MICHIGAN STATE
UNIVERSITY

Project Plan
RFID Mobile Device Tracking System

The Capstone Experience

Team Quicken Loans

Keyur Patel
Josh Rasor
Jacob Riesser

Department of Computer Science and Engineering
Michigan State University

Spring 2014

From Students…
…to Professionals

Project Overview

- The Mobile Device QA team needs a way to track devices that are checked out
- Automated process using RFID technology
  - Each device will be tagged with a unique RFID tag
  - RFID reader will be responsible for knowing what devices are in a cabinet
- Front end website to track and manage devices
- Back end web service to handle business logic

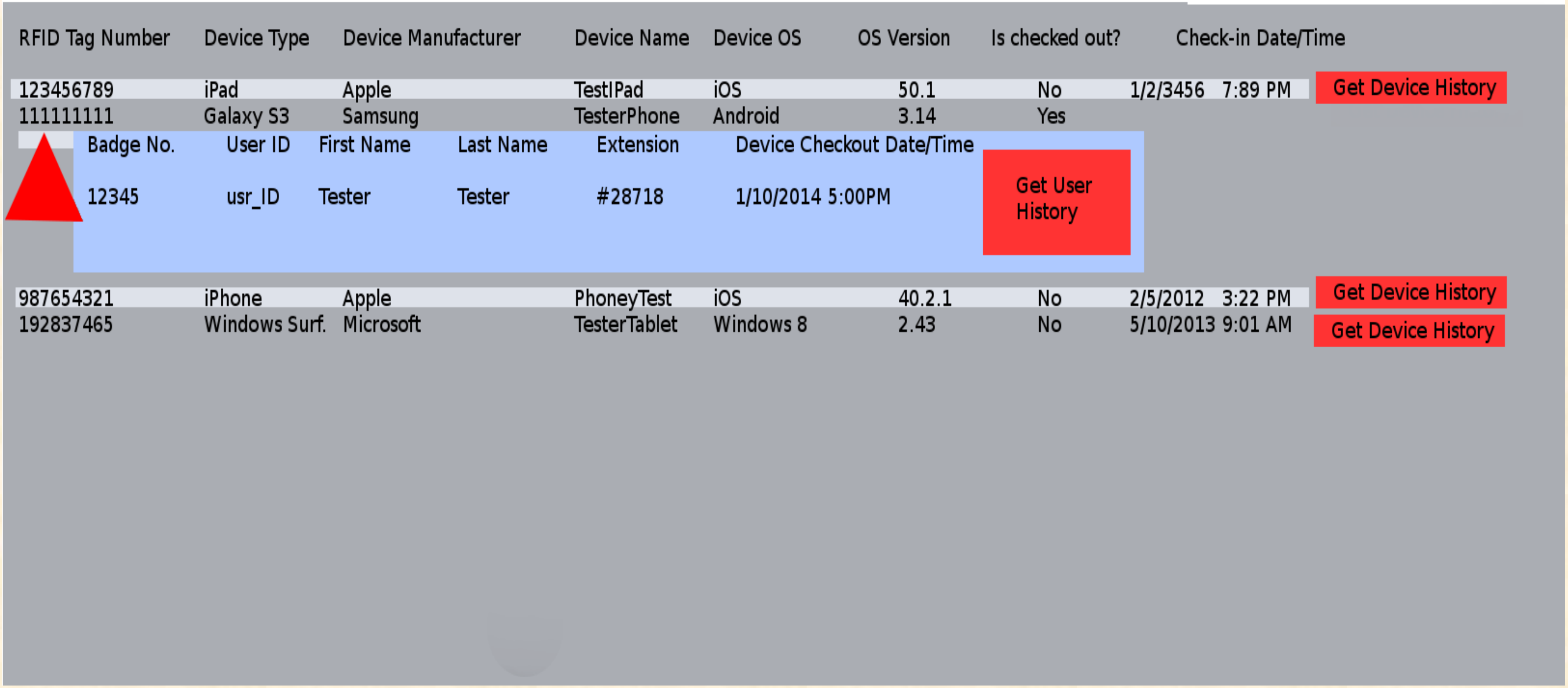
Functional Specifications

- The system automatically checks devices in or out for employees; employees interact with the system only by scanning their identification badge.
- The system authenticates users attempting to access the Mobile Device Cabinet.
- The system can determine the states of all registered devices at any time
- When users check devices in or out, the system will maintain the device check in and out date and time.
- 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.
- A web application is available to review current inventory information, device history, and user history.
- The web application has admin interface to manage inventory information.

Design Specifications

- The front-end web application is built with ASP.NET MVC
  - Styled with Twitter Bootstrap, modified to be consistent with Quicken Loans' visual standards
- The back-end web service is Microsoft Web API 2.0
- RFID Scanner will be able to read (via antennae) devices in cabinet.
- The badge scanner interacts with the locking mechanism and is affixed to the cabinet

Screen Mockup: Inventory Page

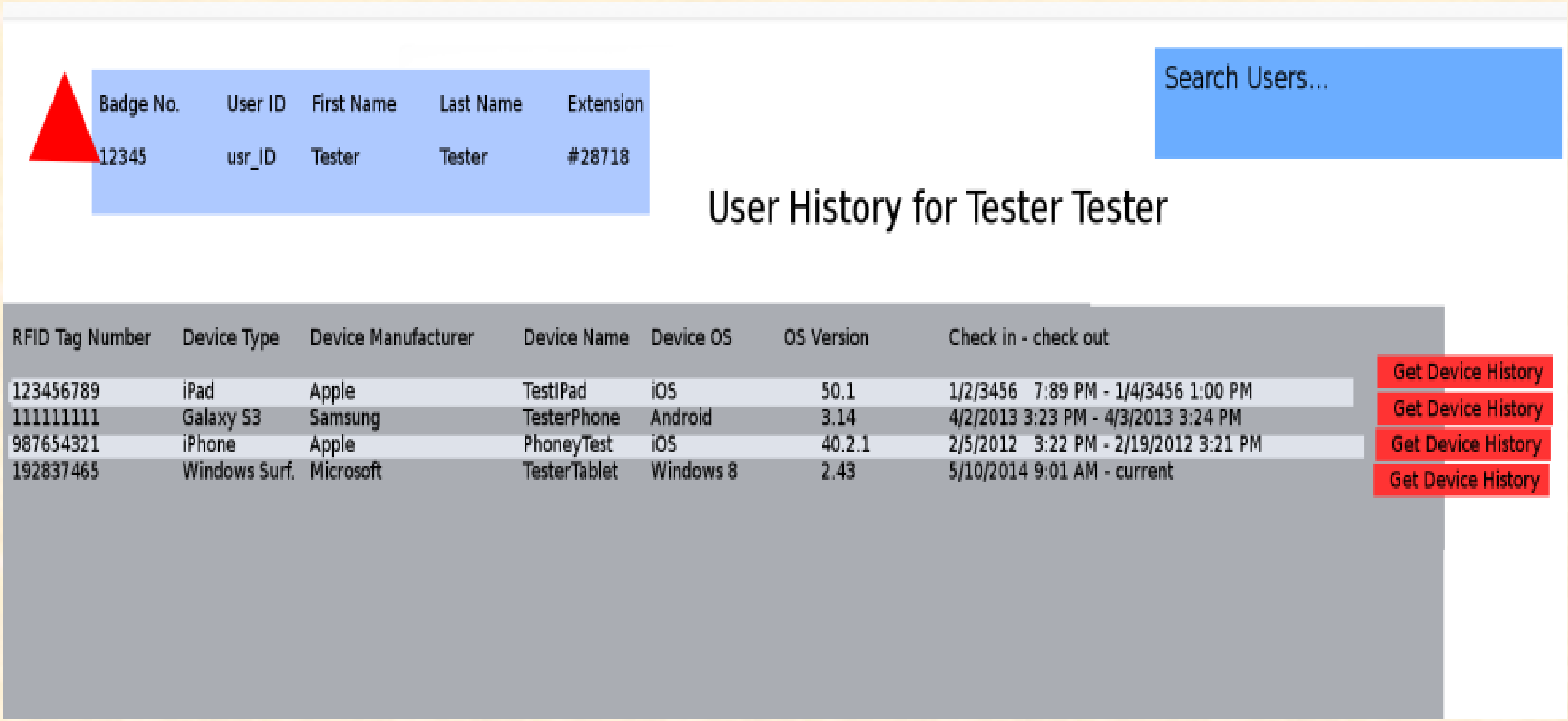


RFID Tag Number	Device Type	Device Manufacturer	Device Name	Device OS	OS Version	Is checked out?	Check-in Date/Time	
123456789	iPad	Apple	TestIPad	iOS	50.1	No	1/2/3456 7:89 PM	Get Device History
111111111	Galaxy S3	Samsung	TesterPhone	Android	3.14	Yes		

Badge No.	User ID	First Name	Last Name	Extension	Device Checkout Date/Time	
12345	usr_ID	Tester	Tester	#28718	1/10/2014 5:00PM	Get User History

987654321	iPhone	Apple	PhoneyTest	iOS	40.2.1	No	2/5/2012 3:22 PM	Get Device History
192837465	Windows Surf.	Microsoft	TesterTablet	Windows 8	2.43	No	5/10/2013 9:01 AM	Get Device History

Screen Mockup: User Information



Badge No.	User ID	First Name	Last Name	Extension
12345	usr_ID	Tester	Tester	#28718

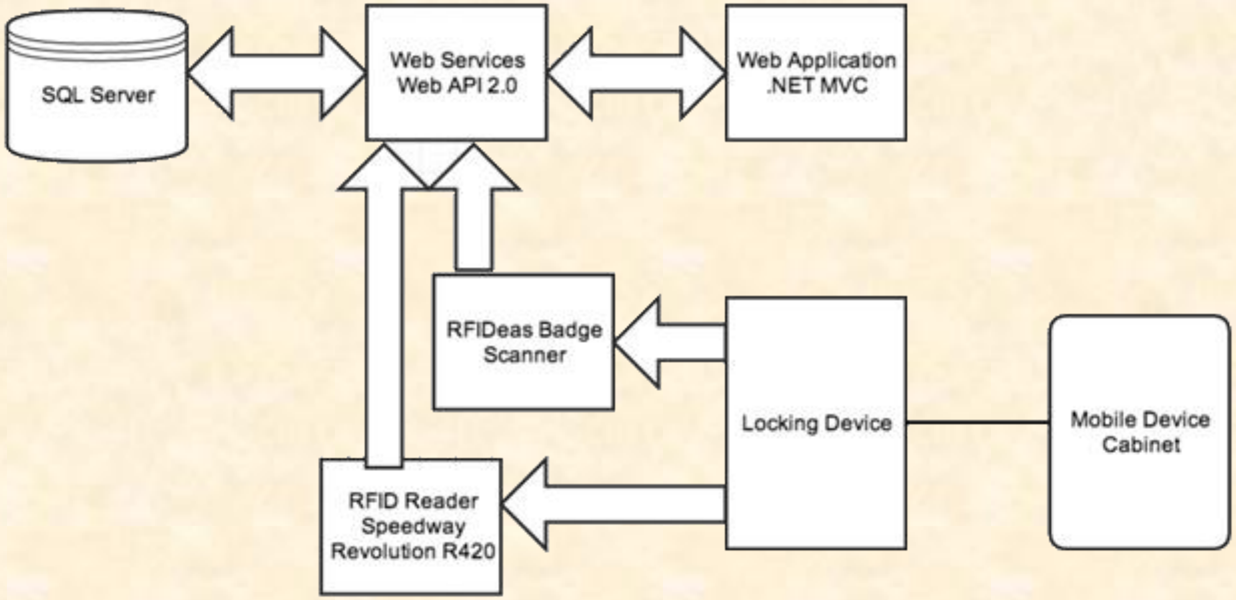
User History for Tester Tester

RFID Tag Number	Device Type	Device Manufacturer	Device Name	Device OS	OS Version	Check in - check out
123456789	iPad	Apple	TestIPad	iOS	50.1	1/2/3456 7:89 PM - 1/4/3456 1:00 PM
111111111	Galaxy S3	Samsung	TesterPhone	Android	3.14	4/2/2013 3:23 PM - 4/3/2013 3:24 PM
987654321	iPhone	Apple	PhoneyTest	iOS	40.2.1	2/5/2012 3:22 PM - 2/19/2012 3:21 PM
192837465	Windows Surf.	Microsoft	TesterTablet	Windows 8	2.43	5/10/2014 9:01 AM - current

Technical Specifications

- RFID Reader is the Speedway Revolution R420 equipped with Speedway Connect middleware
- RFIDeas RDR-6081AKU NFC Badge Reader; no equipped middleware
- Web Application is ASP.NET MVC 4.5
- Microsoft SQL Database in back-end.

System Architecture



System Components

- Hardware Platforms
  - Windows Server 2008 equipped with Microsoft SQL database 2014
  - Speedway Revolution R420 RFID reader
- Software Platforms / Technologies
  - Speedway Connect middleware
  - ASP.NET MVC, Web API 2.0
  - Microsoft SQL database

Testing

- Isolate RFID Reader's behavior with dummy POST method in web service
- Web Application is easily testable; mock fake data in MVC controller
- Web service can be tested in isolation, inserting and pulling dummy values

Risks

- RFID Reader – what is the post format?
  - Mitigate by isolating in single web API method
- Badge Scanner – need to extract output
  - Badge scanner configured to be simple
- Web API 2.0 – unfamiliar technology
  - Well-documented, popular, client resources
- Locking mechanism – need physical results from system
  - Lock's total role limited to very simple input/output