6.4.3 Availability
It is expected that the tablet application be available regardless of the presence of WiFi or Data network. With the exception of downloading “new” bridge data, all functions of the 3DWBIS will be available for the user, including performing and saving a Bridge Inspection and taking photos. These data will just not be saved to the remote server until connectivity is reestablished.

6.4.4 Security
The WBIS will require users to authenticate themselves through a unique username-password combination. These login credentials will be cached on the device for a fixed period of time (e.g. 24 hours) and used to authenticate requests to submit or receive data from the remote server. The login credentials will be encrypted before being sent in any request. The remote server will compare the encrypted credentials against those stored in its secure database and, if they do not match, will reject the associated request. While the WBIS will not be demonstrated over HTTPS, should the customer later decide to implement HTTPS, the software design as elaborated in these requirements will not preclude its implementation.

6.4.5 Maintainability
This pilot software system as designed and implemented by MTRI is considered in beta test after the initial year of development. MTRI will be the sole point of support for the system front end (Tablet Application) and back end (Server / Database Interface) until the close of the current project phase. MTRI will work closely with MDOT IT/Database personnel to propagate the database extensions and middleware functionality to an MDOT in-house capability by close of project.

6.4.6 Portability
The final version of the application will look and feel the same, to the greatest extent possible, across devices. Specifically, the application's user interface, user interaction, and behavior will not vary between the Android and iOS implementations.

The beta (or first year version) will only be supported on Android devices, due to the resource constraints imposed by developing a native 3D application that were not foreseen at the start of the project.

6.5 Design Constraints
The development team at the Michigan Tech Research Institute does not have access to the Oracle license required to host and manage an Oracle enterprise database such as the existing Bridge Management System (BMS). As the development team desires to work with a representative database during active development and for testing purposes, the development team will need to create a surrogate Bridge Management System (sBMS). The sBMS is intended to mirror the BMS database schema but also to extend it where necessary to support new
functionality (e.g. the storage and retrieval of field photographs). The sBMS will be contained by a PostgreSQL database management system instance. In designing the database middleware, the development team will select software that is known to work or will be made to work with both PostgreSQL and Oracle databases.

6.6 Logical Database Requirements

As the Wireless Bridge Inspection System is intended to read from and write data to the existing Bridge Management System (BMS), the back-end database required to support the application is a relational enterprise database. The surrogate Bridge Management System (sBMS) will be a relational database modeled after the existing BMS database schema. The database management system (DBMS) selected for this purpose, PostgreSQL, is an enterprise-level DBMS that along with the existing BMS supports the following features thought to be necessary to support the software requirements:

- ACID compliance (atomicity, consistency, isolation, durability)
- User and role permissions at the object level
- Triggers
- Procedural languages
- Spatial object support
## 7 APPENDIX A

### 7.1 Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition as used in this requirements document</th>
</tr>
</thead>
<tbody>
<tr>
<td>AASHTO</td>
<td>The American Association of State Highway and Transportation Officials (AASHTO) is a standards setting body which publishes specifications, test protocols and guidelines which are used in highway design and construction throughout the United States.</td>
</tr>
<tr>
<td>Appraisal rating</td>
<td>During a routine inspection, a rating that describes the condition of bridge components compared to a new structure built to current standards.</td>
</tr>
<tr>
<td>Bridge Model Schema</td>
<td>A project-defined data representation of a bridge that is specific, complete, human-readable, and allows the Renderer to “draw” a 3D representation of the bridge. Implemented as an XML document.</td>
</tr>
<tr>
<td>Condition rating</td>
<td>During a routine inspection, a rating that describes the existing condition of in-place bridge components compared to their original, as-built condition; this includes the 4 NBI Condition Ratings.</td>
</tr>
<tr>
<td>CoRE Elements</td>
<td>&quot;Commonly Recognized&quot; Bridge Elements as defined by AASHTO.</td>
</tr>
<tr>
<td>Extended inspection forms</td>
<td>All other forms currently in the Bridge File, namely Fracture Critical, Fatigue Sensitive, Underwater, Other Special, Damage, Scour Action Plan (which may be supported in future).</td>
</tr>
<tr>
<td>MiBridge</td>
<td>A web based structure management application allowing Bridge Owners, Engineers, Inspectors, Consultants, and Managers to view and enter information for bridge and culvert assets across the State of Michigan.</td>
</tr>
<tr>
<td>Previously downloaded report</td>
<td>A PDF read-only version of a previously submitted Report, as from the MiBridge &quot;Bridge File&quot; document.</td>
</tr>
<tr>
<td>Routine inspection</td>
<td>The inspection performed per-bridge at least once every two years (but sometimes more frequently) that includes the CORE Element Inspection, the Structure Inventory and Appraisal, and the NBI.</td>
</tr>
<tr>
<td>Scoping Survey</td>
<td>Evaluating a bridge for various repair alternatives and recommending the most economical rehabilitation or treatment, then developing a scope of work and cost estimate for the selected alternative.</td>
</tr>
<tr>
<td>Standard inspection forms</td>
<td>The three MDOT forms whose inputs are supported by this WBIS; the three &quot;main&quot; forms known as CORE Element, Bridge Safety Inspection Report, and Structure Inventory and Appraisal Forms.</td>
</tr>
<tr>
<td>3d Bridge Model</td>
<td>A 3D representation of the bridge to serve as a method of...</td>
</tr>
</tbody>
</table>
displaying and navigating to individual CoRE bridge elements. It is not meant to be a solid model of the bridge, nor an engineering facsimile, but a manipulatable analog of the bridge that the Inspector can mark up, localize conditions on, and attach photos to.

7.2 Coordinate System for Bridge Rendering

The Application maintains an internal coordinate system for rendering the bridge elements (parts) in the correct relative position and relation to each other. The origin of this local reference frame is one of the (symmetrical) bottom-right corners of the deck surface, with the X direction following the span-direction or on-bridge traffic direction, and the Y direction in the lateral, with positive Y values starting on the bridge. These coordinates are not typically used by the inspector, but provides the Application with a consistent rendering of bridges in a local reference frame.

If the default labels are incorrect, the Inspector will have the opportunity to update them during an inspection action.
Coordinate Systems – a Metrical one for internal calculations / rendering, and the “labeling convention” that matches current MDOT bridge inspection practice

- **Internal Bridge Coordinate System**
  - Orientation ‘quantized’ to cardinal directions
  - +x is East (in E/W bridges)
  - +y is North (in N/S bridges)
  - All coordinates in feet

  "z" is implicit – abutments/piers/columns rendered perpendicular to the XY plane

- **Inspection Convention Coordinates – for navigation / labeling only**
  - If bridge is E-W, inspection starts from West
  - If bridge is N-S, inspection starts from South
  - Elements are numbered from starting direction and if 2D, clarified with second direction Eg Beam South 1 West 1, always starting from S/W “corner”

Figure 3: Coordinate System for Bridge Rendering
8.5 Implementation Action Plan

**Project Title:** Wireless Data Collection Retrievals of Bridge Inspection/Management Information

**Project Number:** 2013-0067, Auth. No. 2 (R1, R2)

**Principal Investigator:** Colin N. Brooks

**Description of the Problem:**

Currently, MDOT is faced with the task of inspecting its entire bridge inventory using a paper form process. Considering that every bridge must examined every two years, or more frequently as condition demands, this is a time-consuming process. The size of the task has only increased in recent years given the FHWA's demands for compliance with the National Bridge Inspection Standards by checking the completeness and accuracy of bridge data. The collection of AASHTO element-level data further increases the demand on inspectors.

To meet and exceed the requirements of the new regulations, MDOT is interested in incorporating mobile digital technology into the bridge inspection process to increase its efficiency and reliability. By switching to a digital inspection process, MDOT will eliminate the need for manual data transcription from paper to digital. The fact that most mobile platforms come equipped with a built-in camera means that MDOT can streamline the process of associating images with defect information, which otherwise must be done manually.

To facilitate the collection of element-level data, MDOT wishes to use interactive 3D bridge models that the inspectors can mark up with defect information. The inspection program will then automatically perform tallying to generate quantity information for element-level reporting. Additionally, an interactive 3D format for bridge inspections lends itself to bridge management decision support, since it provides detailed information about defects and their location on a bridge that is not captured by element-level reporting alone. Tracking individual defects also provides information on how those defects deteriorate over time, further aiding in management decision-making.

**Major Discovery:**

During this project, MTRI developed a mobile application (the 3D Bridge App) for displaying and interactively marking 3D bridge models with element-level defects. The application was built using Unreal Engine 4, a cross-platform game engine that allows the application to be deployed to a variety of operating environments including Windows, iOS, and Android. The application automatically tallies condition state and defect quantity information, freeing the inspector from that burden. The captured bridge
inspection data can be transmitted wirelessly to MDOT to be stored for bridge management purposes.

Additionally, using information from MDOT’s BMS database, MTRI developed a system for generating representative 3D bridge models for common concrete bridge construction styles. While these models may not be perfect, in concert with user tuning they are designed to be more than sufficiently useful for enabling inspectors to recognize the bridge structure and intuitively interact with it.

**How the Application Will Be Used by MDOT**

MDOT has the option of using the application to revise the current state of practice for bridge inspections. Initially, this use should consist of an implementation-focused trial period in which a few interested inspectors attempt to utilize the application in day-to-day operations. MDOT should use the trial period to identify challenges that will be faced in deploying the application statewide, and to formulate improvements for meeting those challenges. Potential problems could include but would not be limited to hardware issues (battery life), errors in the application itself that were not apparent during development, integration issues with the MDOT BMS, and deficiencies in user training. Discovering and correcting these issues during a trial phase would be critical for long-term success in revising MDOT practices, as a rocky deployment could burden an otherwise useful tool with a poor reputation. Conversely, a smooth deployment could improve adoption rate and support further development. The MTRI team would work with MDOT to address issues discovered throughout the implementation trial period.

During the trial phase, problems that actively disrupt the inspector’s workflow should be addressed immediately. Continuing the trial without correcting disruptive problems could mask other issues, rendering the ongoing trial usage ineffective. On the other hand, smaller obstacles such as unintuitive interactions and cosmetic flaws should be documented for later evaluation and correction as time and funding permit. Trial participants should be informed of this differentiation so they do not become fixated on perceived flaws that do not reduce the overall functionality of the application. This is especially critical since different inspectors may have different opinions as to what the “best” approach is. Continually revising noncritical functionality will squander the time and effort spent on the trial implementation, distracting from the identification of more serious issues that could be barriers to successful deployment.

At the conclusion of the trial period, MDOT should evaluate the feedback generated by the trial participants. Ideally, any key issues will already have been solved, but if they have not, this is the time to evaluate the readiness level of the application. If the key issues identified cannot be fixed quickly, the application may need further development and another trial implementation before it is ready for wider distribution and full
introduction into day-to-day usage. If the application completes the trial period without such setbacks, MDOT can evaluate the trial participants’ feedback and make a final decision about deployment.

Value Added to MDOT Operations

As a digital application, the 3D Bridge App eliminates the need for the paper forms currently used by bridge inspectors. This allows them to perform any number of inspections without access to a printer. Since inspection information is ultimately stored in the BMS database, the application also eliminates the manual data transcription process, which is costly in terms of time consumed and is an additional source of potential error in the transcribed data. Additionally, the application has great potential to streamline the inspection process, improving inspector efficiency and accuracy in the field. This is especially the case since the application allows inspectors to capture AASHTO element-data in an intuitive manner (by marking the bridge model with defects as the inspector observes them in real time). Further, the application automatically aggregates the information for reporting, freeing inspectors from that burden. The application also is able to display context-sensitive information concerning the inspection process, such as the condition-state guidance tables from the Michigan Bridge Element Inspection Manual. Integrating those tables into the application allows the inspectors to quickly verify their choices without flipping through the physical manual.

As a digital platform for bridge inspections, the application offers a wealth of new opportunities in the future. Integrating more-detailed bridge models, such as those used during the construction of the bridge, could better facilitate lifetime management of infrastructure. Additionally, the rendering capabilities of Unreal Engine 4 could be leveraged to display remote sensing data as overlays on the 3D models, aiding both inspections and management decision-making.

To enable MDOT to take advantage of the full value of these mobile app technologies and the investment made in the 3D Bridge App, the Michigan Tech team has recommended four tasks for a potential third phase of the project. These tasks would focus on implementing and deploying the application into day-to-day usage at MDOT. They are as follows:

1. Integrate the 3D Bridge App with MDOT’s database using the current version of BrM. Currently, the application will save inspections as XML files that are output to specific locations on the tablets. To integrate the application with MDOT’s database, all of the information within the XML would need to be uploaded to MDOT’s database.

2. Update key features identified by MDOT to make the application more user-friendly and the bridge inspector’s job even easier. MTRI has recorded all of the suggestions
made throughout Phase II of the project, and suggests MDOT look at some of these for potential incorporation (see Appendix 8.6).

3. Use the 3D Bridge App for a wider set of bridges. Current implementation focuses mostly on generic highway overpass bridges in Michigan. More detailed models could be created by rendering non-generic bridge elements and improving material and mesh fidelity (material is how the model is 'painted', the mesh refers to the geometry itself) to help mimic reality. MTRI would ensure that the application creates models for the majority of bridges that are accurate enough for use.

4. Perform alpha and beta tests to bring the 3D Bridge App to the point of deployment into day-to-day usage.

**Implementation Plan Checklist:**

<table>
<thead>
<tr>
<th>Results achieved through this research (check all that apply)</th>
<th>Actions needed to implement results (check all that apply)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Knowledge to assist MDOT</td>
<td>X Management decision</td>
</tr>
<tr>
<td>X Manual change</td>
<td>X Funding</td>
</tr>
<tr>
<td>Policy development or change</td>
<td>X Training</td>
</tr>
<tr>
<td>X Development of software/computer application</td>
<td>X Information technology deployment</td>
</tr>
<tr>
<td>X Development of new process</td>
<td>X Information-sharing</td>
</tr>
<tr>
<td>X Additional research needed</td>
<td>X Other (specify) implementation-focused trails and database integration.</td>
</tr>
<tr>
<td>Project produced no usable results</td>
<td></td>
</tr>
<tr>
<td>Other (describe)</td>
<td></td>
</tr>
</tbody>
</table>

**8.6 List of Possible Future Developments for the 3D Bridge App**
(The development times noted in parentheses are estimates; some tasks may take more or less time than expected.)

**Features to Integrate (Short Amount of Time)**

- Develop a “Home View” button that when clicked would reset the camera to a set location so that users know where they are. Would help if the user gets “lost” within the 3D environment.
- Develop common views in addition to just navigating using the pinch/slide method. Common views include looking down at deck, looking up at deck soffit, right and left elevations, and front and back face of substructure.
• Develop square and round columns through user tuning.
• Render an approach slab for every bridge. (Dimensions are not in the database, but MTRI could hard-code them, so inspectors could record approach slab information.)
• Show length and width of a defect instead of total area and aspect ratio.
• Develop a button that would give directions to the bridge through Google Maps.
• Show bridge name somewhere on the screen.
• Show time stamp in the corner.
• Develop an exit-without-saving button.
• Develop zoom capabilities in defect editor mode.

Features to Integrate (Medium Amount of Time)

• Develop an Orientation Viewer—Have a side button named “Viewer” to assist in orienting the inspector to the bridge. Once the user clicked this button, a list of every individual member would be displayed and organized by element, span, bay, etc. If a user clicked the individual member, the camera would be placed in a position for viewing that member. A rough example is shown below.
  o Beams
    ▪ Span1
    ▪ Span2
    ▪ Span3
      • Beam 3S 1W
      • Beam 3S 2W
      • Beam 3S 3W
      • Beam 3S 4W
      • Beam 3S 5W
    ▪ Span4
    ▪ Span5

• Develop a label schema that can be toggled on and off with a button to clearly label every individual member.
• Add the option to enter defects according to the strict unit reported in the MBE rather than surface area only. For instance, select all of an element when reported by “each”. For instance with bearings, users would not want to highlight one face of a bearing; the whole bearing should be bad. Or, when a beam end is bad, since beams are reported by linear feet, then it would highlight the entire surface area of the beam for the length of the beam that is bad. For most elements it is useful that there is the option (columns for instance), but many users would want their inspectors to match.
• Have a view-only mode, especially on desktop computers, where inspectors could view inspection models but could not edit any information.
• Develop a copy-and-paste functionality for defects.
• Limit defect dimensions to the dimensions of the element it is attached to.
• Limit the defect total quantity so that it cannot exceed the total quantity of the element. Currently, users can make the defect as big as they want.
• Require inspector to take a minimum number of photos before pushing the data (i.e., sending the data to the application’s back-end for storage).
• Implement different materials to simulate concrete, steel, etc.
• Develop capability to draw over pictures taken with camera.

**Features to Integrate (Long Amount of Time)**

• Have the application use GPS to allow for the inspector to be better oriented. (What happens if GPS gets disconnected? How reliable will GPS be under a bridge?)
• Pinch-to-rotate view as in Sim City game. Camera could be confined to a region, as opposed to a rail. (Camera rails are easy and can be made to help the user avoid getting lost when navigating around a bridge in the app.)
• Draw a defect in defect editor mode.
• For cracks, draw the crack by setting a series of points, which the application would then connect to draw a line.
• Implement customized elements—splayed spans, curved girders, beam shapes, box beams, T-beams, straight beam curved decks, etc.
• Integrate model into Google Maps to overlap with Google Maps’ version of the bridge based on the latitude and longitude coordinates, as some CAD models are able to do.
• Add voice-to-text for comments, or ability to write in comments with stylus; this is especially useful when inspectors are on the deck and have to watch out for traffic.
• Have the compass reflect the actual direction the inspector is facing.
• Render pin-and-hanger assemblies as well as diaphragms (need user input).
• Create decals for every individual defect to better reflect what each defect looks like instead of showing the defect as a rectangle or circle.
• Add spell-checker feature for comments.
• Toggle protective system or coating from whatever element the coating is on.
• Make the deck transparent to see defects on the top or bottom surface of the deck.
• Integrate CAD models into the app, and also integrate map metadata into the CAD model.
• For reviewing past inspections, highlight old defects that have not been reviewed by the inspector.
• Develop 3D models focused on design, not operations and inspection/maintenance (add in areas on operations and inspection/maintenance).
• Need top/bottom layers to show corrosion above bottom spalls.
• Use camera as recording device to take video, or use unmanned aerial vehicles to take video or pictures.
• Add skew to the bridge model.
• Include settlement units—deflection between elements.
• Add button to export all photos to a photo log and organize the folders correctly. This would be useful since normally, one inspector inspects the bridge while another inspector is taking photos.