

May15-05 Final Report

A.R.T.I.M.U.S

Augmented Reality Tractor Information Management and Utility System

For John Deere Intelligent Solutions Group

Jesse Walther, Team Lead

Han Sang Youn, Testing Lead

Haoyu Liu, Webmaster

Brian Moran, Key Concept Holder

Tanner Hildebrand, Communications Lead

Manimaran Govindarasu, Advisor

Contents

- Figures and Tables.....iii
 - Figuresiii
 - Tablesii
- Project Introduction..... 1
 - Project Statement 1
 - Final Product 1
 - Specifications 2
 - Key Functional Requirements 2
 - Non-functional Requirements 5
- System Design 6
 - System Requirements 6
 - System Block Diagram 7
 - Application Block Diagram 8
 - Functional Decomposition 9
 - Tractor Vision 9
 - Vehicles 9
 - Settings..... 9
- Detail Description 10
 - Communication 10
 - Interface Specifications 11
 - Primary Application Modules..... 11
 - Vuforia API..... 11
 - MTG API 11
 - Hardware Specifications..... 11
 - Mobile Device Specifications 11
 - MGT Specifications 11
 - Software Specifications 12
- Issues and Challenges 12

Image Recognition	12
Learning Curve	12
WiFi Restrictions	12
Testing Procedures and Specifications	13
Testing Requirements	13
Summary and Conclusion	14
Future of the Application.....	14
Appendix 1 Operation Manual.....	1-1
Sign Up and Log in.....	1-1
Add Vehicle	1-1
Connect to Vehicle.....	1-1
View Information.....	1-3
View Alerts	1-3
Use TractorVision	1-4
Log Out.....	1-4
Appendix 2 Alternative Versions.....	2-1
Original intent for WiFi	2-1
MyJohnDeere Connectivity	2-1
Appendix 3 Other Considerations	3-1
Assumptions	3-1
WiFi.....	3-1
Augmented Reality.....	3-1
Appendix 4 Code	4-1

Figures and Tables

Figures

Figure 1: Tractor Connection Problem.....	1
Figure 2: ARTIMUS Screenshots.....	2
Figure 3: System Block Diagram	7
Figure 4: Application Screen Flow	8
Figure 5: Functional Decomposition	9
Figure 6: Basic Communication.....	10
Figure 7: Adding a Vehicle.....	1-1
Figure 8: Choosing Vehicle for Connection	1-2
Figure 9: After a successful connection the screen will appear as normal.....	1-2
Figure 10: View Tractor Info	1-3
Figure 11: View History and Alerts.....	1-3
Figure 12: Displaying Alerts	1-4
Figure 13: Using TractorVision.....	1-4

Tables

Table 1: Combine Model S690 Farmer Parameters.....	3
Table 2: Combine Model S690 Dealer Parameters	3
Table 3: Sprayer Model R4038 Farmer Parameters.....	4
Table 4: Sprayer Model R4038 Dealer Parameters	4
Table 5: Tractor Model 8R Farmer Parameters	5
Table 6: Tractor Model 8R Dealer Parameters.....	5
Table 7: Communication Protocol.....	10

Project Introduction

Project Statement

John Deere tractors have, onboard, a device called the Modular Telematics Gateway, or MTG, which acts as a server to collect data from the various instruments, gauges, and equipment throughout the vehicle. The MTG has a cellular card which allows it to forward this data to John Deere servers which can be accessed through the John Deere API.

However these tractors are often in the field and may not have access to cellular connection to download this information directly. Our application for iOS devices bridges this gap by connecting to the MTG which, when connected to a TPLink device, can act as an Access Point. By establishing a TCP connection to a server application running on the MTG we are able to download the maintenance data directly from the tractor and display it.

Our application makes use of augmented reality software to identify tractors using the iPhone camera to see the tractor and display pertinent alerts real time.

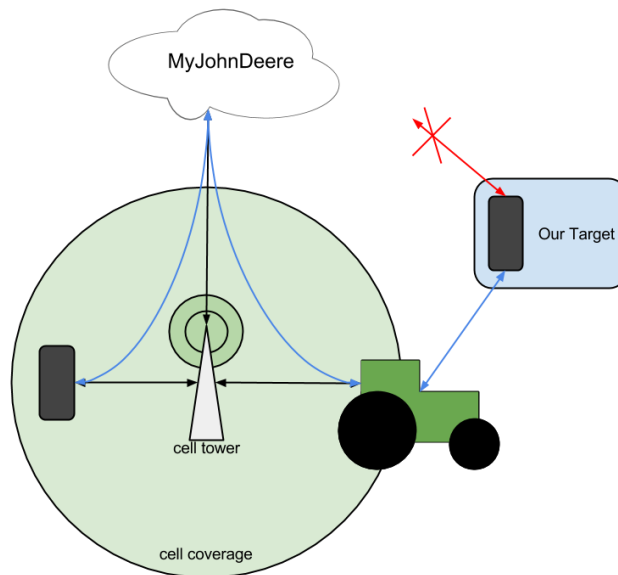


Figure 1: Tractor Connection Problem

Final Product

Our team has created an iOS application running on iOS 7 and up which solves the problem statement listed above by communicating with the device on board John Deere tractors and displaying information on an Augmented Reality display.

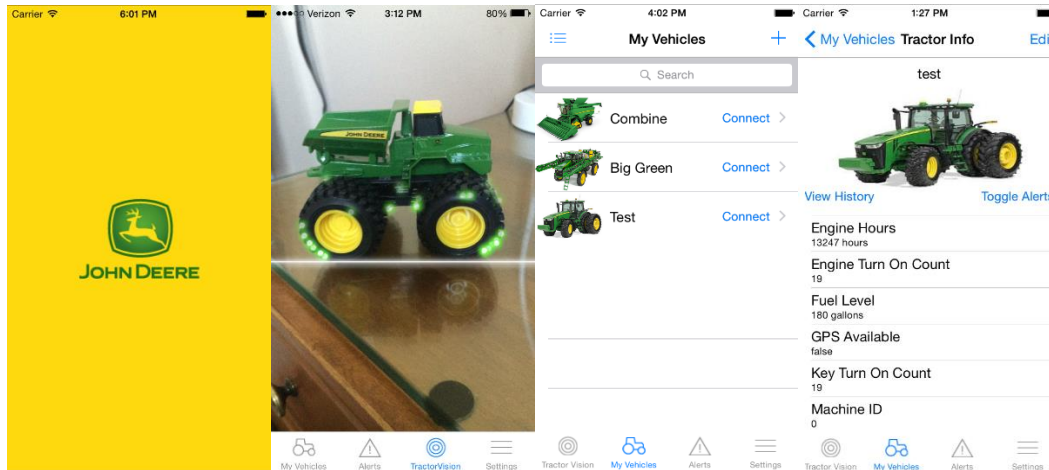


Figure 2: ARTIMUS Screenshots

Specifications

This application provides the following

Key Functional Requirements

- Connects directly to MTG device on board the tractor via wireless and downloads maintenance data securely to the mobile device.
- Augmented Reality View—provides a point and shoot method to identify a tractor and display maintenance information on this tractor.
- Maintenance Information and Alerts—provides information on the maintenance state of the tractor and will display alerts when maintenance parameters pass pre-defined thresholds.

The application works in 3 modes: farmer, dealer and developer. The distinction between modes helps to separate what information is important to each user type and only displays alerts as necessitated by the user type. For our proof-of-concept application we worked with only 3 tractor types and this data set:

- The application will work in the following 3 modes:
 - Farmer Mode—provides all the features listed above.
 - Dealer Mode—provides all the features of the farmer mode and in addition this mode will allow a dealer to quickly sort, classify, tag, and search for vehicles he has registered on his app.
 - Developer Mode—provides all the features of the dealer mode and in addition the app will be able to download logs from the MTG and may provide some information about these logs.

The application will track the following maintenance information and provide alerts as needed for (continued on next page):

Combine Model S690

Parameter to be displayed if user is a Farmer

Parameter	Create an Alert if value is above (or below) this value
Fuel Level [Capacity]	Less than 65 gallon [Capacity is 330 gal]
Machine Hours	Above 50 hrs. Also display total machine hours each time.
Coolant Temperature	Above 85C [if above 95C, engine stop code is generated]
Hydraulic Oil Temperature	55C
RDA License Available	NA
WDT License Available	NA
Machine Sync Available	NA
GPS Working	No GPS info for more than 5 min.
Connected Devices	NA
Tire Pressure	If less than 160kPa [recommended value is 193kPa, 28psi]

Table 1: Combine Model S690 Farmer Parameters

Parameter to be displayed if user is a Dealer

Everything that is displayed to a Farmer

Parameter	Create an Alert if value is above this value
MTG Serial Number	NA
Last Cell Call In time	More than 5 days.
Last reprogramming status	Alert if last reprogramming attempt failed.
MTG Firmware Version Number	NA

Table 2: Combine Model S690 Dealer Parameters

Parameter to be displayed if user is a Developer

Everything that is displayed to a Dealer and MTG Debug Logs

Sprayer Model R4038

Parameter to be displayed if user is a Farmer

Parameter	Create an Alert if value is above (or below) this value
Fuel Level [Capacity]	Less than 40 gallon [Capacity is 155 gal]
Machine Hours	Above 50 hrs. Also display total machine hours.
Coolant Temperature	Above 85C [if above 95C, engine stop code is generated]
Hydraulic Oil Temperature	Above 88C
RDA License Available	NA
WDT License Available	NA
Machine Sync Available	NA
GPS Working	No GPS info for more than 5 min.
Connected Devices	NA
Tire Pressure	If less than 420kPa [recommended value is 441kPa, 64psi]

Table 3: Sprayer Model R4038 Farmer Parameters

Parameter to be displayed if user is a Dealer

Everything that is displayed to a Farmer

Parameter	Create an Alert if value is above this value
MTG Serial Number	NA
Last Cell Call In time	More than 5 days.
Last reprogramming status	Alert if last reprogramming attempt failed.
MTG Firmware Version Number	NA

Table 4: Sprayer Model R4038 Dealer Parameters

Parameter to be displayed if user is a Developer

Everything that is displayed to a Dealer and MTG Debug Logs

Tractor Model 8R- with Group 47/48/49 Tires

Parameter to be displayed if user is a Farmer

Parameter	Create an Alert if value is above (or below) this value
Fuel Level [Capacity]	Less than 50 gallon [Capacity is 190 gal]
Machine Hours	Above 50 hrs. Also display total machine hours.
Coolant Temperature	Above 85C [if above 95C, engine stop code is generated]
Hydraulic Oil Temperature	55C
RDA License Available	NA
WDT License Available	NA
Machine Sync Available	NA
GPS Working	No GPS info for more than 5 min.
Connected Devices	NA
Tire Pressure	If less than 70kPa [recommended value is 80kPa, 12psi]

Table 5: Tractor Model 8R Farmer Parameters

Parameter to be displayed if user is a Dealer

Everything that is displayed to a Farmer

Parameter	Create an Alert if value is above this value
MTG Serial Number	NA
Last Cell Call In time	More than 5 days.
Last reprogramming status	Alert if last reprogramming attempt failed.
MTG Firmware Version Number	NA

Table 6: Tractor Model 8R Dealer Parameters

Parameter to be displayed if user is a Developer

Everything that is displayed to a Dealer and MTG Debug Logs

Non-functional Requirements

- Application runs on iOS 7 and up.
- This application makes use of best practices in order to avoid transmitting sensitive data over unsecured networks and take steps to protect any data that does pass over a network through any medium.

System Design

System Requirements

This application runs on Apple's proprietary iOS mobile operating system starting from version 7 (iOS 7) and up (current highest version is iOS 8).

Devices using this application must have a working camera to use any Augmented Reality or image recognition features.

In order for this application to be functional WiFi connectivity must also be available on the device and enabled so that the device may connect to the MTG on board the tractor to communicate.

For TractorVision, the Augmented Reality portion of our application, we rely heavily on the Vuforia SDK. As such there are some limitations imposed by using this framework. With recent releases of Vuforia 64 bit support has been added and there have been few to no restrictions on our platform. However, since Vuforia updates its software independently of Apple there could be some lag in update time between Vuforia's framework and the iOS platform, therefore in the future a close eye should be kept on the development roadmaps for both.

System Block Diagram

Current versions of the MTG do not have a built in WiFi chip but it did have an ethernet port. Using a TPLink device configured to work as an Access Point and a Gateway for the MTG we could establish an individual connection. Future versions of the MTG may have native WiFi capabilities.

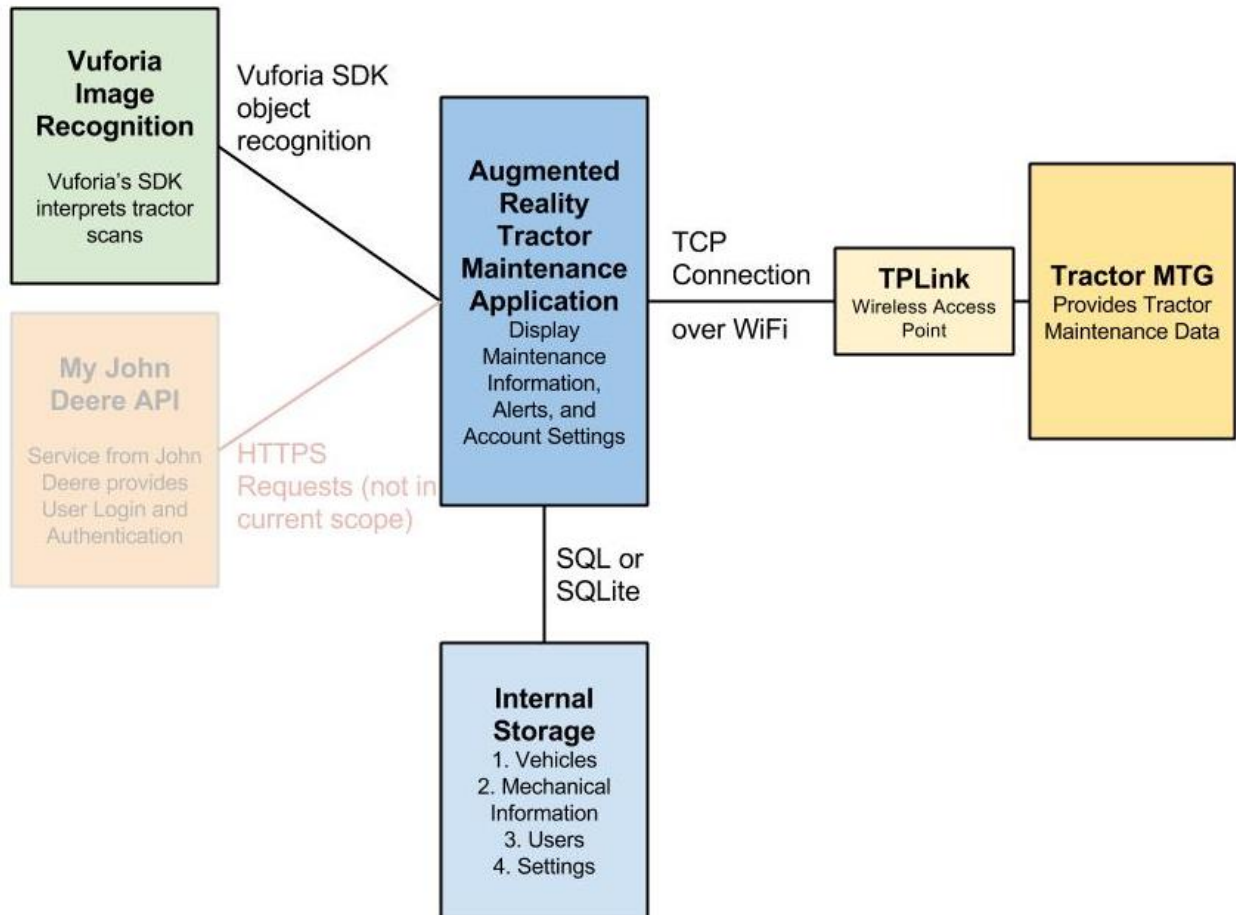


Figure 3: System Block Diagram

Application Block Diagram

Application flow relies on the Apple iOS navigation stack which contributes well to ease of use in our application and a natural, native feel for users. Where ever possible the design adheres to Apple’s iOS design guidelines.

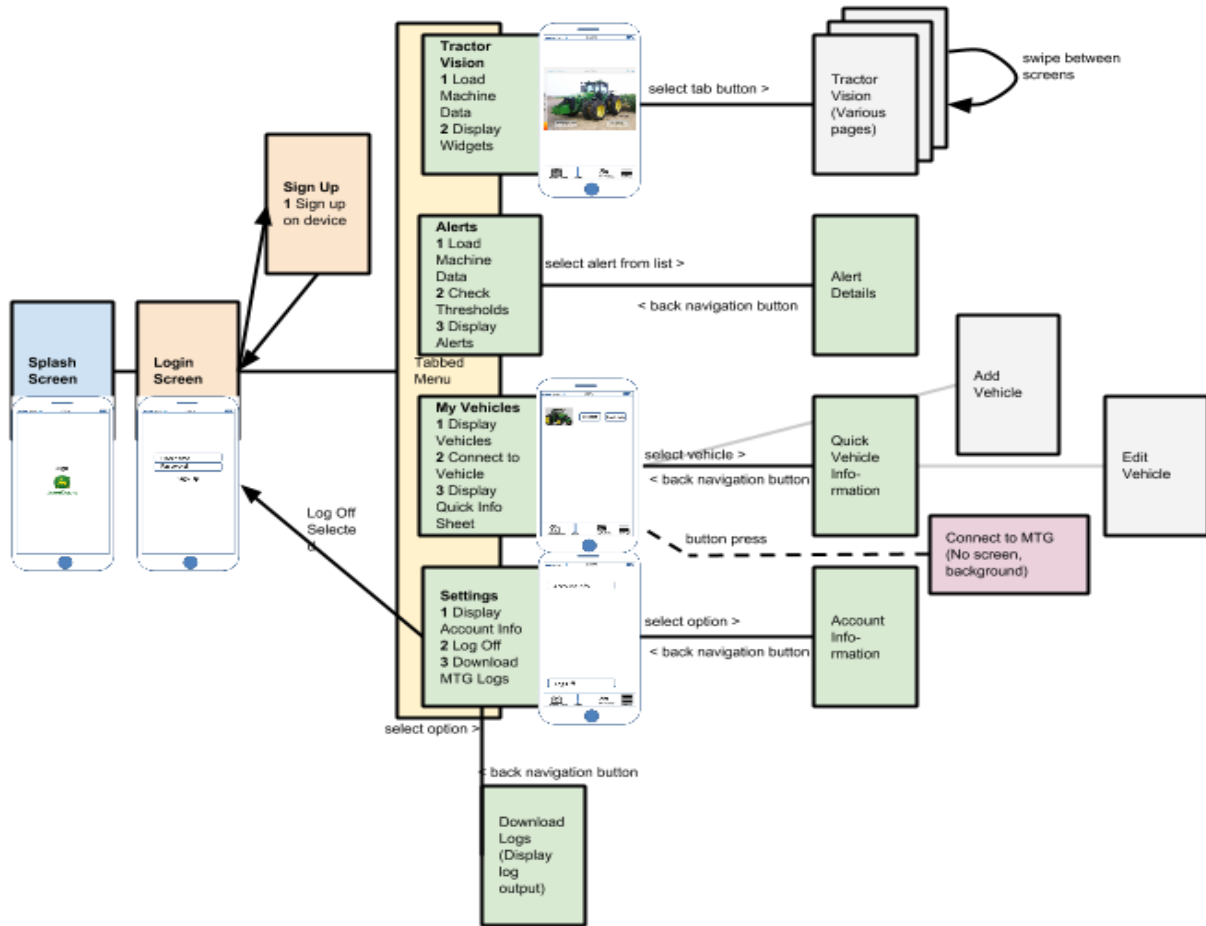


Figure 4: Application Screen Flow

Functional Decomposition

There are 3 main buckets in which the design process falls under in our application: TractorVision, the augmented reality view for displaying alerts; Vehicles, the system to manage tractors and quickly view maintenance information; and Settings, the back end to manage all the details of connecting to a tractor.

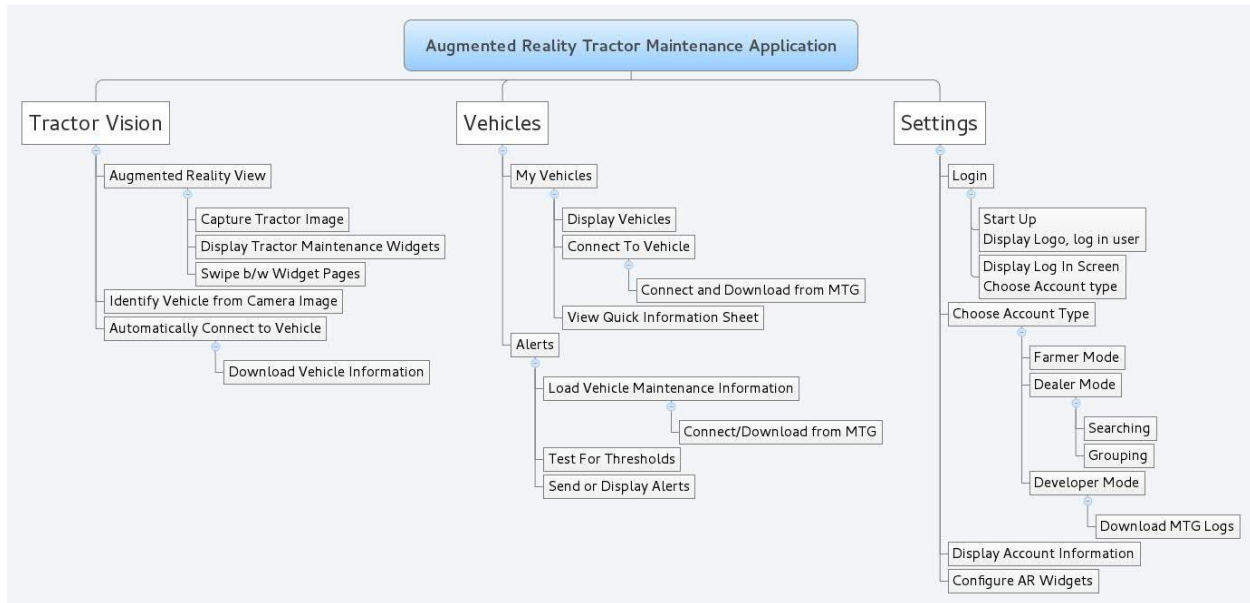


Figure 5: Functional Decomposition

Tractor Vision

TractorVision is an augmented reality view for the application which will recognize tractors and display alerts if any exist.

Vehicles

The user is able to create, edit, and manage the tractors they own which can connect to this application through the My Vehicles page. Also from here the user can transition into a ‘Quick Information’ view which displays all downloaded data pertinent to that user type.

From the Alerts page the user can quickly see if any maintenance information on the application surpasses thresholds defined by John Deere.

Settings

The settings section provides access mainly intended for developers and high level users to manage connection information and stored data.

Detail Description

Communication

This device communicates primarily through wireless with a device on board the Tractor called an MTG.

In order to identify and connect to a machine the user must connect to the TPLink device connected to the MTG as an access point. TCP connection is then established to send requests to the tractor. The protocol for downloading data is simple involving a simple connection and minimal-to-no handshaking. The functions and services on the MTG will be provided by ISG through these TCP requests.

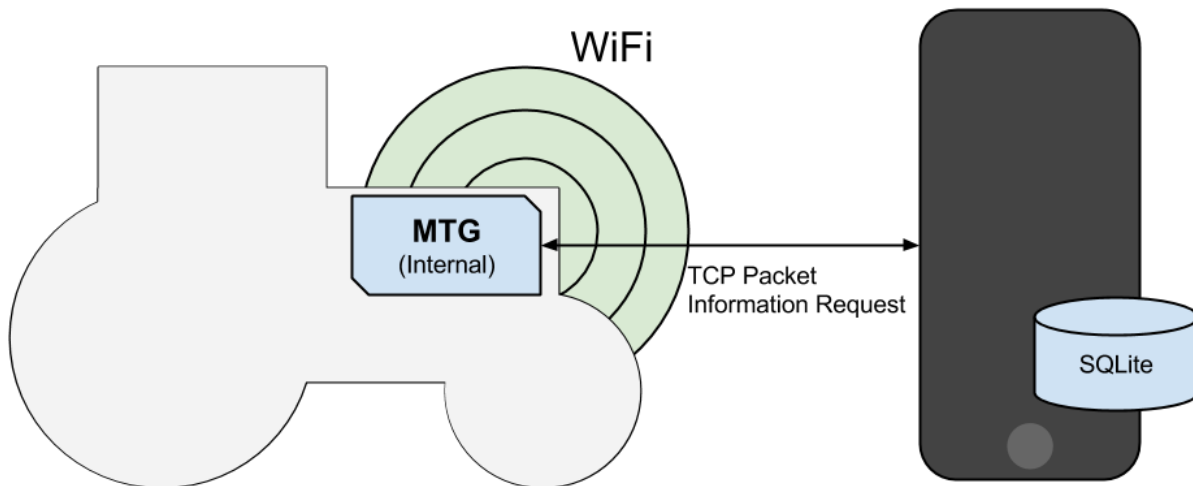


Figure 6: Basic Communication

The connection protocol looks like this:

Application		MTG
Set up:		
Connect to AP for vehicle		
Connect:		
TCP Connection request to 172.16.0.2 port 45544	→	MTG accepts connection
Receive data structure	←	Immediately transmit maintenance data structure
Finish:		
Close Connection	(both sides)	Close Connection
Parse data structure		Await new connections

Table 7: Communication Protocol

Interface Specifications

Primary Application Modules

This application does not provide an API for use outside of the application.

Vuforia API

This application makes use of the Vuforia Augmented Reality software framework/library for iOS. The recently released version (4.0) for 64 bit iOS is used for our system.

<https://developer.vuforia.com/>

MTG API

This application will connect to a tractor's MTG using wireless connection over TCP and make requests to the device in TCP packets. See Communication section for more information on communication protocol.

Hardware Specifications

Mobile Device Specifications

The application will support only the standard Apple iPhone devices running the appropriate iOS version. As such the hardware supported will only be defined to be authentic factory Apple iPhones from iPhone 4 and up.

List of all recent Apple iPhone products (for other specific versions, i.e. 5s, 5c):

<http://www.everymac.com/systems/apple/iphone/index-iphone-specs.html>

MGT Specifications

The current version of the MTG runs Windows CE and does not have a WiFi chip on board however we are able to connect a TPLink device to the MTG which can be configured to work as an access point and a gateway for the MTG. With this we are able to join the TPLink's network and connect to the MTG which has an Ethernet port and a hardcoded IP address.

Future versions of the MTG are likely to have a WiFi card as development continues. Additionally the future versions will run on the Linux operating environment which may serve to solve some of our issues with altering WiFi connections in app. See Issues and Challenges section for more on WiFi and potential solutions.

Software Specifications

The application is only supported for iOS 7 and up (current version is iOS 8). Additional frameworks for development may include Augmented Reality software for iOS and image recognition software.

Vuforia is the primary SDK for AR software for our application considering it is freely licensed to use and widely accepted and documented. See more information on Vuforia Developer Portal here: <https://developer.vuforia.com/>

Issues and Challenges

- Image Recognition on Mobile Devices
- Learning curve for developing with iOS, mobile development, and Augmented Reality
- Security limitations of the iOS system which prevent easy handling of WiFi and other built in libraries provided by the Apple system.

Image Recognition

Augmented Reality is a fairly new but rapidly expanding field. The technology is improving at impressive rates but there are still kinks in the program that are being worked out. Our application is able to run Vuforia well enough, though not without troubles. Current support is much better for Android devices though iOS support is being rolled out with a bit of a delay.

Image recognition is the greatest challenge in using Vuforia. While there is an object scanner available to scan 3-dimensional objects and create a profile for them which the SDK can recognize, objects are currently limited in scale and so only small objects can be scanned—and they can only be scanned using an Android device at this time. This posed only a small challenge to us and was overcome but the size of objects we can recognize remains small.

Learning Curve

The majority of our team does not use Apple devices nor have we ever worked with programming for Apple's platforms so the entire experience was new to us. While stymied by Apple's many restrictions for the most part we were able to work through it.

WiFi Restrictions

Because our application works by switching networks in order to connect to each new Tractor ideally our application would have been able to automatically handle the connection to each tractor. Unfortunately Apple has completely restricted an individual applications ability to do any such thing for security reasons. While we tried many options listed below, in the end we had to settle for manually connecting to each device in turn.

Attempted solutions and current status:

- Private APIs (did not work)
- Use deprecated solution (did not work)
- Transition to wireless page in app (did not work)
- Issue step-by-step instructions on how to connect (current)
- Allow iOS to handle automatically connecting to known WiFi signals (current)

Future versions of the MTG are expected to run on Linux operating systems which will allow us to run MDNS services such as Avahi and Bonjour which will facilitate service detection and hopefully allow the application to avoid this WiFi issue all together. This MTG is currently still under heavy development and is not available for use with our application at this time.

Testing Procedures and Specifications

Testing for application interfacing is manageable through the iOS simulator available with Apple's Xcode software. This software will allow our team to test and debug the fundamental usability of the application but will not be sufficient to fully test the Augmented Reality and connectivity of the application.

XCode provides Unit Testing capabilities as of version 5. However the way which it handles of the unit testing does not facilitate our applications process well. XCode unit testing doesn't allow for set up or tear down which does not play well with our applications database driven systems.

For testing we developed a simple server which could emulate an MTG connection over WiFi and constructed an auditing process which allows us to run batch processes and compare the database output to a defined 'correct' state.

An iPhone 4 has been provided by ISG for testing of the application which will serve to test the interface and hardware capabilities of the application. ISG will also provide a harness with an out of body MTG when testing reaches such a stage which requires testing on the physical hardware.

Testing Requirements

In order for a feature to be accepted to the main application branch it had to show three things:

1. The new feature had to show that it worked as expected under normal usage conditions.
2. Under any usage conditions it must show that it did not cause abnormal behavior in any other currently accepted feature—unless the abnormal behavior revealed a bug in previous features

Often new features would require adjustment to previously established classes and resources making #2 the more difficult task to prove. Using our auditing script we could show that the modifications did not alter the way the data was structured from other processes and updates.

Summary and Conclusion

This application was completed within the two allotted Semesters for this project and with careful selection of 3rd party software and came well within the \$1000 budget.

Future of the Application

There is a good chance that there will be another team next Semester working to develop and improve the application as John Deere completes and improves the next generation of MTG and looks to enhance the services provided by our application.

There are a few features which our team has considered for the future of the application:

- Use of MDNS services (such as Bonjour and Avahi) to locate services on the network (in order to resolve WiFi connection issue)
- Configure TPLink in Ad Hoc network mode to perhaps reduce need to constantly switch networks.
 - This would also alleviate WiFi connection issue but would require a unique IP for each MTG. If each MTG has its own WiFi chip, however, this would be a non-issue.
- Detection of large scale tractors.

Appendix 1 Operation Manual

Follow these instructions on basic use of the application.

Sign Up and Log in

Upon first opening the application you will be presented with a splash screen displaying the John Deere logo for a couple seconds before the application switches to the Log in Screen.

You will have to create an account before you can start using the application. Choose ‘Sign Up’ from the upper left and enter the desired user information. There are 3 levels of user types to choose from. Developer has the most access while farmer has the least. For the most part, all useful features are available at any level.

Once the account is created use the user name and password to log in to the application.

Add Vehicle

Once you are logged in you can add a vehicle to your list of Tractors. From any screen click on the ‘My Vehicles’ tab and then on the ‘+’ button in the upper right corner. Enter all the information requested and select ‘Choose Tractor’ in the bottom right to select the type of tractor to associate with this vehicle. Connection ID is not currently used but is purposed for future development.

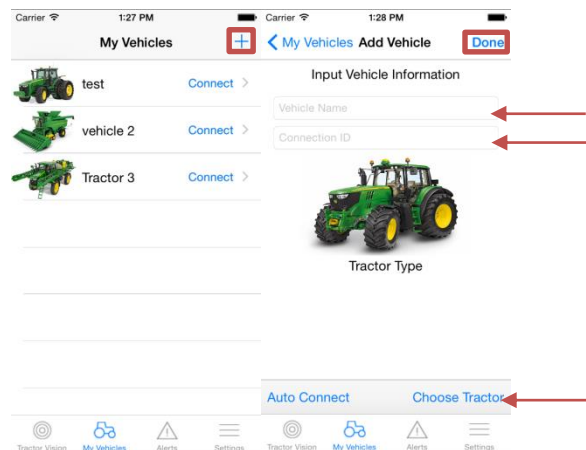


Figure 7: Adding a Vehicle

When completed choose ‘Done’ from the upper right corner to complete the process and save your vehicle.

Connect to Vehicle

Now that you have a vehicle you can start downloading information for this vehicle. Before you can download you’ll need to connect to the appropriate link. If running the simple

server you can connect to the same network as the computer running the application. If you are using the MTG, connect to network provided by the TPLink device.

Once connected you'll need to select the appropriate IP address. Click on the Settings tab and then on the 'Manual Connection' option. In the fields provided enter the appropriate IP addresses. If using the simple server you will enter the IP of the computer running the server and the port will be 4444. If using the MTG the IP will be 172.16.0.2 and the port will be 45544.

Finally you can return to the My Vehicles page and choose 'Connect' for your vehicle. You will be asked if you are connected to the appropriate network, choose the option that indicates you are to continue.

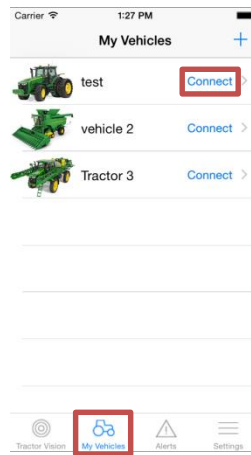


Figure 8: Choosing Vehicle for Connection

The screen will transition to the Tractor Info screen and indicate that it is downloading. You will not be able to use the app while it downloads. When the download is complete your information will display.

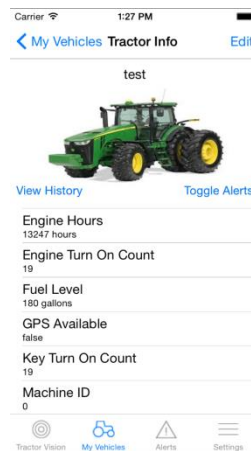


Figure 9: After a successful connection the screen will appear as normal

View Information

At any time from the My Vehicles page you may select the row for any of your tractors to display the information for that vehicle.

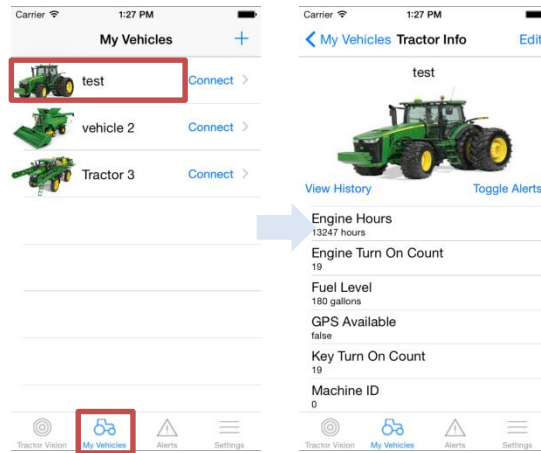


Figure 10: View Tractor Info

From this screen there are two other options: View History and Toggle Alerts.

View History will display up to 300 downloaded values from the past and the Alerts view will display all values which are indicated as alerts by the application.

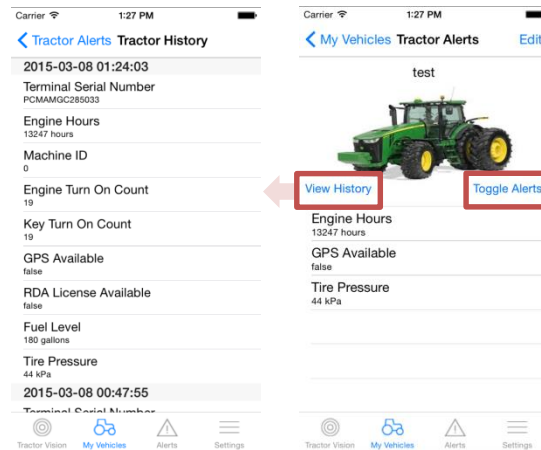


Figure 11: View History and Alerts

View Alerts

The Alerts tab will display alerts for all of your tractors but only if that tractor has an alert and only if the alert has not expired (after 7 days alerts go stale).

To view alerts for a Vehicle click on the row with its name and picture to expand the alerts view. Alerts can be marked as 'checked' by clicking on them and 'unchecked' by clicking

again. This can indicate issues you have solved but have not downloaded the updated data from your machine yet.

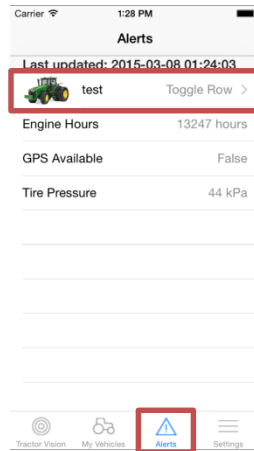


Figure 12: Displaying Alerts

Use TractorVision

To use TractorVision select the TractorVision tab and then click to start the Augmented Reality view. This will start the camera and the Vuforia scanner.

If the camera is pointed at an object which has been loaded into the applications image recognition repository the application will recognize it and display any information or alerts it has in relation to the type of vehicle it recognizes.



Figure 13: Using TractorVision

Log Out

Once done, if you would like to log out of your session you can transition back to the settings screen and choose the Log Out button to return to the log in screen and clear your session.

Appendix 2 Alternative Versions

Our application was developed using an iterative process, as per ISG's request, alongside with ISG's development of the MTG capabilities. As such there was very little versioning but the application would grow with each monthly meeting with ISG to refine our plan of action for the application.

However, there were some initial assumptions that proved to be a challenge and some which were impossible to implement given the platform restrictions:

Original intent for WiFi

Originally we intended for the application to be able to automatically select the appropriate vehicle network and transition to it in-application. This, however, was rendered impossible by Apple as they systematically disabled this feature in iOS over the various versions.

This means that users currently have to select the appropriate network themselves prior to establishing a connection to the tractor. This is not a heavy burden, however, as once the phone has seen the network once it is capable of automatically connecting to a known network (such as one provided by the tractor) and the user will not need to manually connect in this case as long as it is the only network in the area. Since our application specifically is targeting a field environment where there are few signals, this reduces the impact of the issue.

MyJohnDeere Connectivity

John Deere is currently developing a 'cloud' service for the tractors to call in and update their maintenance data directly to John Deere servers. While our application fills the void of being in the field where cell connection is not available, it was always considered a possibility that we could integrate some of the remote cloud services into our application.

While this never became a requirement for us our development process and final product left some availability for it to be added in the future. This service could allow the farmer to automatically synchronize data while not in the immediate vicinity of his equipment and assist developers in debugging between cloud services and tractor services.

Appendix 3 Other Considerations

Our project proved to be a bit rough at times.

Assumptions

Initially we made the rather foolish assumption that we would be developing on an Android platform which would have given us a great deal of flexibility. However, more than 50% of John Deere's user base uses Apple devices and so it was required we work on the iOS platform instead.

This was a bit of a pain. Only one member of our group was a regular user of Apple products so we immediately ran into the problem of not having the appropriate equipment to develop applications for iPhones. The only software for developing for iOS is Xcode which itself only runs on Mac computers. Luckily CSG was able to configure the machines in the TLA to work for us, but posed the problem of limited resources as they filled quickly with other students.

WiFi

As previously mentioned connecting in app to WiFi was a never ending challenge for us. It represented the vast majority of our research time, experimentation time, and frustration. At some points we had 2 to 3 team members working directly to find a way around this problem but eventually we had to admit that the iOS platform would not allow us to do this and we settled for plan B, as described above.

Future versions of the MTG are anticipated to run on Linux OS and will facilitate software which can help solve this issue using MDNS solutions.

Augmented Reality

Augmented Reality was our second biggest challenge as we needed to find a service that was free for us to develop on and also well maintained. We found that in Vuforia and have had a good deal of success with it.

During our development time of about 7 months Vuforia went from version 2.0 to version 4.0 on an accelerated development plan as the market for this software explodes. This was good for us because initially the software did not work well on 64 bit platforms.

The addition of the object scanner for Android allowed us to scan 3 dimensional objects for the application to better recognize tractors.

Appendix 4 Code

To see more on the code and interfaces behind the application visit:

<http://may1505.ece.iastate.edu/docs/html/index.html> for current documented classes.

For a more in depth explanation of the design and implantation of the code see Appendixes A and B of the final Design Document at: <http://may1505.ece.iastate.edu/documents.html>

All code was source controlled using the ECpE git server: <http://git.ece.iastate.edu>