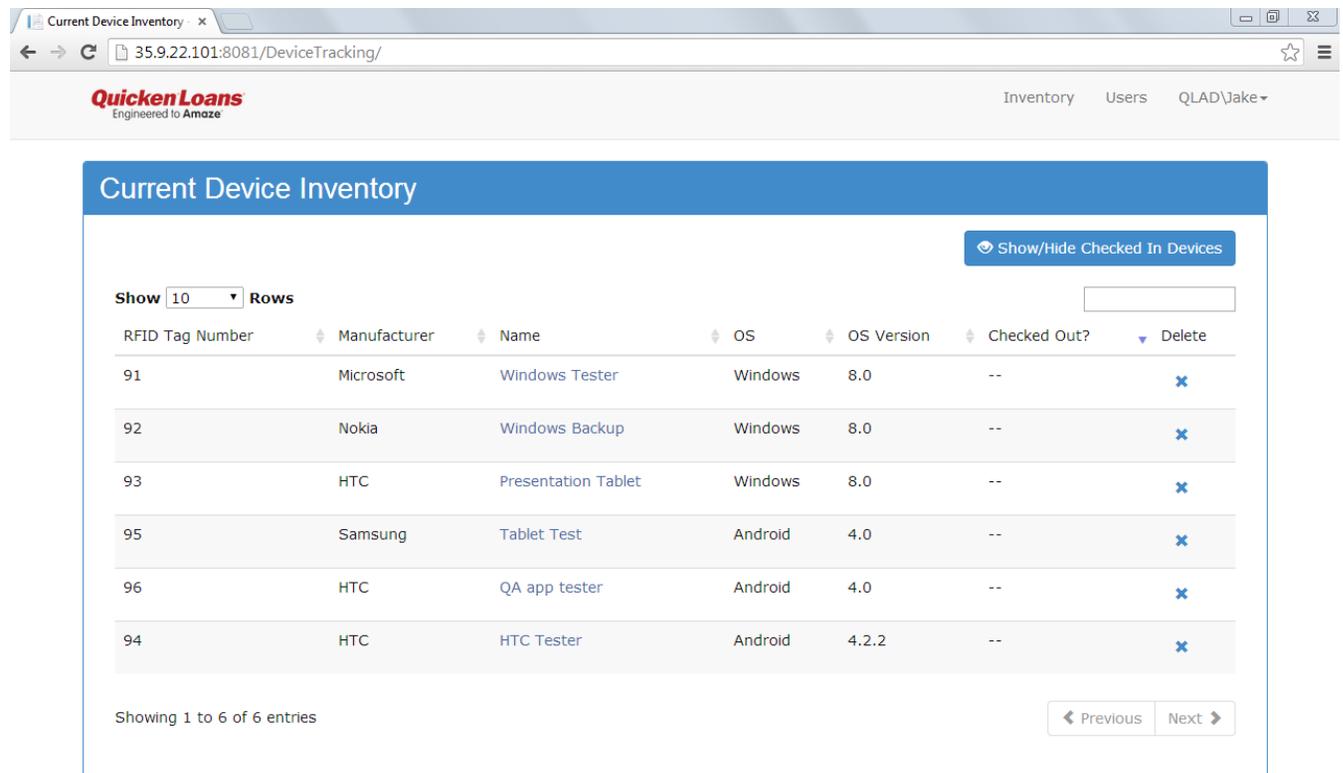


Figure 1: Web Application Device Inventory Page

As shown in figure 1, the web application displays the list of all registered devices with their current status and information. This design will draw on the web services to obtain the data.

User Information

Name	Rasor, Josh	Badge Number	27273
User Id	5	Extension Number	314

Device Checkout History

RFID Tag Number	Device	Check Out Time	Checked In Time
91	Windows Tester	3/25/2014 2:21:53 PM	3/25/2014 2:25:10 PM
91	Windows Tester	3/21/2014 10:26:55 AM	3/21/2014 11:10:30 AM
91	Windows Tester	3/14/2014 12:00:00 AM	3/18/2014 12:00:00 AM
92	Windows Backup	3/25/2014 2:25:10 PM	3/25/2014 2:27:42 PM
92	Windows Backup	3/21/2014 10:22:55 AM	3/21/2014 10:26:55 AM
93	Presentation Tablet	3/25/2014 2:25:10 PM	3/25/2014 2:27:42 PM
93	Presentation Tablet	3/21/2014 10:22:55 AM	3/21/2014 10:26:55 AM

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.

Device Information

Device Name	Presentation Tablet	RFID Tag Number	93
Checked Out?	Checked In	Type	HTC Windows 8.0

Device History

Full Name	Username	Check Out Time	Check In Time
Rasor, Josh	QLAD\Josh	3/25/2014 2:25:10 PM	3/25/2014 2:27:42 PM
Patel, Keyur	QLAD\KJP	3/24/2014 1:38:30 PM	3/25/2014 2:20:00 PM
Patel, Keyur	QLAD\KJP	3/21/2014 11:22:26 AM	3/21/2014 11:25:17 AM
Patel, Keyur	QLAD\KJP	3/21/2014 11:19:12 AM	3/21/2014 11:19:55 AM
Rasor, Josh	QLAD\Josh	3/21/2014 10:22:55 AM	3/21/2014 10:26:55 AM

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.

RFID READER

The RFID reader must be able to take a snapshot of the mobile device inventory. The reader is equipped with an antenna which will monitor the Mobile Device cabinet. The antenna, shown on the right in figure 4, must be placed such that all devices in the cabinet can be scanned by the reader.



Figure 4: The Speedway Revolution reader with antenna

The RFID scanner can scan multiple tags simultaneously without requiring line-of-sight to each device. Multiple antennae can be connected to the RFID scanner to handle larger cabinets.

SEQUENCING

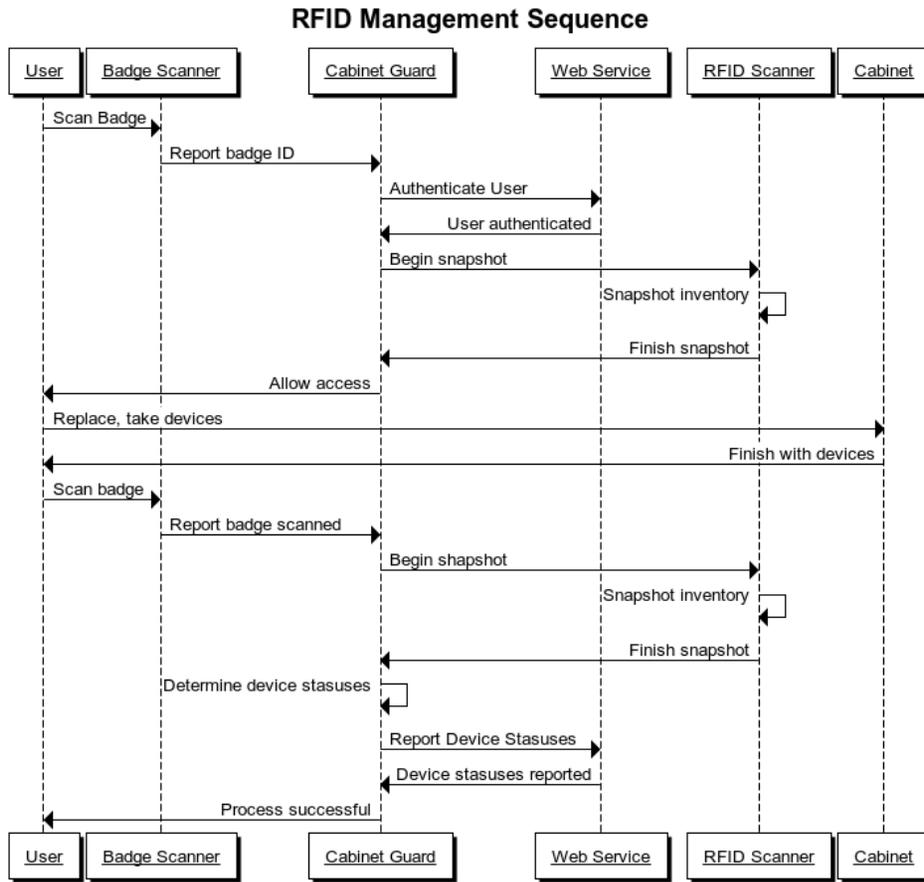


Figure 5: Sequence Diagram for employee use case

As shown in figure 5, the user only interacts with the tracking system by scanning their identification and by receiving notifications. The locking device, RFID scanner, and badge scanner all communicate only to the cabinet guard, which in turn communicates with the web service; the web service is equipped to handle requests from the inventory tracking system, the web application, and possible future presentation-layer developments.

TECHNICAL SPECIFICATIONS

SYSTEM ARCHITECTURE

The Mobile RFID Inventory Tracking System consists of several sub-systems. The system involves a locked cabinet, an RFID reader, a badge scanner, a database, a web service, and a web application. The database, web service, and web application form a three-tier architecture, with business logic isolated in the web services layer. This architecture allows the presentation layer, where the web application resides, to be extended.

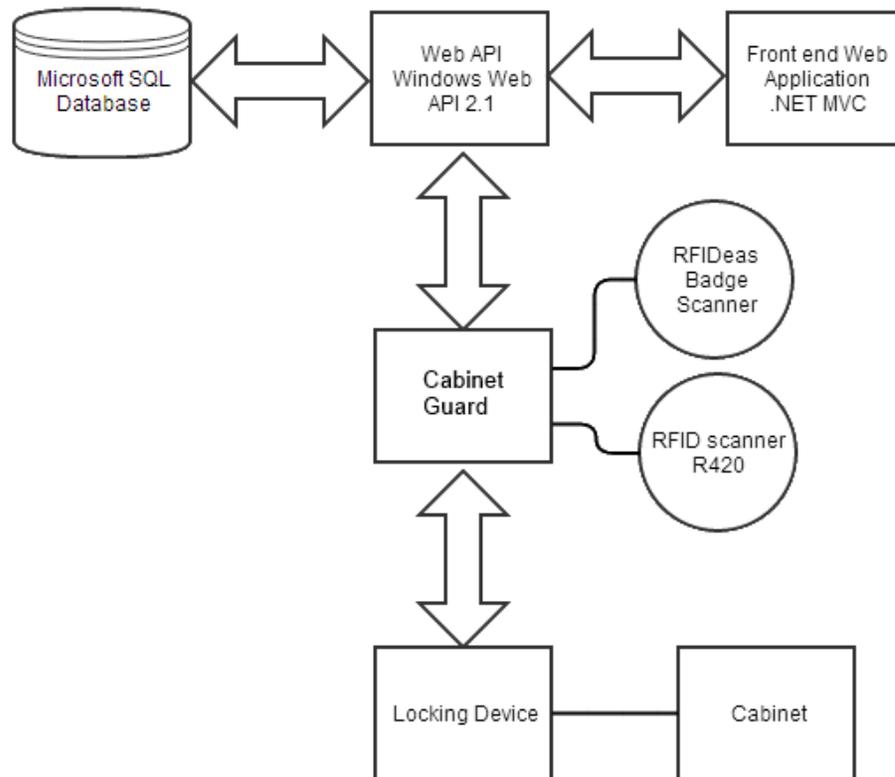


Figure 6: The system architecture of the RFID Tracking system

As shown in figure 6, the three-tier architecture interfaces with the RFID reader and badge scanner at the business logic layer. Separating the web application from the web service allows the web service to be interfaced with possible future applications.

BADGE SCANNER

The badge scanner allows users to interact with the tracking system. The badge scanner is accessible outside of the Mobile Device cabinet, and communicates with both the web service and the locking device. When a badge is scanned, the cabinet unlocks and device inventory changes are attributed to the badge's owner via the web service.

RFID SCANNER

The RFID scanning technology present in this system is the Impinj Speedway Revolution R420 RFID Reader. The reader is equipped with Speedway Connect software. The reader communicates with the Mobile RFID Inventory System via HTTP post with the following variables defined:

BADGE SCANNER

The badge scanning technology present in this system is the RFIDeas RDR-6081AKU NFC Badge Reader. The reader runs without proprietary middleware. The badge scanner requires a channel to communicate with the locking system and the web services. The scanner deposits text to a connected device as a human-input-device. The connected device communicates with the web service to identify users.

CABINET GUARD

The cabinet guard integrates the services of the RFID Scanner, Badge Scanner, Locking Mechanisms, and sends their information to the web service. The cabinet guard is a Windows Forms application, and runs on a device near the cabinet.

WEB APPLICATION

The front end web application uses .NET MVC 4.5. The web application is the top layer of the three-tier architecture. The application is styled using the Twitter bootstrap and follows the Quicken Loans style guidelines for visual consistency. It communicates to the web service via HTTP POST, GET, PUT, and DELETE.

WEB SERVICE

The web services use Web API 2.0. This is the second layer of the three-tier architecture. This layer enforces business logic, and should be accessible for future top-layer applications. This layer provides the interface for the RFID scanner to modify the database based on snapshot readings. The service uses Entity Framework to interface with the database.

DATABASE

The backend database is built using Microsoft SQL 2014. It is the bottom layer of the three-tier architecture.

LOCKING MECHANISM

The locking mechanism is a specialized lock solution for individual cabinets. The Cabinet Guard is designed to integrate with potential locking solutions. Because cabinets may be so specialized with integrated locks, the Mobile RFID Inventory Tracking System does not contain its own lock.

RISKS

BADGE SCANNER

- This system authenticates Quicken Loans employee to access the cabinet of devices and then unlock the cabinet for device retrieval. Not receiving employee badge information in a useable form means that the system cannot properly track devices.
- Mitigation: The badge scanner inputs as a keyboard, and connects via USB; configuration and use is designed for simplicity.

PHYSICAL LOCKING DEVICE

- Locking devices are highly specialized to cabinets; it is not feasible to develop a solution only for particular systems.
- Mitigation: Add abstraction in Cabinet Guard to interface with highly specialized locking solutions.

RFID READER

- Must receive information from RFID Reader, which exists on a network independently of computers running Cabinet Guard
- Mitigation: Speedway Connect software handles the RFID reader's output, allowing the Cabinet Guard to read from a port on the RFID reader.

WEB API 2.1

- Front end for the website should reply on a backend API.
- Mitigation: Implementation of API resembles that of .NET MVC and basics of communication are broken down into simple components. Unified services to handle JSON communication.

TIMETABLE

Week 1 (1/6/14 – 1/10/14)

- Received project description
- First conference call with Quicken Loans Team

Week 2 (1/13/14 – 1/17/14)

- Configured Team Computers
- Installed & Configured Windows 7
- Installed & Configured Visual Studio 12
- Installed & Configured Windows Server 2008 R2
- Installed SQL Server Management Studio 14
- Set-up GIT Repository
- Created a Skelton for Front End
- Email Mockups of Front End

Week 3 (1/20/14 – 1/24/14)

- Status Report Presentations – January 22nd
- Configured SQL Server Management Studio 14
- Activated IIS
- Installed Middleware on RFID Speedway Revolution
- Designed and implemented database schemas
- Implemented a Skeleton for web service
- Work on and complete Project Plan Document & PowerPoint

Week 4 (1/27/14 – 1/31/14)

- Project Plan Presentation – January 27th
- Host prototype website on IIS
- Configure RFID reader to communicate with preliminary Cabinet Guard designs

Week 5 (2/3/14 – 2/7/14)

- Implement web service functionality
- Store and retrieve data from database
- Integrate badge scanner with Cabinet Guard

Week 6 (2/10/14 – 2/14/14)

- Implement simple front end website
- Prepare for alpha presentation, with complete check-in/check-out

Week 7 (2/17/14 – 2/21/14)

- Alpha Presentation – February 17th
- Continued development on front-end and web service functionality

Week 8 (2/24/14 – 2/28/14)

- Usability tweaks to front-end
- Continue front-end and web service functionality

Week 9 (3/10/14 – 3/14/14)

- Configure windows authentication
- Upgrade Cabinet Guard to Windows Forms application

Week 10 (3/17/14 – 3/21/14)

- Finalize usability constraints and definitions
- Implement all notification systems

Week 11 (3/24/14 – 3/28/14)

- Prepare for presentations
- Develop API usability
- Finalize implementation of front-end CRUD operations

Week 12 (3/31/14 – 4/4/14)

- Beta Presentations – March 31st
- Finalization of all features – use-case and administration functional and usable
- Documentation development
- Front-end niceties

Week 13 (4/7/14 – 4/11/14)

- Record audio and record screen for project video
- Finalize source code
- Develop manuals

Week 14 (4/14/14 – 4/18/14)

- Time cushion – plan for this time to be taken by emergencies and unforeseen problems
- Edit project video
- Develop manuals

Week 15 (4/21/14 – 4/25/14)

- Project Videos – April 21st
- Submit All Deliverables – April 23rd
- Design Day Setup – April 24th
- Design Day – April 25th

ADDITIONAL INFORMATION

POINT OF CONTACT

For further information regarding this document and project, please contact **Dr. Wayne Dyksen** at Michigan State University. All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.

ACKNOWLEDGEMENTS

We would like to acknowledge the Quicken Loans team for their contribution and working with us on the Senior Design Program at Michigan State University. Moreover we would specially like to thank Breanna Perrett, Bryan Wisner, Channing Pritchett, Christina Mathes, Sarah Kindinger, Patrick Hartford, Jim Brzuch, Xulu Zhang as well as any other persons we neglected to include.

Additionally, we would like to thank Michigan State University Department of Computer Science, Malcom Doering, Dr. Wayne Dyksen in preparing us for the real world.