## May1609

# A.R.T.I.M.U.S. Phase II

## Augmented Reality Tractor Information Management Utility System

# Contents

# References

## Figures

# Project Statement

Currently John Deere tractors have the capability to send their own vehicle health information to their MyJohnDeere account which can be accessed on the web. This process is achieved largely due to the onboard device on the tractors called the Modular Telematics Gateway, or MTG. The MTG performs as a server that collects data from the tractor such as: hydraulic oil temp, fuel level, machine hours, tire pressure, and others as well. The problem that arises from this implementation is that the MTG can only send its machine data to the MyJohnDeere API if there is a cellular data connection available, this is where our project fills the gap.

Our iPhone application communicates to the MTG via Bluetooth to request all of its data from the tractor. From there our easy to use interface allows the user to easily navigate through all of their tractors and view the alerts for each tractor if they're in the field with no cellular signal. In addition to this our mobile application also offers some new and cutting edge augmented reality software that allows the user to simply point and shoot a picture of their tractor, which then displays visual alerts on the screen.

## What Value Does Our Project Offer

Our project is designed for the owners of John Deere tractors in order to give them an easy resource to gain information and know the status of their vehicle. In a sense our project is designed to be a one stop shop for the farmer so that they have access to all of their vehicle's health information in one place. Through the use of image recognition and augmented reality software our application provides the user with a quick, interactive way of displaying the vehicle’s information by just having the user simply take a picture of their vehicle. The design of this application will provide developers with information of the tractor through the use of log files. These log files will be logged to the MyJohnDeere API and be available to developers. Vehicle dealers are intended to benefit from having a convenient method to look up, sort, get alerts, and quickly display information about vehicles they have registered

## Is this Project Feasible

Given our resources and schedule this project is definitely feasible. Last year’s project team completed a lot of the major functionality and provided us with a great base. Much of the required data is available through John Deere’s developer API and the constant connection with their tractors. Our team will not have to make any changes in hardware and will be focused mainly on communication, user interface, and a robust augmented reality component. Mobile applications are extremely powerful and common, which means the support for developers is great. More specifically the resources for Apple iOS development are large and will become extremely beneficial.

# Project Requirements

## Functional Requirements

- Cloud Storage
  - Additionally to storing vehicle health information on the mobile phone, we also incorporated a cloud aspect to store and retrieve data through the Firebase Cloud Service
- Bluetooth Communication
  - All communication between the MTG and the mobile app will be done through Bluetooth
  - The MTG is the peripheral and the phone is the central
    - The phone will scan for the MTG and then connect and read its service values for the MTGs GATT (Generic Attribute) profile
- Augmented Reality
  - Augmented Reality and image recognition is incorporated into our application to provide the user with a more visual, easy to use, and interesting experience
  - Wikitude Augmented Reality SDK is used
  - Simple point and shoot for the user to visually see what is wrong with their vehicle

## Non-Functional Requirements

- User Interface
  - The user interface abides to the John Deere style guide that is used for their other mobile apps
- Runs on new versions of iOS
  - Wikitude along with all of the Xcode source code can run on the newest version of iOS
- Security
  - User authentication
  - Bluetooth communication with MTG can only be established if the user is a John Deere customer
  - Cloud reading and writing abilities can only be done through our firebase account

# System Design

## System Requirements and Standards

| Component | Description |
|---|---|
| System 1 | iOS devices with iOS 7+ |
| System 2 | Yukon MTG Low Energy Bluetooth Capability (client provided) |
| Xcode | Version 7+  Apple's iOS development Platform |
| MyJohnDeere | Cloud services provided by client API |
| Wikitude | Augmented Reality and image recognition software (free trial) |
| Vuforia | Augmented Reality and image recognition software (free) |
| Bluetooth Connection | Establish connection securely with Yukon through Bluetooth |
| UI | User interface specific to fit client and user needs |
| AR UI | Augmented reality to provided user with easy point and shoot capability |

Table 1

Our Project will rely heavily on John Deere's MTG or Modular Telematics Gateway, as shown below. This hardware is the core of our project. The device is embedded in the John Deere vehicles that our application connects to. The MTG is equipped with low energy Bluetooth and an on board Wi-Fi chip. In addition this device gathers all of the important vehicle health information that will be sent to the mobile app. To find out more information about the MTG follow this link: https://www.deere.com/common/docs/products/equipment/telematics/brochure/john_deere_mtg.pdf

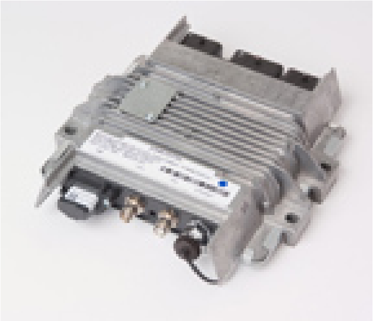


Figure 1

## Functional Decomposition

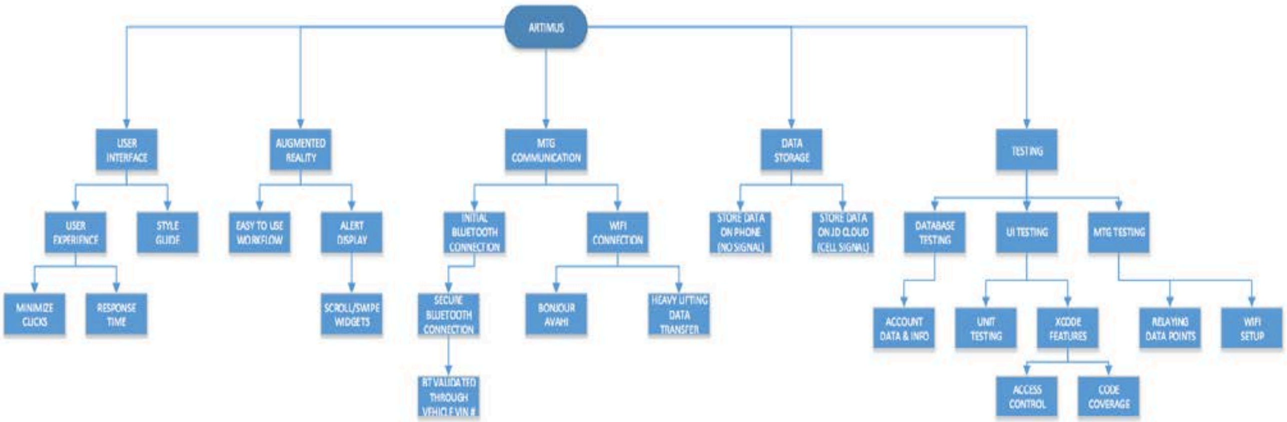


Figure 2

If image is too small go here http://may1609.sd.ece.iastate.edu/uploads/treeDiagram.jpg

## Overall Project Completion Chart

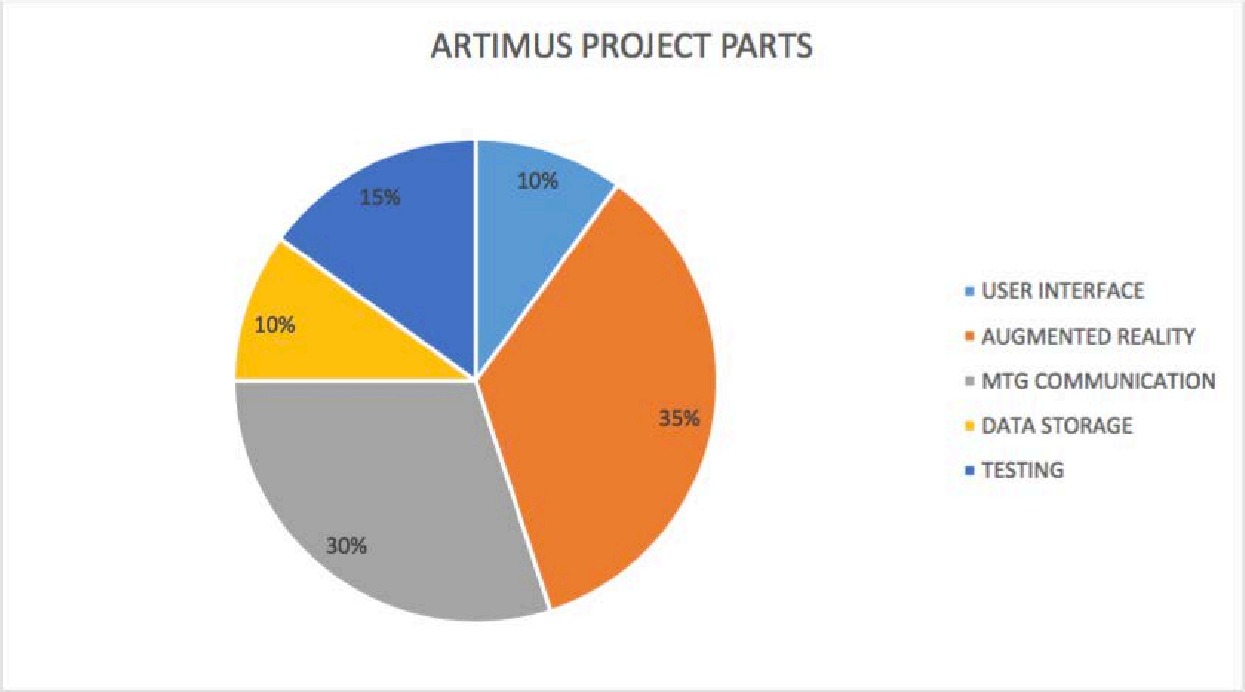


Figure 3

## System Analysis

This project is comprised of the following pieces and resources:

*Wikitude Augmented Reality SDK*
Our team will be using the Wikitude library to handle the image recognition, and augmented reality experience of the application. This will replace the Vuforia SDK that was used in the first iteration of this application. Wikitude provides far more robust image recognition, and is well documented and tested for iOS devices, the target device of our project.

### Xcode iOS Development

The Xcode IDE provides all the tools necessary for developing iOS applications. An especially useful tool for our team will be the Interface Builder, which will be used to design the application's user interface to match John Deere design standards.

### External APIs

Communication with each tractor's onboard MTG will involve the use of John Deere APIs, which are maintained and supported by the Intelligent Solutions Group. There are no anticipated issues with those.

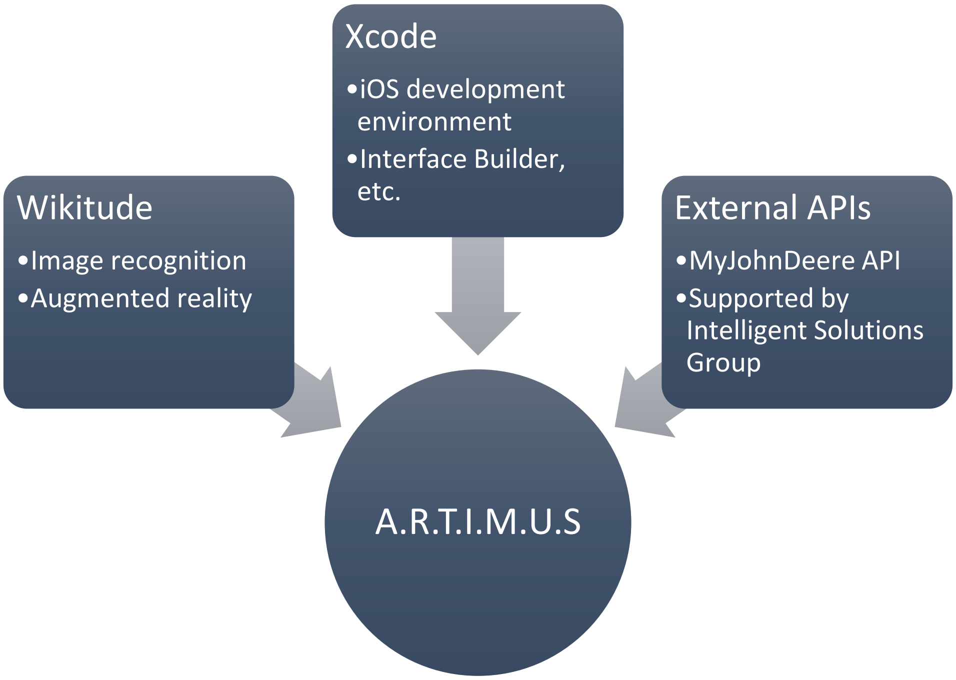


Figure 5

## System Block Diagram

Currently data is being stored on the mobile device itself and we are looking to store all of the data on a Firebase cloud component instead.

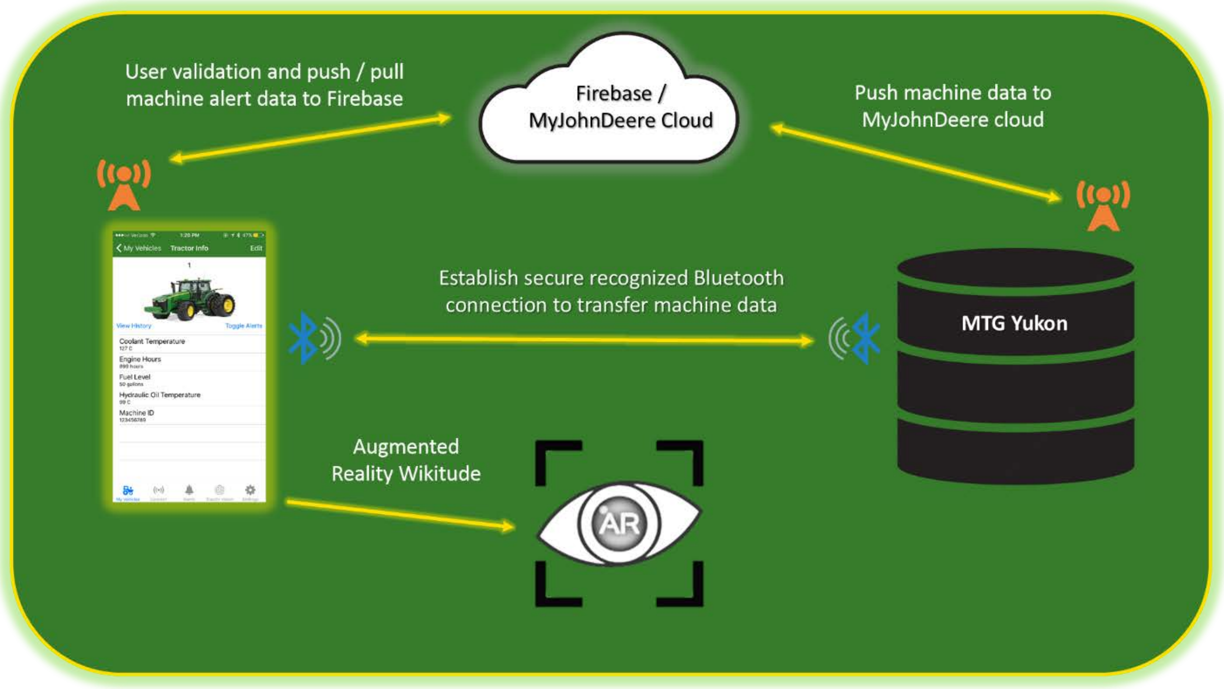

User validation and push / pull machine alert data to Firebase
Firebase / MyJohnDeere Cloud
Push machine data to MyJohnDeere cloud
My Vehicles
Tractor Info
Edit
1
View History
Toggle Alerts
Coolant Temperature
127 C
Engine Hours
899 hours
Fuel Level
50 gallons
Hydraulic Oil Temperature
99 C
Machine ID
123456789
My Vehicles
Establish secure recognized Bluetooth connection to transfer machine data
MTG Yukon
Augmented Reality Wikitude
AR

Figure 6

## Application Block Diagram

Our application makes use of the Apple iOS navigation stack, which provides the user with a familiar and natural user interface. Our design standards are meant to comply with John Deere and Apple.

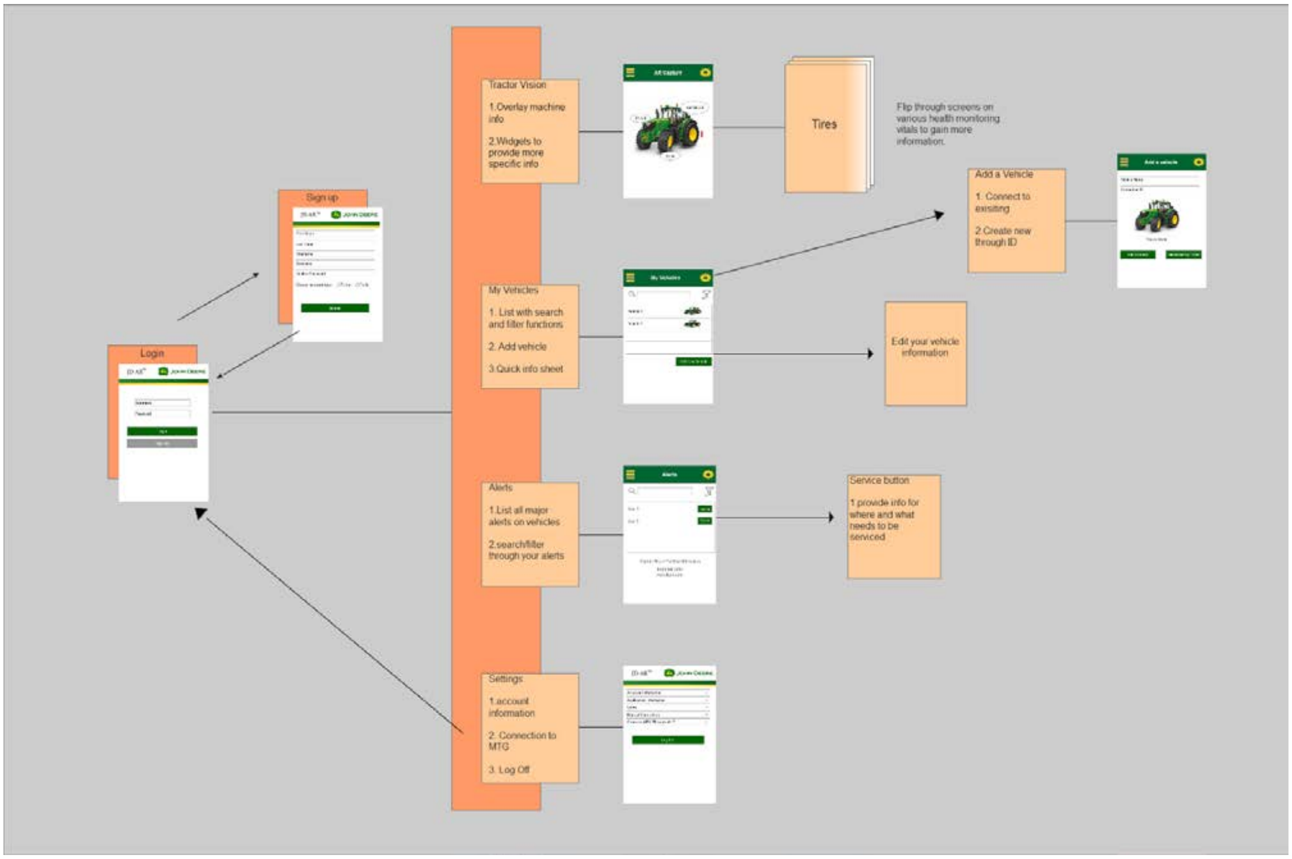

Tractor Vision
1.Overlay machine info
2.Widgets to provide more specific info
Tires
Flip through screens on various health monitoring vitals to gain more information.
Sign up
Add a Vehicle
1. Connect to exisiting
2.Create new through ID
My Vehicles
1. List with search and filter functions
2. Add vehicle
3.Quick info sheet
Edit your vehicle information
Login
Alerts
1.List all major alerts on vehicles
2.search/filter through your alerts
Service button
1 provide info for where and what needs to be serviced
Settings
1.account information
2. Connection to MTG
3. Log Off

Figure 7

# Detail Description

## I/O Specification

A user’s iPhone device communicates with a device on board the tractor called an MTG. The phone will use Bluetooth and wireless to transmit and receive data.

In order to identify and connect to a machine Bluetooth is used to probe for available vehicles to connect to.  Based on the received information, the user can connect to the MTG and continue to receive real time health information from their machine.

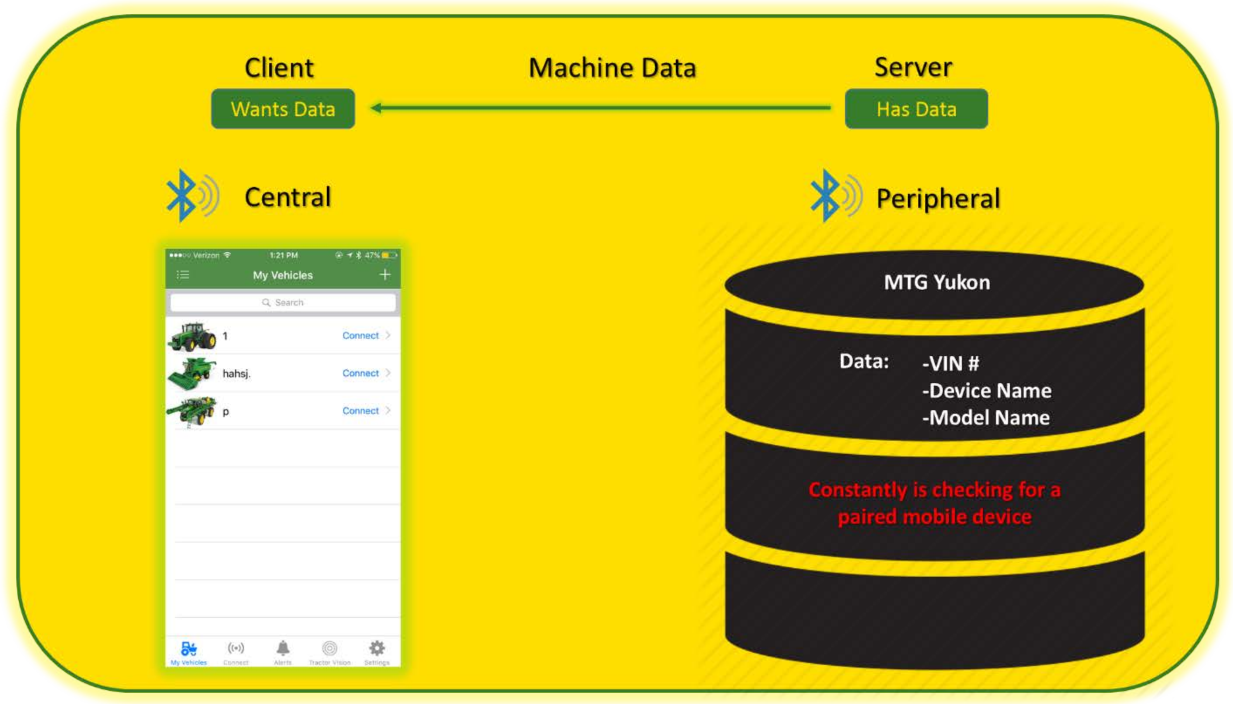


Figure 8

The application will also use the camera on the device to capture the tractor.  The camera features are to be handled through Apple camera framework provided with the iOS SDK.

For the use of some frameworks and libraries, such as the Wikitude Augmented Reality library.

# User Interface Specifications

## Login and Signup

After the application loads the user will be first presented with the login screen. The user will have to create an account before entering the application.

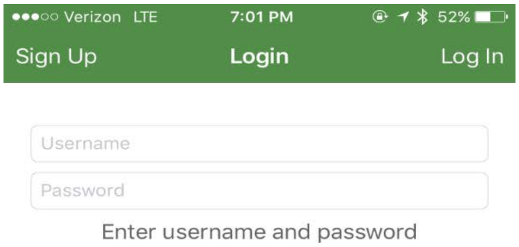


Figure 9

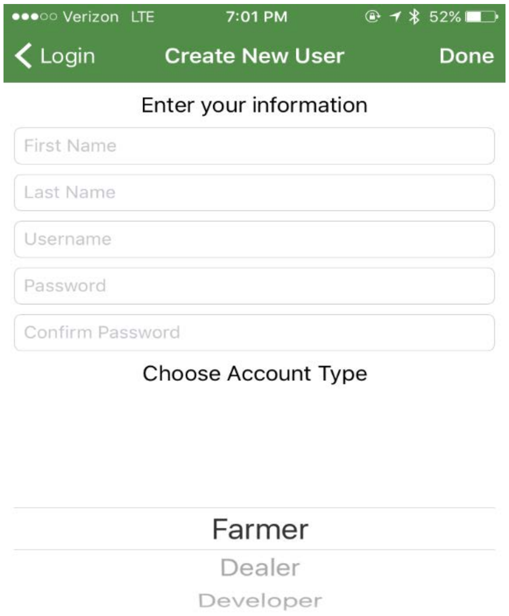


Figure 10

The user will input the information shown above with some password constraints. The password constraints are 6-10 characters with one numerical, and capital letter. Upon creating the account the user will receive a notification showing that their account has been successfully created and then they can login.

## My Vehicles

The user can then navigate to the my vehicles page once they have successfully logged in. This is the default home screen. Here the user can see their vehicles in a list view and search throughout the list based on some standard filters or by name. Those filters will include: date and type. Within the my vehicles page the user can also create a new entry if they choose.

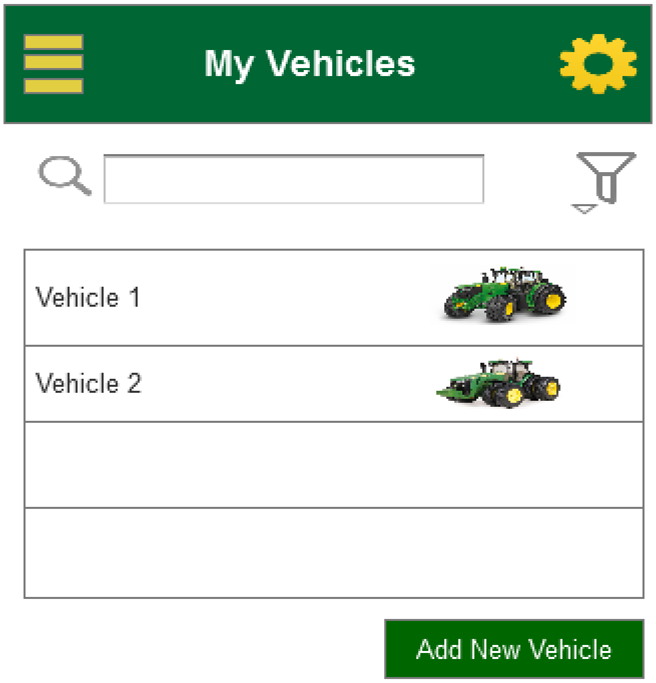


Figure 11

## Add Vehicle

This page provides the user with a variety of three tractor types to choose from, then they can name the tractor and begin connecting to the MTG to receive alerts.

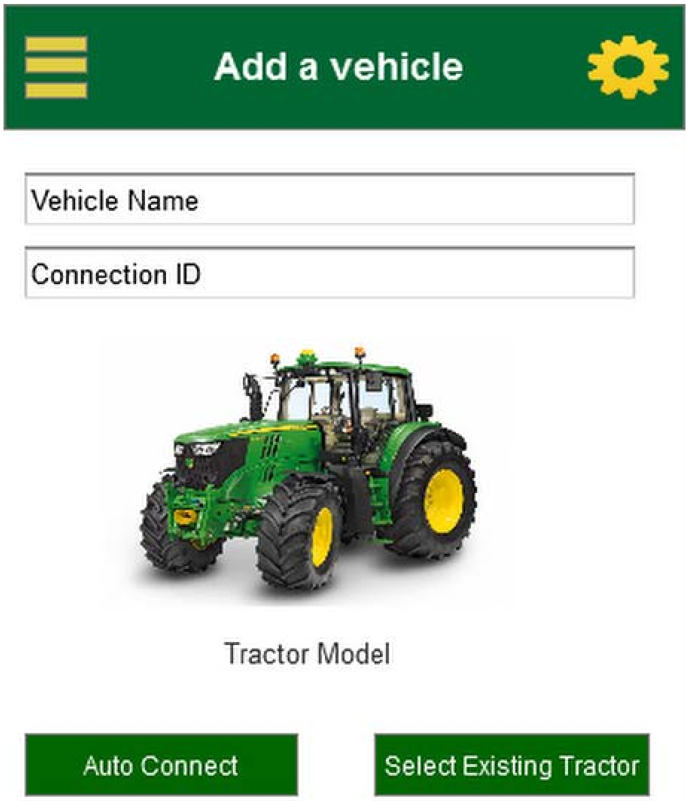


Figure 12

## Settings

The settings page can be accessed by selecting the settings icon in the upper right corner of the screen. From here the user has the option to see account and application information along with connection information.

The connection information is designed to provide the user with easy instruction on how to connect to their vehicle through the MTG.

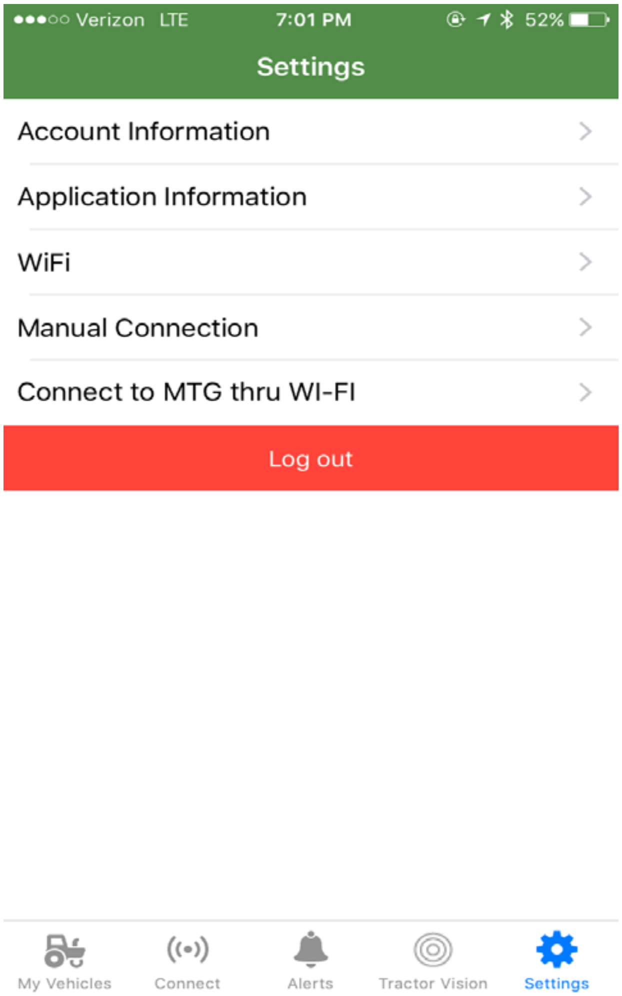
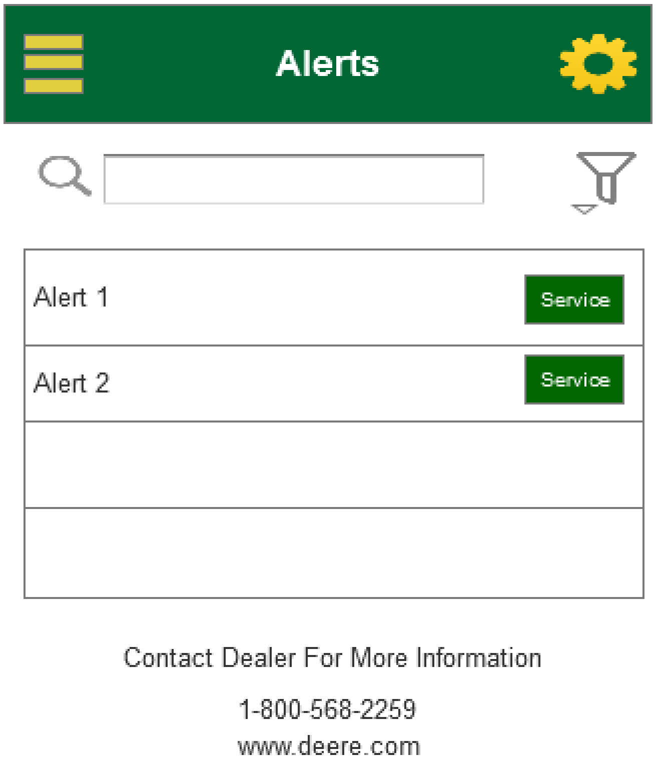


Figure 13

## Alerts

The alerts page provides the user with a list view of all major machine health alerts of all their vehicles. The user can search through these alerts by filtering them based on category, importance, and date. From there the user can also click on the service button to bring to John Deere's website for maintenance scheduling.



Contact Dealer For More Information

1-800-568-2259
www.deere.com

Figure 14

## Augmented Reality

The augmented reality capture screen allows the user to take a picture of their vehicle and in turn see machine information being overlaid on top of their vehicle. From there the augmentation will allow the user to navigate through widgets for more information about the status of their vehicle.

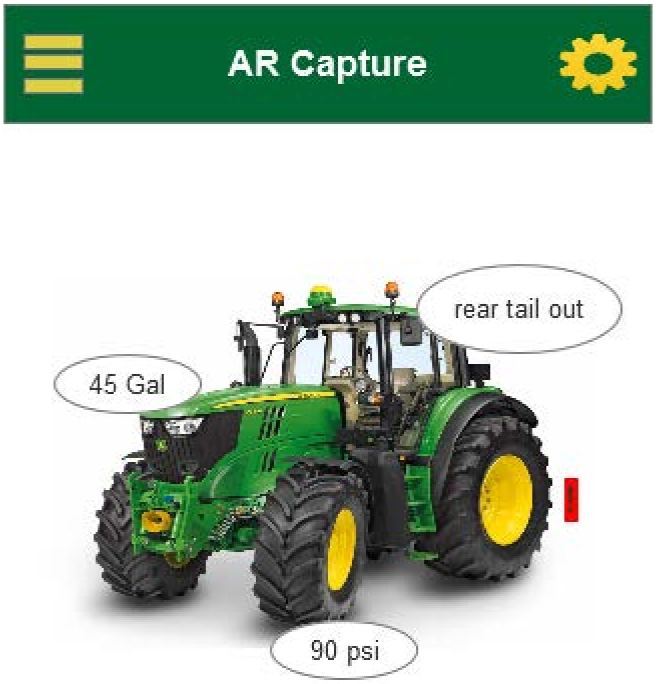


Figure 15

# Use Cases

## Use Case 1: Login

Primary actor: application user

Preconditions: the application has been launched successfully

Basic Flow:
The user launches the application, sees the splash screen, and then continues to the login screen. The user reaches the login screen and enters their credentials. The user is validated and saved through MyJohnDeere and is navigated to their home screen.

Alternative flow 1: Signup
The user cannot login because they haven’t created an account. The user chooses the signup button and then is brought to the signup screen. The user has to enter first, last, email, and password. Password requirements are 6-10 characters with one number and a capital letter. If the password requirements are not met the signup process doesn't complete and an alert pops up stating the password requirements aren't met.

Alternative flow 2: Login fails
The user launches the application,sees the splash screen, and then continues to the login screen. They enter their credentials and an alert pops up stating that their username or password is incorrect.

Alternative flow 3: Still logged in
The user launches the application, sees the splash screen, and then instead of being brought to the login screen the user is brought to the last screen they were currently on in their session. The session timeout for this will be 1 hour.

## Use Case 2: Vehicles (Home Screen)

Primary actor: application user

Preconditions: the application has been launched successfully and the user logged in.

Basic Flow:
The user has completed the preconditions and will be brought immediately to their vehicles screen. This screen shows all of the users vehicles in the form of a list view.

Alternative flow 1: Add vehicle
The user has completed the login process and selects the plus icon from their their vehicles (home) screen to add a new vehicle. The user is then brought to the add vehicle screen.

Alternative flow 2: Connect to current vehicle

The user has completed the login process and selects the connect button from one of their previously added vehicles in the list. The user will be prompted with a success or failure alert upon choosing connect.

## Use Case 3: Add Vehicle

Primary actor: application user

Preconditions: the application has been launched successfully and the user logged in. The user selects the plus icon from their home screen to add a new vehicle.

Basic Flow:
The user has completed the preconditions and will be brought to the add vehicle screen.The user can input a vehicle's name or the connection ID from which they are already accustomed to knowing. From there they select the tractor by clicking on the choose tractor button which will show all of the possible vehicles.

Alternative flow 1: User selects connect button
The Application will attempt to connect to connect the tractor via TCP connection over wireless. If successful an alert will be shown and connect will be disabled, if not successful an alert will be shown and button will still be enabled.

## Use Case 4: Augmented Reality

Primary actor: application user

Preconditions: the application has been launched successfully and the user logged in. The user selects the tractor vision icon from their home screen menu.

Basic Flow:
The user has completed the preconditions and will be brought to the tractor vision screen. The user is provided with a camera view of their surroundings ready to capture an image. The user snaps a picture and is presented with their augmented view of their vehicle.

Alternative flow 1: User selects a widget
Upon snapping the picture of the vehicle the user selects a widget of some machine information displayed. The user is brought to a new screen with tabs where they can see more information on the status of their vehicle. They can continue swiping horizontally to go through all of the widgets for that vehicle.

## Use Case 5: Alerts

Primary actor: application user

Preconditions: the application has been launched successfully and the user logged in. The user selects the settings icon from their home screen.

Basic Flow:
The user has completed the preconditions and will be brought to the alerts screen. All alert information on all of the users vehicles will be displayed. The user can search through each alert to gain more information on how to service the issue for their vehicle.

Alternative flow 1: User selects service
The user will be brought to a new screen specific to the alert they selected. Based on the type of the alert the user will be provided with options and guidance for how to contact John Deere about the issue or service the vehicle themselves.

## Use Case 6: Settings

Primary actor: application user

Preconditions: the application has been launched successfully and the user logged in. The user selects the settings icon from their home screen.

Basic Flow:
The user has completed the preconditions and will be brought to the settings screen. The user will be provided with a list of settings options and a logout option.

Alternative flow 1: User selects account information
The user will be brought to a new screen displaying their account information that is saved in MyJohnDeere.

Alternative flow 2: User selects application information
The user will be brought to a new screen with application information: version, date modified, creator.

Alternative flow 3: User selects logout
The user’s session will expire and they will be brought back to the login screen.

## Use Case 7: Bluetooth Connection to MTG

Primary actor: application user

Preconditions: the application has been launched successfully and the user logged in. The user selects the connect to tractor via Bluetooth from the home screen.

Basic Flow:
The user has completed the preconditions and will be brought to the bluetooth screen. The user will have directions displayed on how to pair with their tractor. A successful pair will happen if the device name shows up correctly as the user expected.

Alternative flow 1: User pairs unsuccessfully
The user will see an alert that pops up showing that there was no device paired. From here the user can try pairing again or navigate to somewhere else in the application.

Alternative flow 2: User pairs with incorrect device
The user will pair with a device but then see an incorrect device name. The user will click on the button to unpair the device and retry to pair with their correct device.

# Hardware/Software Specifications

## Hardware Specifications

### Mobile Device

The application will only support Apple iPhone devices running the supported iOS version. See the following links for specifications for each device.

- □ iPhone 4: https://support.apple.com/kb/SP587
- □ iPhone 4s: https://support.apple.com/kb/SP643
- □ iPhone 5: https://support.apple.com/kb/SP655
- □ iPhone 5s: https://support.apple.com/kb/SP685
- □ iPhone 6: https://support.apple.com/kb/SP705
- □ iPhone 6s: http://www.apple.com/in/iphone-6s/specs/

### MTG

The MTG supports WiFi and Bluetooth. Full specifications are not available.

## Software Specifications

The application currently uses iOS 7 and Vuforia. The application will replace Vuforia with Wikitude for augmented reality.

### Interface Specifications

The application does not have an external API. The internal API is currently unchanged from Phase I.

### John Deere

The application will use the John Deere API for login and integration with John Deere's cloud service, MyJohnDeere. See the following link for the full API.

https://developer.deere.com/

### Firebase

Free mobile app cloud service. See the following link for the full API.

https://www.firebase.com/

### Wikitude

The application will use Wikitude for Augmented Reality. See the following link for the full API.

http://www.wikitude.com/documentation/

### MTG

The application will communicate with the MTG. Details on the API are not currently available.

# Simulations and Modeling

- By come end of the first semester we hope to have Bluetooth communication working and integrated into the previous team's app
  - In addition to this we will hope to also show a new and improved user interface that is more tailored to John Deere's style guide a features
- Additionally monthly demonstrations will show progress for the application throughout.
- Regular monthly check-ins with John Deere will provide quick feedback to assist in our iterative development life cycles
- Our iOS mobile app will not be available on the app store, the apps location will be local and be able to be downloaded onto group members iPhone's for testing and usability protocols

# Implementation Issues and Challenges

Based on our requirements we believe our project has great potential for a good, finished product. Currently we already have major application functionality working thanks to the previous project team and we know how to move forward. Given our resources and knowledge the additional requirements we are asked to do can be done in the allotted time. Some concerns however arise with the timeline on when the new Yukon MTG will be completed with the low energy Bluetooth. We will be able to develop and test this without the actual MTG however problems may arise when put in place with the actual device. In addition the augmented reality portion of this application might not be what our client and our group hopes it to be. This is due to the augmented reality software being a new technology and not quite meant for our situation. Here is a table that better illustrates all of the issues and challenges of our project solution:

| Issue/Challenge | Evaluation |
|---|---|
| Bluetooth communication | There are currently some challenges and unknowns with the Bluetooth communication. iOS Bluetooth communication is one-way and there needs to be a standard for how our team will send data with John Deere's MTG. See appendix A. This can be through the creation of a GATT (generic attribute) Bluetooth profile or some easy work arounds through storing vehicle information on a file in a specific location. |
| Validation of user before pairing device | Upon the MTG sending information back to the user, the user will have a decision to pair with the MTG given the information they received. We just don't want anyone to be able to connect to the MTG, so we would want the user to enter a password of some form if they wished to pair their device with the MTG. The issue is that the scope of this project is designed when there's no cell signal. Ideally we would want to match the password they enter with a MyJohnDeere password, but we would be unable to process that request. |
| Augmented Reality software | For this project phase 1's group was unable to provide John Deere with a working augmented reality aspect for this application. Their augmented reality could only recognize small toy tractors. Our group has been looking at a new augmented reality software, Wikitude, in comparison to last year's choice of Vuforia to achieve better results. |
| Data storage | Last year's project phase 1 team managed to store all machine and account information on the mobile device. We are looking into doing the same thing with some additions. Initially we would want to store all of this data on the phone because there would be no cell signal, however once cell signal is available we want to transfer this data to Firebases's cloud services. |

Table 2

# Test Plan

## Unit Testing

### Access Control

In Xcode 7 you can develop with the new swift 2 language instead of objective C. Swift 2 brings some new great ways to unit test. The first is in access control. This is done by using the @testable attribute the unit test target has access to every internal entity of another module in the project. The @testable attribute allows some basic unit testing practices like the assertEqual function.

### Code Coverage

There is also another great feature for testing that comes with the new Xcode 7 version- code coverage. This process is done by some simple configuration in the Xcode scheme settings and it allows the developer to run all of their unit tests throughout the workspace at once and view the results. From there the developer can click on the individual unit test case and go straight to the source code to see how many times the test was called and the4 number of successes or failures occurred. This in combination with access control make unit testing pretty simple in Xcode.

## UI Testing

There has never been a great, built in way to test the UI in the past versions of Xcode, thankfully that has changed. The process is an expansion on Apples XCTest framework through adding three classes: XCUIApplication, XCUIElement, and XCUIElementQuery. UI testing is only available for iOS 9+. Testing the UI in addition to the unit test will cover a lot of ground for testing our application.

## Testing Process

Our testing process will be carried out in three major ways: testing the database entries, unit testing, and communication testing with the MTG.

### Database Testing

1. When a user account is created check the database
2. When a user adds a new vehicle check the database
3. When a user is logged out make sure the session variable is false
4. When an alert happens on a vehicle make sure the alert is stored

### Unit Testing

1. Check major components of UI to make sure updating is correct
2. Check logic within the models to make sure variables are correct
3. Check that the view controllers are navigating properly

### MTG Testing

1. Make sure Bluetooth connection has correct name when paired
    a. Includes correct VIN and model number

2. Make sure MTG is relaying all of the data points to the mobile device

# Test Results

## Unit Testing

### Access Control

Information passed to other view controllers within the application worked perfectly. All information was consistent and allow of the @testable statements passed.

### Code Coverage

We did not have time to test our application in this way

## UI Testing

Overall the UI testing went extremely well unfortunately we do have one error. Sometimes after leaving the Tractor Vision (augmented reality) view it will cause the application to freeze once returning to this view. We suspect we need to handle the view's super method videdidreload differently. It seems that the EaglView aspect in the Wikitude SDK isn't reloading properly. This is the component that is used for the augmentation and image recognition that uses the iPhone's camera.

### Database Testing

1. PASSED
2. PASSED
3. PASSED
4. PASSED

### Unit Testing

1. PASSED
2. PASSED
3. PASSED

### MTG Testing

We did not receive the MTG from the client due to them not having their newest version ready. We know how the MTG sends and receives data so we created a mock MTG within another iPhone application.

# Validation and Acceptance Test

## Reliability

The reliability aspect of our validation and acceptance will be a combination of our testing plan in pair with other reliability testing. The first reliability testing will take place when the phone has access to cell service to see if the data is transferred to MyJohnDeere properly and how long that process takes. Other testing will be done with the MTG communication. We will test to make sure that the MTG will always be able to woken up via Bluetooth and that the wireless is set up accordingly. We will also test to see exactly how far away the user can be from the vehicle to remain connected to the MTG.

## Scalability

There isn’t much we can do for a scalability test for our project. The only we see fit for this is to have the user perform a high number of tasks such as adding a lot of vehicles to see how the system reacts. This scenario relates more towards the application operating in dealer mode having a large number of vehicles on their account.

## Usability

Our usability testing will be done through validating that our use cases our functioning properly. This means that based on the action done by the user that they’re being presented with the proper screen, information, and dialogue. In addition we will check that the customer is okay with the UI as far as functionality, look, and feel.

## Security

Our procedure for secure will be to demonstrate this to the customer directly. This will be done to show that the user can only connect to their added vehicles and they won't accidently or purposely connect to another’s vehicle.

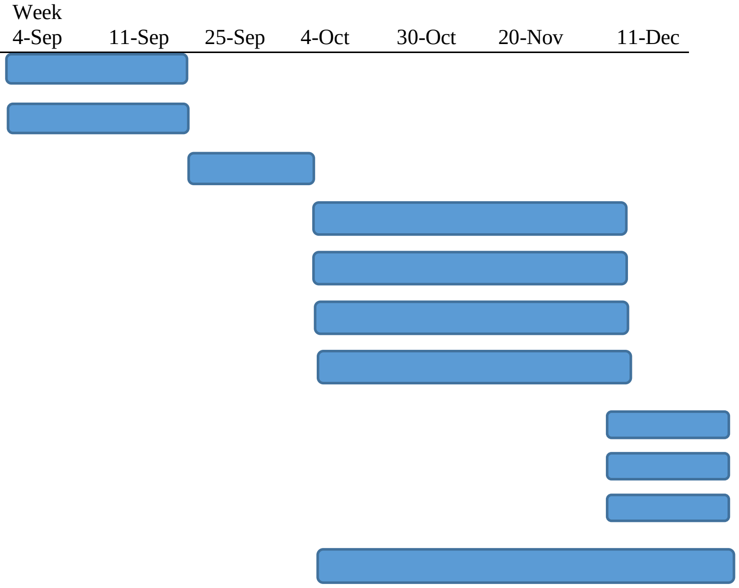
## Maintainability and Configurability

We will leave this up to the customer on how they will provide security patches or updates for the application. The configurability aspect will be validated by demonstration on how the customer can modify the application to better suit their needs in the future.

# Software Design

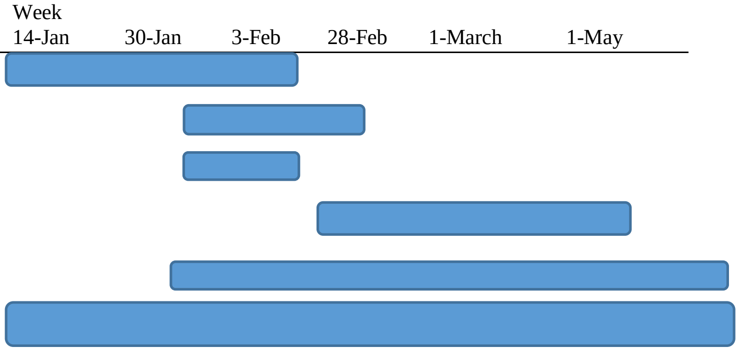
## Software Design Schedule Semester I

| Activity | Start | Duration |
|---|---|---|
| Understand Phase I | 4-Sep | 7 |
| Requirements | 4-Sep | 7 |
| Design | 11-Sep | 7 |
| Bluetooth | 4-Oct | 21 |
| Wi-Fi | 4-Oct | 21 |
| User Interface | 4-Oct | 35 |
| Data storage | 4-Oct | 35 |
| Augmented Reality | 11-Dec | 35 |
| MTG Communication | 11-Dec | 35 |
| Integration with Bonjour | 11-Dec | 35 |
| Testing | 4-Oct | 56 |



## Software Design Schedule Semester II

| Activity | Start | Duration |
|---|---|---|
| Bluetooth fixes | 4-Sep | 37 |
| Cloud | 3-Feb | 25 |
| User Interface | 3-Feb | 25 |
| Data storage | 28-Feb | 35 |
| Augmented Reality | 30-Jan | 75 |
| Testing | 14-Jan | 56 |



# Conclusion

This project is going rely heavily on the communication with the MTG, thus this is our main focus to get the application running efficiently. Next our focus is on the augmented reality workflow and design. Our goal is to make the app as easy for the user as possible, however given that augmented reality software today isn't still super robust we may have to be creative in how we incorporate it into our project. Through our documentation we believe we have a great base plan to continue and modify development for this application. All of the resources for this project are accounted for, and at the moment we do not think there will be any additional resources needed to start development. Based on our extensive research and meetings we believe to have some great ideas for the augmented reality and improved MTG connection portions for this project.

# References

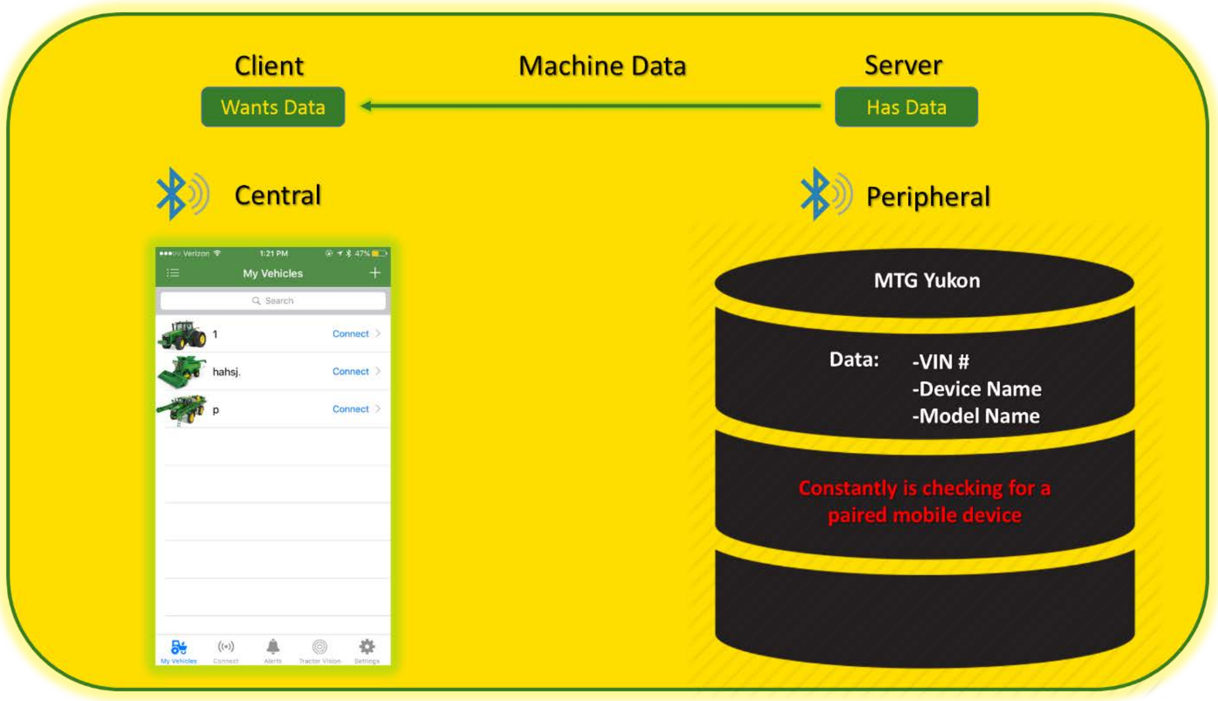
"Bluetooth Interoperability and Profiles." *Custom Profile Development*. Web. 14 Oct. 2015. <https://developer.bluetooth.org/DevelopmentResources/Pages/Custom-Profile-Development.aspx>.

"May1505 - My John Deere App - Project Documents." *May1505 - My John Deere App - Project Documents*. N.p., n.d. Web. 14 Oct. 2015. <http://may1505.ece.iastate.edu/?p=docs>.

"Services | Bluetooth Development Portal." *Services | Bluetooth Development Portal*. Web. 14 Oct. 2015. <https://developer.bluetooth.org/gatt/services/Pages/ServicesHome.aspx>.

"Telematics and Work." (1994): Web. <https://www.deere.com/common/docs/products/equipment/telematics/brochure/john_deere_mtg.pdf>.

# Appendix A



Client

Wants Data

Machine Data

Server

Has Data

Central

Peripheral

MTG Yukon

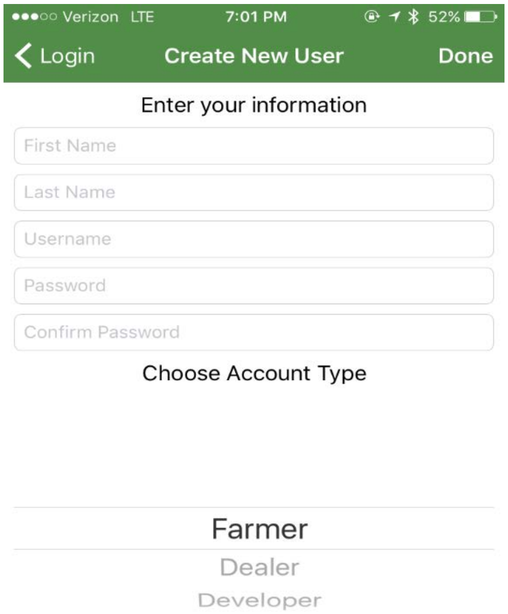
Data: -VIN #
-Device Name
-Model Name

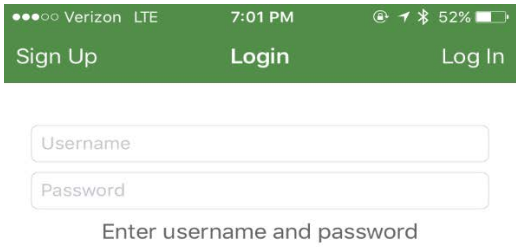
Constantly is checking for a paired mobile device

# Appendix B Operation Manual

## Step I Account Creation

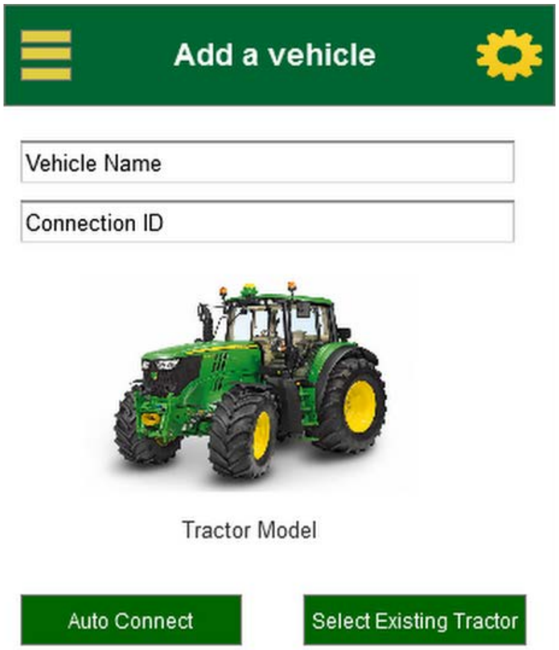
The first time the application is launched from download the user will have to create an account. Currently we aren't validating the user with John Deere's database due to privacy issues so that will be implemented on their end. For now anyone can create an account and login. Once the account is created if there are no errors in password constraints the user is brought to the login page.





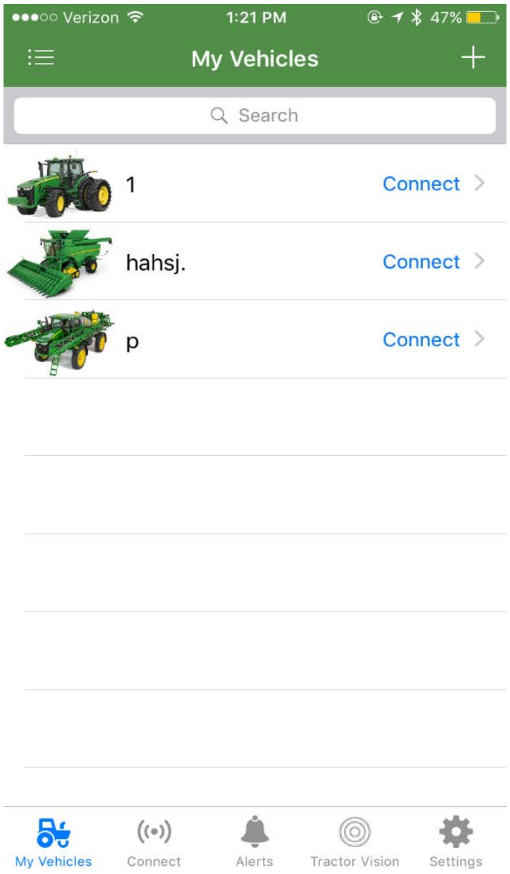
## Step II Add a Vehicle and Connect

Once successfully logged in the user can now create a vehicle and connect to it. Creating a vehicle is easy, with three types of tractors to choose from, simply select the type and then name it. Once the vehicle is added the user can connect to it by simply just clicking on the MTG Bluetooth device connect button.



## Step III View alerts

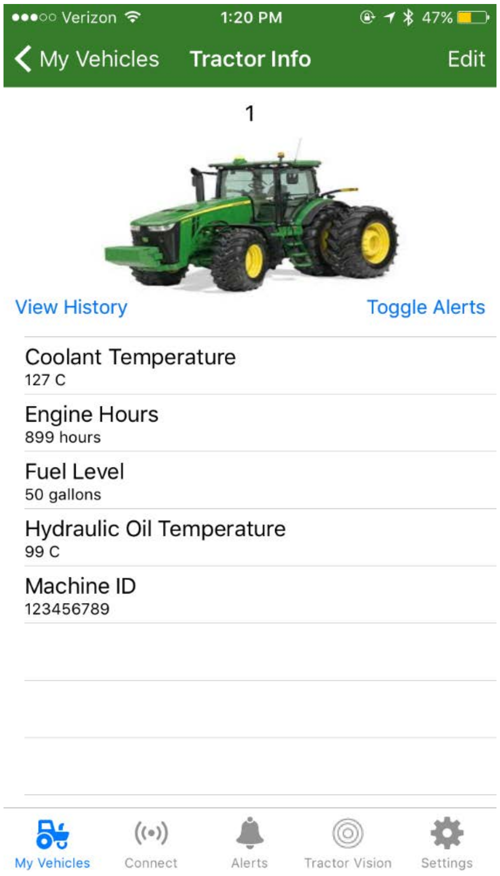
Once successfully connected to a vehicle the alerts page will provided the user with information regarding each of their vehicles they have connected to using the my vehicles page.



1 Connect

hahsj. Connect

p Connect



1

Coolant Temperature
127 C

Engine Hours
899 hours

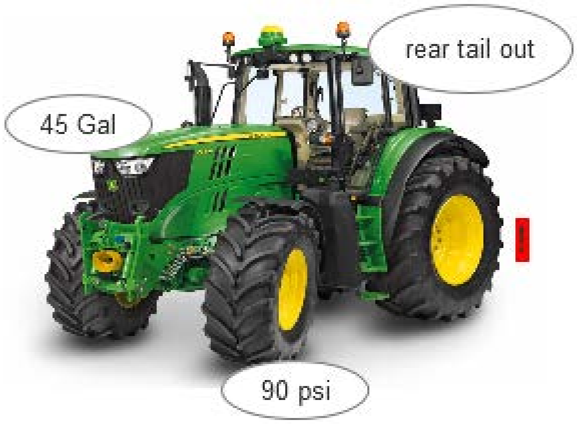
Fuel Level
50 gallons

Hydraulic Oil Temperature
99 C

Machine ID
123456789

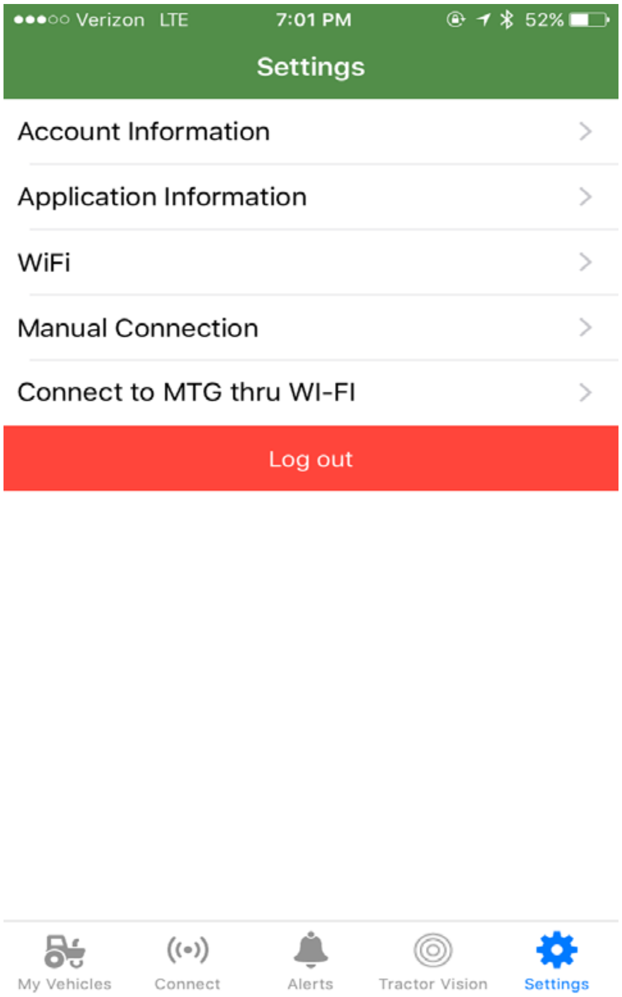
## Step IV Tractor Vision

The other way to view your machine’s alerts is through the augmented reality tractor vision page. This provides the user with a simple point and shoot aspect to see what’s going on with their vehicle instantly. Simply aim the camera at the tractor and once the “recognized,” banner is displayed alerts will be overlaid on top of your tractor displaying any issues.



## Step V Logout and Settings

If the user wishes to logout, we currently have no timed logout implanted, the user can simply navigate to the settings page and click the logout button or view information about the application.



# Appendix C Previous Versions

## Wi-Fi Alteration I

Background:

- Originally the plan was to have Bluetooth be the initial control connection and then the data connection would be using Wi-Fi and Bonjour on the iOS side.

Resolution:

- This idea was scratched by the client because Bluetooth would use less power on the MTG side and still get the job done.

## Bluetooth Alteration I

Background:

- Originally the scope of our project was designed to have Bluetooth establish the initial connection and then Wi-Fi would perform the data transfer from the MTG to the mobile app. The Bluetooth connection would be done using the iOS BLE (Bluetooth Low Energy) library to communicate with an Ubuntu 12.14 OS that would be implemented on the MTG.

Resolution:

- The client realized there was a bug imbedded in the Linux library that's used to communicate with BLE iOS devices so this idea for the Bluetooth communication was scratched.

## Bluetooth Alteration II

Background:

- The next proposal for Bluetooth was to have the client implement a TI (Texas Instruments) on the MTG to house the Bluetooth profile. This would involve us to use the Bluetooth developer studio to create an objective C file that would be able to communicate with the TI board. This would be completely different than using the iOS library because the iOS library does not support this type of BLE profile.

Resolution:

- This idea was scratched by the client again due to them not being able to implement it.

## Cloud Storage

Background:

- The original idea for our application was to use John Deere's cloud storage through their MYJohnDeere API. We were unable to achieve this due to privacy issues so we decided to use a proof of concept reproach instead.

Resolution:

- We decided to use a free open source cloud storage for mobile applications called firebase. Their API provided use with an easy to use service to store and retrieve our data.

# Appendix D Other Considerations

## Proof of Concept

Our mobile application is a proof of concept due to limitations within the project scope. The main limitations for our project were due to client issues. They were unable to provide us with their newest version of the MTG for testing to use the BLE profile and communication protocol. We emulated this by creating our own BLE iPhone test application in correspondence with John Deere on how the data would be sent by the actual MTG. This was successful. Our next limitation was with the cloud services. John Deere was unable to allow us access to their cloud services due to privacy issues as they didn’t have a test environment we could use. Instead we used the free Firebase cloud service and were successful. In conclusion even though we were faced with some major obstacles we still created a successful application that met all of the requirements.

## IoT

Our application is a great example of an Internet of Things application. We meet all the major requirements, mobile access, cloud storage, and wireless connectivity. IoT is becoming more and more prevalent in our developing world so being a part of a team that developed this type of application was really awesome.