May15-05 Design Document

Augmented Reality Tractor Maintenance Application

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

Figures

Tables

Project Introduction

Project Statement

Currently the John Deere API can serve some maintenance information on their tractors from their server that has been downloaded from the tractors connection in the field. However, a farmer in the field may not have connection to cellular or wireless service needed to reach the John Deere servers; this application is intended to fill that gap by connecting directly to the tractor via wireless to a device on board the tractor.

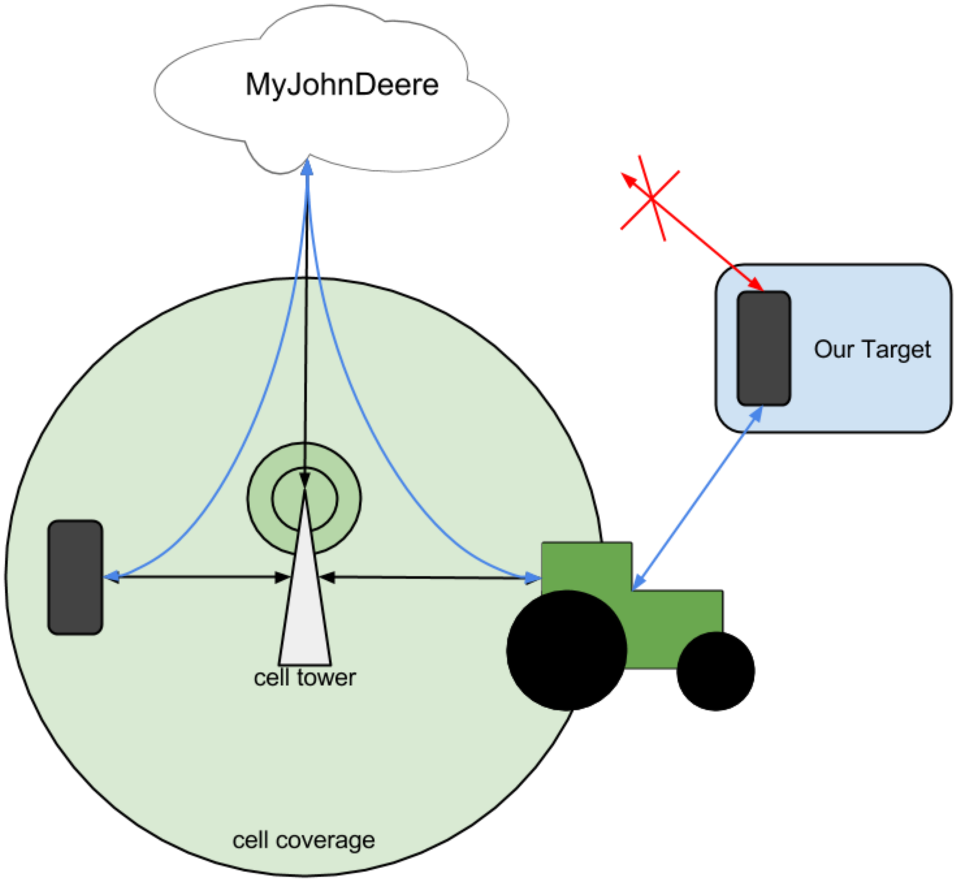

Figure 1: Tractor Connection Problem

Deliverables

The final product for this project will be a mobile application for the Apple iOS platform which runs on iOS 7 and up.

A prototype will be available for demonstration at the end of the first Semester to show to ISG and will work as a proof of concept for future development.

Every month a demo will be available to show ISG and separate tag in the git repository will be created to benchmark these demos.

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 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:

Parameter to be displayed if user is a Farmer

Parameter	**Create an Alert if value is above (or below) this value**
Fuel Level	Less than 100 gallon
Machine Hours	More than 2000 hrs
Coolant Temperature	60C
Hydraulic Oil Temperature	95C
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

Table 1: Farmer Parameters and Thresholds

Parameter to be displayed if user is a Dealer

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: Dealer Parameters and Thresholds

The following additional features are expected.

Non-functional Requirements

- Application will run on iOS 7 and up.
- This application must make 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.
- The applications operation must not expose or reveal any sensitive information to anyone but an authenticated user..

System Design

System Requirements

This application will run on Apple's proprietary iOS mobile operating system starting from version 7 (iOS 7) and up (current highest version is iOS 8). This application is intended for use on iPhones running appropriate OS versions.

Devices using this application should have a working camera to use any Augmented Reality or image recognition features to identify the vehicle.

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.

System Analysis

Feasibility

There are several resources available which make this project very feasible:

Augmented Reality:

We will likely use Vuforia as a framework for doing Augmented Reality. It is free to use and offers options for developers to use their image recognition for free. Additionally their service widely accepted, used, and documented which will aid our development process to no end. This also handles much of the Camera and Image Recognition requirements for this application.

iOS Development:

Xcode is readily available in the TLA in Coover on ISU's campus. Development for iOS on Xcode is greatly facilitated such that learning Objective-C for development is made simple and feasible.

External APIs

The MTG's API will be provided by ISG which we expect will be well supported as we will be working directly with it. Thus we expect no issues with this. The MyJohnDeere API is documented on its resource site and appears to provide all necessary information at this time. Vuforia's API is widely used and well documented; Vuforia is expected to be the most difficult to find support for but due to its popularity does not appear to pose a problem.

System Block Diagram

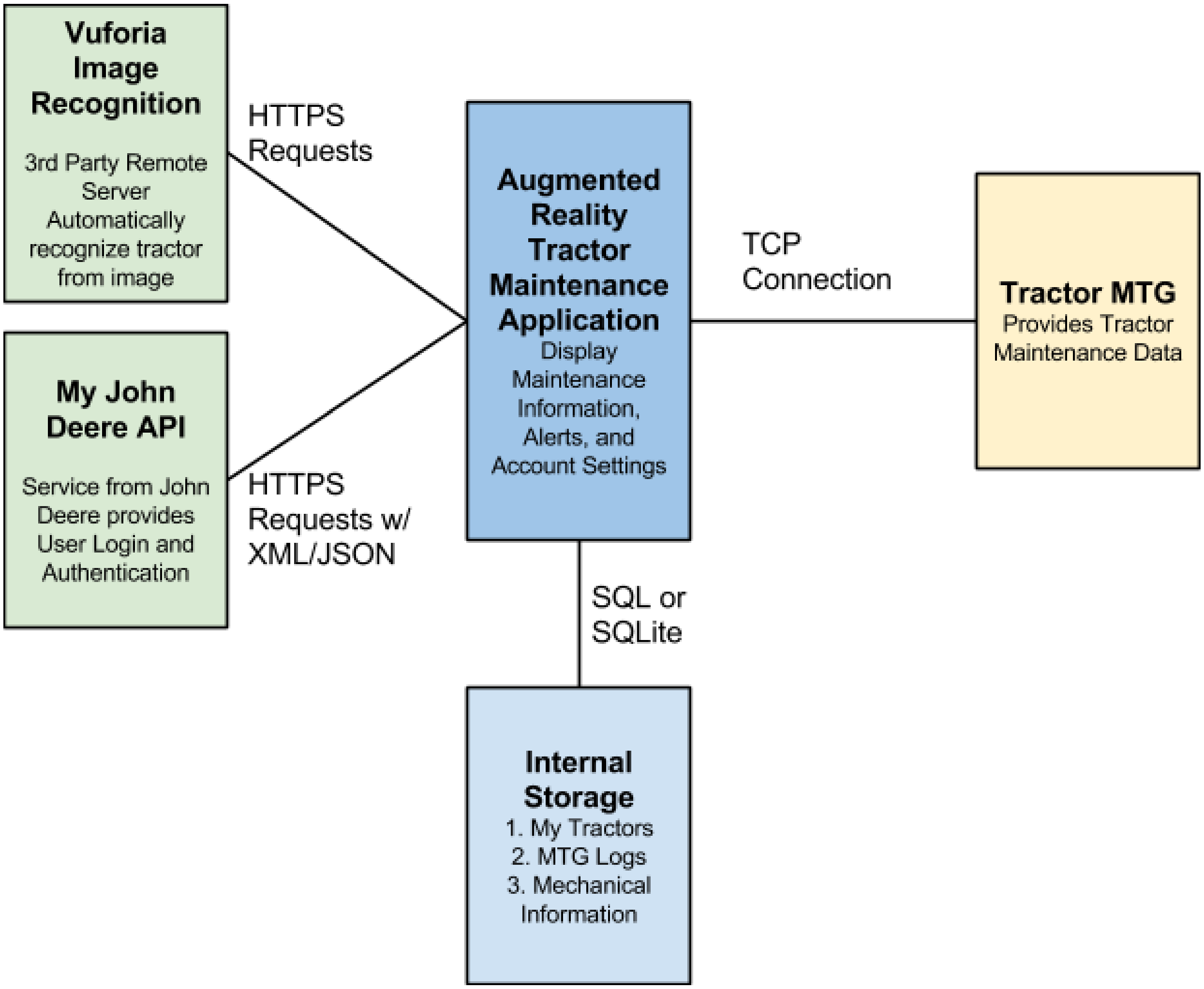
Vuforia Image Recognition
3rd Party Remote Server Automatically recognize tractor from image
HTTPS Requests
My John Deere API
Service from John Deere provides User Login and Authentication
HTTPS Requests w/ XML/JSON
Augmented Reality Tractor Maintenance Application
Display Maintenance Information, Alerts, and Account Settings
TCP Connection
Tractor MTG
Provides Tractor Maintenance Data
SQL or SQLite
Internal Storage
1. My Tractors
2. MTG Logs
3. Mechanical Information

Figure 2: System Block Diagram

Application Block Diagram

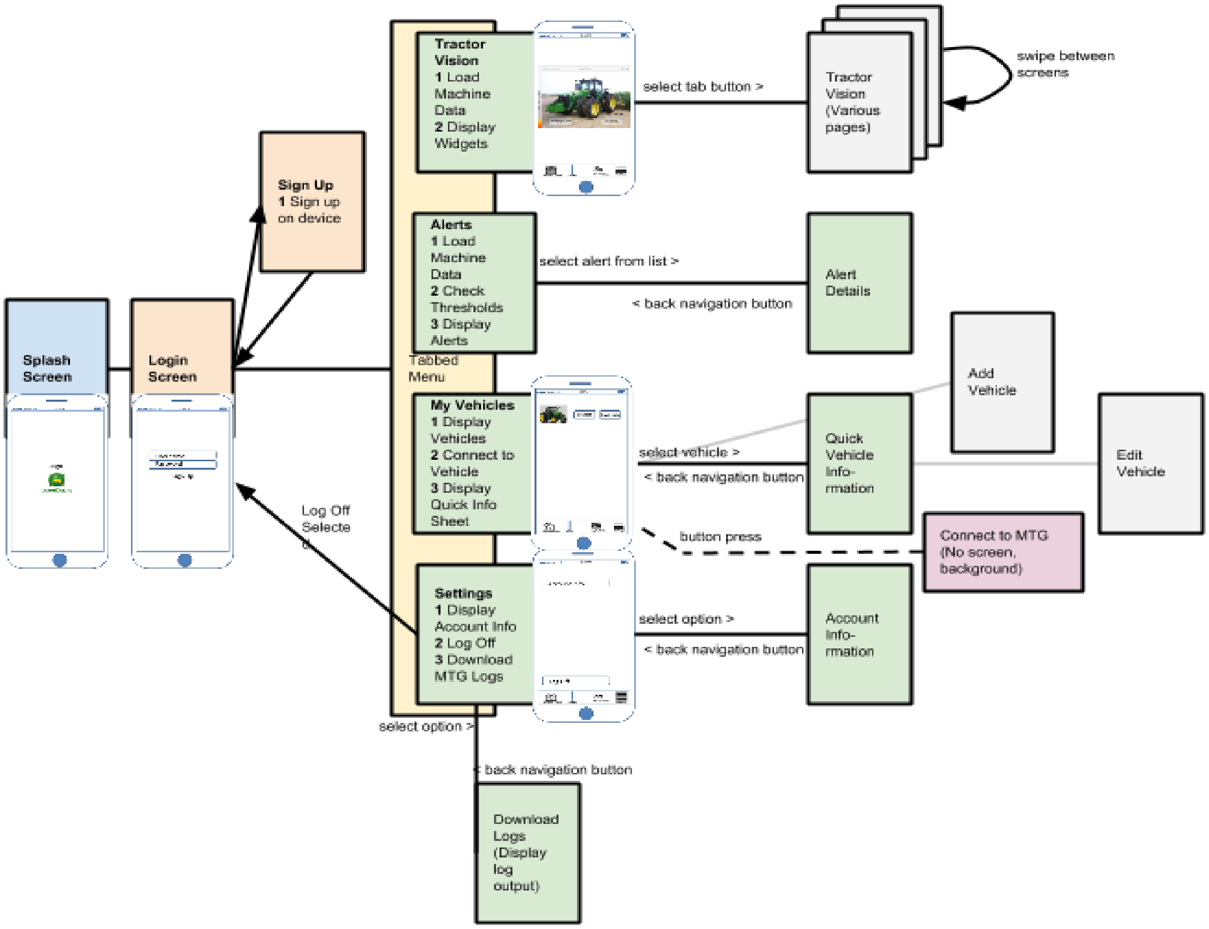

Figure 3: Application Screen Flow

Functional Decomposition

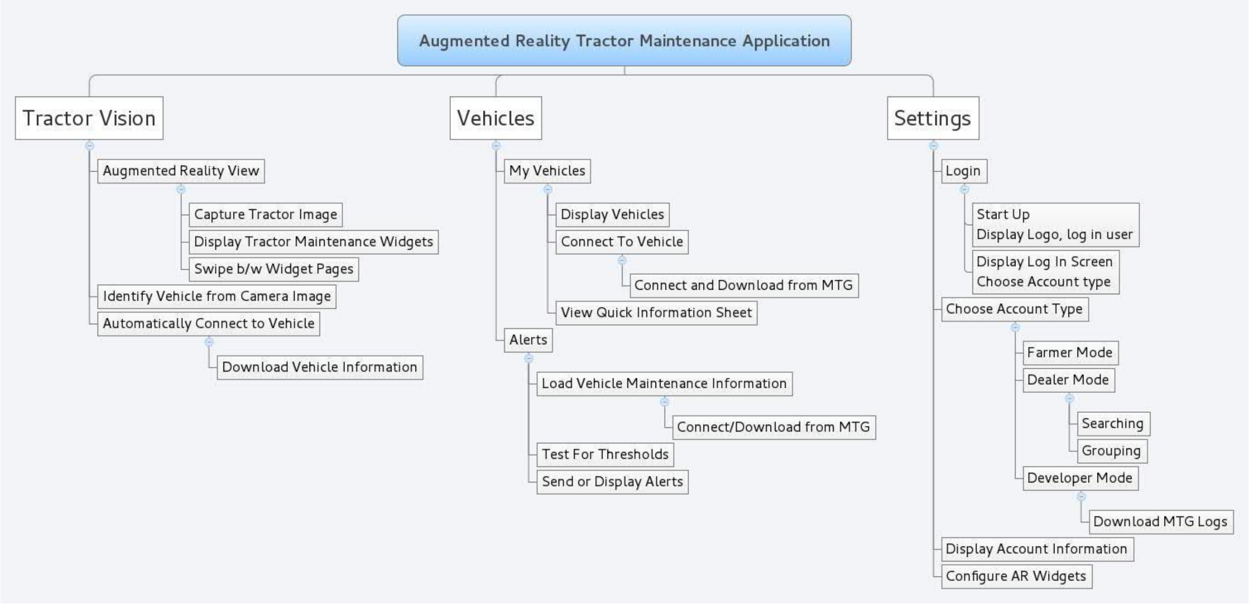

Figure 4: Functional Decomposition

Use Cases

Use Case 1: User Logs In

Actor: Application user

Primary flow:

User starts the application and is presented with a splash screen for a 2 seconds before the log in screen is presented. User enters username and password and chooses log in. User is validated using MyJohnDeere and is brought to the default screen. The application records the login status.

Alternative flow 1: User log in fails

Instead of being brought to the default screen user is presented with a message that login has failed and is brought back to the log in screen to resume primary flow.

Alternative flow 2: User chooses 'Sign Up'

User is transferred via http link to the MyJohnDeere sign up page.

Alternative flow 3: User is already logged in

If the user has already logged into the application and the application remembers the users data, the user will skip the login screen and transfer directly to the application screen.

Use Case 2: View Vehicles

Actor: Application user

Primary flow:

User chooses the 'My Vehicles' tab button and is presented with a list of tractors which are registered to their account.

Alternative flow 1: User selects to view info on the tractor

If the user selects the tractor they will be transferred to the Tractor's Quick Information page with details about the tractor's registered information.

Alternative flow 2: 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 3: View Settings

Actor: Application user

Primary flow:

User chooses the 'Settings' tab button and is presented with a list of settings options.

Alternative flow 1: User selects account information

Information about the user account with MyJohnDeere is displayed.

Alternative flow 2: User selects application information

Options specific to the application are displayed here, used for debugging and checking version number, etc.

Alternative flow 3: User selects log out

The user's status is changed to logged out and the application is brought back to the login screen.

Use Case 4: View Alerts

Actor: Application user

Primary flow:

User selects the 'Alerts' tab button and is brought to the alerts page. The application checks parameter values in data downloaded from the MTG against a list of thresholds and displays alerts if they are present.

Alternative flow 1:

User selects an alert and is brought to a page with details about the alert and recommendations on how to fix.

User Case 5: View Augmented Reality

Actor: Application user

Primary flow:

User selects 'Tractor Vision' tab button and is brought to the Augmented Reality screen view. The camera is started and begins to try and recognize a tractor on the screen. If it recognizes a tractor it will attempt to display information on that tractor with overlaid widgets.

Alternative flow 1: User freezes the screen

If the user taps on the screen they may freeze the screen so they do not have to hold the camera up constantly. All widgets stay present but the camera no longer shifts.

Alternative flow 2: User swipes through screens

If the User swipes across the screen they may view different screens containing different data widgets much like a home screen with application icons/widgets.

Detail Description

I/O Specifications

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 UDP messages may be used to probe for available vehicles to connect to and to request identification information. Once a tractor has been identified and selected for connection TCP messages will be used to make requests to the tractor. The functions and services on the MTG will be provided by ISG through these TCP requests.

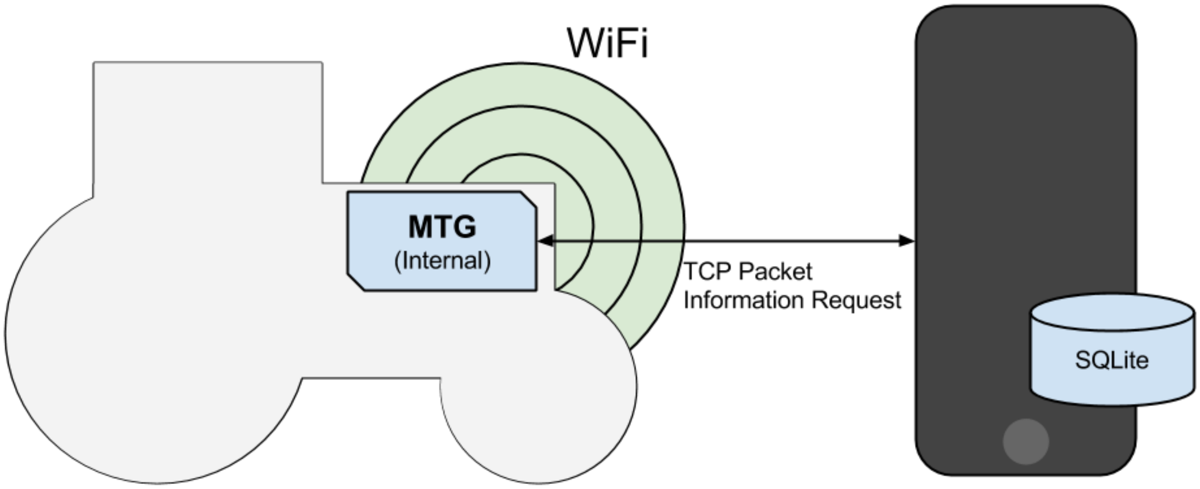

Figure 5: Basic Communication

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

For the use of some frameworks and libraries, such as the Vuforia Augmented Reality library, it may be necessary to connect over cellular or wireless in order to make image recognition requests to the Vuforia server.

Interface Specifications

Primary Application Modules

This application does not provide an API for use outside of the application. For information on internal interfaces see Appendix A.

John Deere API

The Application will interface with the MyJohnDeere API in order to provide user log in and authentication. This will all be done over HTTPS in order to maintain user information security and privacy.

https://developer.deere.com/#/home/landing

Vuforia API

This application will make use of the Vuforia Augmented Reality software framework/library for iOS.

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. This API is to be provided by ISG once it is ready.

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.

See these references for further information on the iPhone hardware specifications.

iPhone 4:

http://www.everymac.com/systems/apple/iphone/specs/apple-iphone-4-verizon-cdma-specs.html

iPhone 5:

http://www.everymac.com/systems/apple/iphone/specs/apple-iphone-5-a1429-cdma-lte-sprint-verizon-specs.html

iPhone 6:

http://www.everymac.com/systems/apple/iphone/specs/apple-iphone-6-a1549-4.7-inch-cdma-verizon-north-america-specs.html

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

MTG hardware specifications will be provided by ISG.

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.

Currently Vuforia is the primary option for AR software for iOS considering it is freely licensed to use and widely accepted and documented. Additional software frameworks and libraries may be needed to facilitate image recognition in the case that there is no connectivity to conduct image recognition via Vuforia's services. Given that this application is intended to aid farmers in situations where they have limited connectivity, this is considered a likely event.

See more information on Vuforia Developer Portal here:

https://developer.vuforia.com/

Modelling

- An initial prototype displaying basic functionality and proof of concept to be completed by the end of the first semester.
- Additionally monthly demonstrations will show progress for the application throughout.

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.

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 useful Unit Testing capabilities as of version 5. For every major modular interface Unit Tests should be provided for each method which will be used by the application. See more information in the interface specifications above and Appendix A for internal module interface information.

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.

Further Resources and Documents

The human interface will interact in accordance with Apple's design guidelines and John Deere's trademark guidelines detailed in the links below. The interface must emphasize simplicity and ease of use to in order to meet the applications purpose and targeted user base.

The application will be interfaced with via the iPhone’s capacitive screen and therefore must maintain a human interface easily operated by touch.

Minimum expected screen size is 3.5 (iPhone 4) to 5.5 (iPhone 6 Plus). Minimum resolution on the LCD screen is expected to be minimum 960×640 (iPhone 4) to 1920×1080 (iPhone 6).

For more information on designing for iOS see:

https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/

Web resources for development

John Deere Developer Resources:

https://developer.deere.com/#/home/landing

Additional Guidance

Information on John Deere’s trademark guidelines:

http://www.deere.com/en_US/docs/Corporate/citizenship/john_deere_tm_guides_sponsorships.pdf

Summary and Conclusion

This application is fully expected to be complete within the two allotted Semesters for this project and with careful selection of 3rd party software the project should come well within the $1000 budget.

Appendix A – Internal Module Interfaces

This appendix describes classes used internally for the efficient use of the application.

User Manager

The User Manager provides an interface for interacting with all user data stored in the database.

All requests to create, edit, and access user information should be processed through the user manager.

Tractor Manager

The Tractor Manager provides an interface for interacting with all vehicles stored in the database.

All requests to create, edit, and access vehicle information should be processed through the tractor manager.

Database Manager

The Database Manager provides an interface for all interactions with the internal SQLite databases. Two databases exist currently on the device: AppCore.db and AppData.db.

AppCore contains application settings essential for app usage and configuration. To use the AppCore database, the user need only use the standard 'init' initializer. Specific AppCore functions are available through the manager—however these functions are not useful when initializing another database and may fail if attempted.

AppData contains the dynamically growing application information available through the user interface. To use the AppData database is created if it does not exist on initialization. To see information on initializing for specific databases see the interface documentation with the source to initialize with database name 'AppData.db'. Individual managers are responsible for defining their own tables.

Field Arrays are used to specify get and set. When selecting from a database field arrays with the 'get' values set are used and provided as opts. For updating and adding field arrays with set values are used. Ensure types are specified appropriately to avoid errors.

Maintenance Manager

The Maintenance Manager provides an interface for interacting with all tractor maintenance information stored in the database.

All requests to create, edit, and access maintenance information should be processed through the maintenance manager.