**Software Requirements Specification**

**for the**

**UMBC Virtual Tour 2.0 System**

**Document # CMSC447-05-FA2018-G03-SRS**

Version 1.1

28 November 2018

**Prepared By:**

Noah Johnson

Ronan Kaye

Tyler Little

Ryan Martin

Kristin McLaughlin

Table of Contents

[Table of Contents 2](#_Toc531032813)

[Revision History 3](#_Toc531032814)

[**1.1 Identification 4**](#_Toc531032815)

[**1.2 System Overview 4**](#_Toc531032816)

[**1.2.1 Purpose 4**](#_Toc531032817)

[**1.2.2. Development History 4**](#_Toc531032818)

[**1.2.3 Deployment Locations 4**](#_Toc531032819)

[**1.2.4 System Functions 4**](#_Toc531032820)

[**1.3 Document Overview 5**](#_Toc531032821)

[2 References 6](#_Toc531032822)

[3 Requirements 6](#_Toc531032823)

[**3.1 Required states and modes 6**](#_Toc531032824)

[**3.2 CSCI Functional Requirements 7**](#_Toc531032825)

[**3.2.1 Virtual Tour Interface CSCI (VTI) 7**](#_Toc531032826)

[**3.2.2 Virtual Unity Engine CSCI (VUE) 9**](#_Toc531032827)

[**3.3 CSCI external interface requirements 12**](#_Toc531032828)

[**3.3.1 Interface identification and diagrams 12**](#_Toc531032829)

[**3.4 CSCI internal interface requirements 12**](#_Toc531032830)

[**3.5 CSCI internal data requirements 12**](#_Toc531032831)

[**3.6 Adaptation requirements 13**](#_Toc531032832)

[**3.7 Safety requirements 13**](#_Toc531032833)

[**3.8 Security and privacy requirements 13**](#_Toc531032834)

[**3.9 CSCI environment requirements 13**](#_Toc531032835)

[**3.10 Computer resource requirements 13**](#_Toc531032836)

[**3.10.1 Computer hardware requirements 13**](#_Toc531032837)

[**3.10.2 Computer hardware resource utilization requirements 13**](#_Toc531032838)

[**3.10.3 Computer software requirements 14**](#_Toc531032839)

[**3.10.4 Computer communications requirements 14**](#_Toc531032840)

[**3.11 Software quality factors 14**](#_Toc531032841)

[**3.12 Design and implementation constraints 14**](#_Toc531032842)

[**3.12.1 Standards Compliance 14**](#_Toc531032843)

[**3.13 Personnel-related requirements 15**](#_Toc531032844)

[**3.14 Training-related requirements 15**](#_Toc531032845)

[**3.15 Logistics-related requirements 15**](#_Toc531032846)

[**3.16 Other requirements 15**](#_Toc531032847)

[**3.17 Packaging requirements 16**](#_Toc531032848)

[**3.18 Precedence and criticality of requirements 16**](#_Toc531032849)

[4 Qualifications Provisions 16](#_Toc531032850)

[5 Requirements Traceability 17](#_Toc531032851)

[6 Notes 17](#_Toc531032852)

[**6.1 Background and Rationale 17**](#_Toc531032853)

[**6.2 Abbreviations, Terms, and Definitions 18**](#_Toc531032854)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description of Changes** | **Author(s)** |
| Version 1.0 | 10/26/2018 | * First draft for customer review | Noah Johnson  Ronan Kaye  Tyler Little  Ryan Martin  Kristin McLaughlin |
| Version 1.1 | 11/28/2018 | * Added authors to title page * Changed text alignment to justified * Moved system functions content from section 2 to section 1 * Added development history and deployment location information to section 1 * Moved background information and abbreviations, terms, and definitions from section 2 to section 6 * Revised system CSCIs relationship diagram * Revised description of CSCIs to reflect updated design * Added finite state diagram and description of system modes and states to section 3.1 * Completed major revisions to requirements in section 3.2 to reflect updated system design * Added content for section 3.10.3, Computer Software Requirements * Added mention of version control to section 3.11.1 * Converted qualifications provisions into matrix format and moved them from section 3 to section 4 * Added additional abbreviations, terms, and definitions | Noah Johnson |

**1 Scope**

This specification establishes the functional, performance, and development requirements for version 1.0 of a software application enabling virtual tours of the University of Maryland, Baltimore Country (UMBC) campus.

**1.1 Identification**

Title: UMBC Virtual Tour 2.0

Abbreviation: VT2

Version Number: 1.0

## 1.2 System Overview

### 1.2.1 Purpose

The purpose of the VT2 system is to improve the existing basic UMBC virtual campus tour applications by importing the UMBC campus map and building information into the Unity game engine, enabling users to freely explore a three-dimensional (3D) rendering of the campus. Additionally, the system offers other useful features, such the ability to highlight valid parking locations on campus based on user status. The intended users of the system are prospective students seeking to familiarize themselves with the campus environment and current students, faculty, and visitors trying to find their classes or event venues and seeking the best place to park.

### 1.2.2. Development History

Development of the system began in September 2018, with a prototype of version 1.0 of the system scheduled for completion in early December 2018. The project is sponsored by the UMBC Department of Computer Science and Electrical Engineering, and the development team consists of senior computer science majors at UMBC. If successful, the project will be acquired by UMBC and incorporated into university’s website in the future.

### 1.2.3 Deployment Locations

The only planned operating site for the software is the UMBC main campus located in Baltimore, Maryland. During the next phase of development, however, the software will be extended to include the UMBC campus at the Universities at Shady Grove, located in Rockville, MD.

### 1.2.4 System Functions

The VT2 system includes the following Computer Software Configuration Items (CSCIs):

* ***Virtual Tour Interface (VTI) CSCI***: The VTI CSCI provides a menu-based web interface for the VT2 system based on the WebGL framework. It manages the user’s interaction with the VUE CSCI. The VTI is accessed through the website www.umbcvirtualtour.com.
* ***Virtual Unity Engine (VUE) CSCI***: The VUE CSCI is a customized version of the Unity engine that includes accurate 3D renderings and textures of the UMBC campus buildings. The VUE CSCI consists of several Computer Software Components (CSCs). Notably, the *VUE\_CampusExplorer* CSC enables natural movement around the campus with motion and camera effects similar to first person (1P) and third person (3P) point of view video games, while the *VUE\_ParkingFinder* CSC provides the ability to identify parking lots where the user is allowed to park based on the user’s status (faculty member, commuter student, residential student, visitor, etc.).

The below diagram depicts the relationship between the VTI and VUE CSCIs:

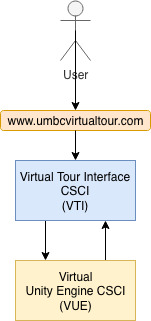


Figure 1 Relationship Between VT2 System CSCIs

**1.3 Document Overview**

This document is organized as follows: Section 1 identifies the scope of this document and provides information about the purpose, development history, and deployment locations of the VT2 system. Section 2 lists the documents referenced in this specification. Section 3 lists the functional requirements for the system, and section 4 lists the qualification methods for each requirement. Section 5 describes the traceability requirements for the system. Finally, section 6 provides background information on the system and a list of abbreviations, terms, and definitions needed to understand this document.

**2 References**

The following standards apply:

MIL-STD-498 Military Standard Software Development and Documentation

UMBC Style Guide https://styleguide.umbc.edu/

Unity User Manual https://docs.unity3d.com/Manual/index.html

WebGL Manual https://docs.unity3d.com/Manual/webgl.html

**3 Requirements**

**3.1 Required states and modes**

The VT2 system’s modes and states include non-operational mode, idle mode, operational mode, and the exception state, as depicted by the following diagram.

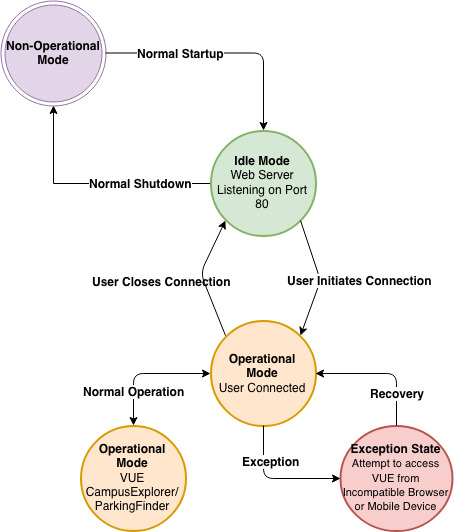


Figure 2 VT2 System Finite State Diagram

The general system flow is as follows:

1. The system begins in the non-operational state with the web server powered down.
2. After the web server is powered on, it assumes an idle state, listening on port 80 for users to connect through hypertext transfer protocol (HTTP).
3. Once a user connects to the website, the system enters the initial operational mode that allows access to help menus and other basic features.
4. From this mode, users on laptop or desktop computers with compatible browsers can access the advanced features of the site, including the features provided by the VUE\_CampusExplorer and VUE\_ParkingFinder CSCs. Users may move freely between the basic and advanced operational modes.
5. If a user attempts to access the advanced features from a mobile device or with an incompatible browser, however, the system enters the exception state. In this state, the system displays an appropriate error message and returns to the initial operational mode.
6. When the user closes the connection to the VT2 website, the system returns to the idle mode.
7. By powering down the webserver, the system can be returned to the non-operational mode.

**3.2 CSCI Functional Requirements**

This section describes the functional requirements for the VT2 system’s two CSCIs: the Virtual Tour Interface (VTI) CSCI and the Virtual Unity Engine CSCI (VUE) CSCI. The VUE CSCI is composed of multiple Computer Software Components (CSCs), of which the two most important are the VUE\_CampusExplorer CSC and the VUE\_ParkingFinder CSC. Requirements specific to these two CSCs will be described in detail in section 3.2.2.

**3.2.1 Virtual Tour Interface CSCI (VTI)**

3.2.1.1 The VTI shall be a graphical user interface (GUI) accessed via the website www.umbcvirtualtour.com.

3.2.1.2 The VTI shall use the WebGL JavaScript API to deploy the VUE.

3.2.1.3 The VTI main page (the website home page) shall include a welcome banner identifying it as the UMBC Virtual Tour 2.0 Website.

3.2.1.4 The VTI shall include a link to the UMBC home page (www.umbc.edu) and display one of any of the UMBC logos according to the UMBC style guidelines.

3.2.1.5 The VTI main page shall display a menu to the user with the following options: “About this Website”, “Help”, “Explore the Campus”, and “Find Parking”.

3.2.1.6 Clicking the “About this Website” option shall take the user to a page that provides a brief introduction to UMBC and identifies the purpose of the website as providing virtual tours of the university.

3.2.1.7 Clicking the “Help” option shall direct the user to a new page on the website that provides a tutorial on how to use the “Explore the Campus” and “Find Parking” features. The length of the tutorial shall be no fewer than 150 words and no more than 300 words for each of the two features.

3.2.1.8 Clicking the “Explore the Campus” option shall direct the user to a new page on the website that launches the VUE\_CampusExplorer CSC.

3.2.1.9 Clicking the “Find Parking” option shall display a menu prompting the user to identify as a student, faculty, or visitor.

3.2.1.10 After selecting “Find Parking”, if the user selects the student option, the VTI shall prompt the user to identify the type of student parking permit possessed. The options for the permit shall be displayed in two parts: a letter in white font within a colored circle, and the name of the type of permit. The options shall be as displayed in the Table 2 below. The VTI shall also allow the user to select the option, “I don’t have a permit.”

Table 1 Student Parking Permit Types

|  |  |
| --- | --- |
| **Letter and Color** | **Name** |
| ‘A’, red | Commuter Student |
| ‘B’, green | Walker Community Resident |
| ‘C’, yellow | Residential Student (Besides Walker) |
| ‘F’, orange | Freshman Resident Student |

3.2.1.11 If the user selects the faculty option, the VTI shall prompt the user to identify the type of faculty parking permit possessed. The options for the permit shall be displayed in two parts: a letter in white font within a colored circle, and the name of the type of permit. The options shall be as displayed in Table 3 below. The VTI shall also allow the user to select the option, “I don’t have a permit.”

Table 2 Student Parking Permit Types

|  |  |
| --- | --- |
| **Letter and Color** | **Name** |
| ‘D’, purple | Faculty/Staff |
| ‘E’, purple | Gated Faculty/Staff |

3.2.1.12 If the user selects the option “I don’t have a permit”, the VTI shall display a message telling the user that current faculty and students can request a permit through the UMBC parking services website and providing a hyperlink to the website. Additionally, the VTI shall display an option allowing the user to “Paid Parking Options.”

3.2.1.13 If the user selects a student or faculty permit type, the VTI shall prompt the user to select a destination building. The VTI will present a drop-down menu with the 25 buildings identified in Table 3 below.

3.2.1.14 Once the user selects a building, the VTI shall direct the user to a new page on the website that launches the VUE\_ParkingFinder CSC with parameters representing the user’s permit status and menu choices. The parameters shall enable the VUE\_ParkingFinder to highlight valid parking lots for the user, as well as the closest valid lot (the “Recommended Parking Lot”).

3.2.1.15 If the user selects the “Visitor” option, then the VPF shall present a menu with the options, “Special Event Parking” and “Paid Parking Options.”

3.2.1.16 If the user selects “Special Event Parking,” the VTI shall direct the user to a new page on the website that launches the VUE\_ParkingFinder CSC, with the stadium lot highlighted as described in VUE\_ParkingFinder CSC requirements section.

3.2.1.17 If the user selects “Paid Parking Options” in requirements 3.2.1.10 or 3.2.1.11, the VTI shall prompt the user to select a destination building. The VTI shall present a drop-down menu with the 25 buildings identified in requirement Table 3 below, as well as a “No specific building” option.

3.2.1.18 Once the user selects an option under requirement 3.2.1.17, the VTI shall direct the user to a new page on the website that launches the VUE\_ParkingFinder CSC with the parameters required to highlight all the paid parking lots on campus, as well as a closest “Recommended Parking Lot” if the user selected a specific building.

3.2.1.19 With the exception of the main page (the website home page), each VTI web page shall have a clickable button labeled “Exit” or “Home” that returns the user to the VTI main page.

3.2.1.20 The website shall store no data related to the user (such as cookies) other than a user agent string necessary for determining browser compatibility.

3.2.1.21 The website shall be compatible with the following web browsers: Firefox 4+, Google Chrome 9+, Opera 12+, Safari 5.1+, Internet Explorer 11+, and Microsoft Edge build 10240+.

3.2.1.22 If the user accesses the “Explore Campus” or “Find Parking” features with a browser other than one listed in requirement 3.2.1.13, the VTI shall display an error message stating that the user is using an incompatible browser and listing the acceptable browser types.

3.2.1.23 If the user attempts to access the “Explore Campus” or “Find Parking” features from a mobile device, the website shall display an error message stating that WebGL is not supported on mobile devices.

**3.2.2 Virtual Unity Engine CSCI (VUE)**

*General VUE CSCI Requirements*

3.2.2.1 The VUE shall use UMBC campus map data from Open Street Map (OSM) as the basis for its map.

3.2.2.2 The VUE shall enhance the basic OSM map by importing models and textures obtained from the UMBC Imaging Research Center (IRC) for the following 25 campus buildings:

Table 3 Names and Abbreviations for UMBC Campus Buildings Included in VT2 System

|  |  |
| --- | --- |
| **Name** | **Abbreviation** |
| Administration Building | Admin |
| Biological Sciences Building | Biology |
| Campus Police and Central Plant | Police |
| Chesapeake Hall | Chesapeake |
| The Commons | The Commons |
| Engineering Building | Engineering |
| Erickson Hall | Erickson |
| Event Center | Event Center |
| Fine Arts Building | Fine Arts |
| Harbor Hall | Harbor |
| Information Technology/Engineer Building | ITE |
| Mathematics & Psychology Building | Math & Psych |
| Meyerhoff Chemistry Building | Meyerhoff |
| Performing Arts and Humanities Building | PAHB |
| Patapsco Hall | Patapsco |
| Physics Building | Physics |
| Potomac Hall | Potomac |
| Public Policy Building | PUP |
| Retriever Activities Center (RAC) | RAC |
| Sherman Hall | Sherman |
| Sondheim Hall | Sondheim |
| Student Development & Success Center (SDSC) | SDSC |
| Susquehanna Hall | Susquehanna |
| True Grit’s | True Grit’s |
| University Center | UC |

3.2.2.3 When in overhead view, the map shall display the abbreviated building name from Table 1 above; mousing over the building shall display the full name of the building.

3.2.2.4 When in 1P or 3P view, the map shall display the full name of the building when the building is visible on the screen and the user is within a 100-foot radius of the building.

3.2.2.5 The VUE shall accurately render the relative heights of the campus buildings.

3.2.2.6 While in 1P or 3P view, the camera view shall remain behind the user at all times and shall not pass through the exterior of buildings.

3.2.2.7 In addition to the campus buildings, the VUE custom map shall also display grassy areas, roads, pavement, sidewalks, and paths that match the actual UMBC campus.

3.2.2.8 The VUE shall display the lot numbers of all parking lots, as well as an icon with a letter within a colored circle, as noted in requirements 3.2.1.10 and 3.2.1.11, that indicates which permit types allow parking in that lot.

*VUE\_CampusExplorer CSC Requirements*

3.2.2.9 The VUE\_CampusExplorer shall load within 15 seconds and shall display a progress bar indicating the time remaining until loading is complete.

3.2.2.10 Upon launch, the VUE\_CampusExplorer initial view shall be zoomed out such that the entire campus map is displayed in overhead view.

3.2.2.11 The VUE\_CampusExplorer shall provide a prompt telling the user to double-click a location on the campus to start exploring.

3.2.2.12 After the user double-clicks a location on the map, the system shall zoom to a 3P street-level view of that part of the map.

3.2.2.13 If the user clicks a non-walkable area of the campus (any area other than a sidewalk or other walkway, road, parking lot, or grass), the system shall zoom in to the closest walkable location; alternatively, the system shall display a message informing the user that an invalid location was chosen and prompt the user to click a walkable location. Similarly, the system shall only allow the user to move on walkable areas of the map. If the user tries to move into any other area, the user’s avatar shall continue to display walking motion, but no forward progress shall occur.

3.2.2.14 The user’s avatar shall take the form of the UMBC mascot, the Chesapeake Bay Retriever True Grit.

3.2.2.15 When in 3P view, the system shall provide a method of shifting to 1P view and the system shall provide a text box informing the user how to shift to this view.

3.2.2.16 In 3P view, the user shall be able to change the camera perspective by moving the mouse or trackpad up, down, left, and right. The camera perspective shall change in the direction opposite of mouse or trackpad movement. Specifically, moving the mouse or trackpad left shall rotate the player’s view to the right, while moving the mouse or trackpad right shall rotate the player’s view to the left. Moving the mouse or trackpad up shall rotate the player’s view down. For example, if the player was directly in front of an object, moving the mouse or trackpad down shall shift the camera perspective so that the player now appears to be looking up at the object. Similarly, moving the mouse or trackpad down shall rotate the player’s view up. Using the above example, moving the mouse or trackpad down shall shift the camera perspective so that the player now appears to be looking down at the object from above.

3.2.2.17 In 1P view, the user shall also be able to change the camera perspective by moving the mouse or trackpad up, down, left, and right. However, in contrast to 3P view, the camera perspective shall move in the same direction of mouse or trackpad movement. Specifically, moving the mouse or trackpad left shall rotate the player’s view to the left, while moving the mouse or trackpad right shall rotate the player’s view to the right. Similarly, moving the mouse or trackpad down shall rotate the player’s view down, and moving the mouse or trackpad up shall rotate the player’s view up.

3.2.2.18 The user shall be able to advance through the 3D rendering of the map by using the *w*, *a*, *s,* and *d* keyboard keys or the up, left, right, and down arrow keys. Pressing these movement keys shall advance the player in a certain direction without changing the direction the player is facing. Specifically, pressing the *w* or up arrow keys shall move the player forwards; pressing the *a or* left arrowkeys shall move the player left; pressing the *s* or down arrow keys shall move the player backwards; and pressing the *d* or right arrow keys shall move the player right.

3.2.2.19 In both 1P and 3P views, the system shall display an inset overhead view map showing the user’s current location.

*VUE\_ParkingFinder CSC Requirements*

3.2.2.20 Upon launch, the VUE\_ParkingFinder shall load within 15 seconds and shall display a progress bar indicating the time remaining until loading is complete.

3.2.2.21 Upon launch, the VUE\_ParkingFinder shall have already received the parameters from the VTI indicating which parking lots it should highlight.

3.2.2.22 If a student or faculty permit was selected, the VUE\_ParkingFinder shall highlight all the parking lots on the map where the user can park based on the user’s parking permit.

3.2.2.23 Additionally, after highlighting all the parking lots on the map where the user can park based on the user’s student or faculty permit, the VUE\_ParkingFinder shall identify the closest parking lot to the destination building where the user is allowed to park. This closest lot shall be highlighted in a different color from the other lots and shall be labeled “Recommended Parking Lot.”

3.2.2.23 If the user selected “Special Event Parking,” the VUE\_ParkingFinder shall highlight the stadium lot.

3.2.2.24 If the user selected “Paid Parking Options”, the VUE\_ParkingFinder shall then highlight all the paid parking lots on campus

3.2.2.25 If the user selected a building with the “Paid Parking Options”, the VUE\_ParkingFinder shall highlight the closest paid parking option to that building in a different color and that lot shall be labeled “Recommended Parking Lot.”

**3.3 CSCI external interface requirements**

**3.3.1 Interface identification and diagrams**

All interaction with the user shall occur through the VTI CSCI. As noted in section 3.2.1, the VTI shall be a web-based GUI.

**3.4 CSCI internal interface requirements**

As noted in section 3.2.1, the VTI shall use the WebGL JavaScript API.

**3.5 CSCI internal data requirements**

All internal data requirements shall be left to the design.

**3.6 Adaptation requirements**

No adaptation requirements have been identified.

**3.7 Safety requirements**

No specific safety requirements have been identified.

**3.8 Security and privacy requirements**

There are no specific security or privacy requirements, as the system will not record any logon information or other personal data related to the user, nor does the system house any confidential data.

**3.9 CSCI environment requirements**

3.9.1 The computer acting as the server and host for the website http://umbcvirtualtour.com shall meet the requirements specified on the “System Requirements for Unity 2018.2” webpage located at https://unity3d.com/unity/system-requirements. Specifically, the computer shall use one of the following operating systems: Windows 7 SP1+, macOS 10.11+, Ubuntu 12.04+, or SteamOS+.

3.9.2 The website shall be accessed through one of the following web browsers: Firefox 4+, Google Chrome 9+, Opera 12+, Safari 5.1+, Internet Explorer 11+, or Microsoft Edge build 10240+.

**3.10 Computer resource requirements**

**3.10.1 Computer hardware requirements**

3.10.1.1 The computer acting as the server and host for the website http://umbcvirtualtour.com shall have a graphics card with DX10 (shader model 4.0) capabilities.

3.10.1.2 The computer acting as the server and host for the website http://umbcvirtualtour.com shall have a CPU with SSE2 instruction set support.

3.10.1.3 The computer acting as the server and host for the website http://umbcvirtualtour.com shall have a minimum 2.4 GHz processor with 8 GB of memory.

3.10.4 The system shall require the user to use a mouse when accessing the system to enable camera movement.

**3.10.2 Computer hardware resource utilization requirements**

No requirements for computer hardware resource utilization have been identified.

**3.10.3 Computer software requirements**

3.10.3.1 The computer acting as the server and host for the website http://umbcvirtualtour.com shall use one of the following operating systems: Windows 7 SP1+, macOS 10.11+, Ubuntu 12.04+, or SteamOS+.

3.10.3.2 Users shall access the system website http://umbcvirtualtour.com using one of the following web browsers: Firefox 4+, Google Chrome 9+, Opera 12+, Safari 5.1+, Internet Explorer 11+, or Microsoft Edge build 10240+.

**3.10.4 Computer communications requirements**

The system shall communicate over the internet through standard web protocols, namely TCP and UDP.

**3.11 Software quality factors**

3.11.1 Maintainability – All code written for the VT2 application shall be properly documented to ensure maintainability. Specifically, all code shall include comments in accordance with the standards set forth in the HTML5 style guide; all files shall include headers with the file name, purpose, author, and last modified date; and all functions shall include pre and post conditions. Additionally, the software shall be version controlled using an appropriate online code repository.

3.11.2 Reliability – The system shall not crash except due to operating system faults or other external errors.

3.11.3 Availability – The system shall always be available when accessed except in the case of external faults such as hardware failures.

3.11.4 Portability – The VT2 application shall be portable to any hardware server that meets the requirements outline in section 3.10. Likewise, the VT2 application shall be accessible by any user whose browser meets the requirements in section 3.9.

**3.12 Design and implementation constraints**

**3.12.1 Standards Compliance**

3.12.1.1 The VT2 system shall comply with the standards described in section 1.3. Additionally, The VT2 system shall comply with the UMBC Style Guide (https://styleguide.umbc.edu/) with regard to the use of the UMBC name and logo.

3.12.1.2 The system relies on UMBC’s IRC to provide the object files necessary for creating three-dimensional renderings of the campus buildings. As of late November 2018, the IRC did not have renderings and textures for some of the buildings. As a result, version 1.0 of the web application will only contain a subset of the 43 buildings that comprise the UMBC main campus.

3.12.1.3 Additionally, the WebGL framework that allows the Unity engine to be run within a web browser is not currently supported on mobile devices. Therefore, the user must access the system through a desktop computer or laptop.

3.12.1.4 Also, a compatible browser as described in the Unity WebGL manual (https://docs.unity3d.com/Manual/webgl-browsercompatibility.html) must be used. Most modern browsers (Firefox, Chrome, Safari, Microsoft Edge) are supported. Future versions of the system will provide support for mobile computing using a different framework.

3.12.1.5 Finally, version 1.0 of the software will not provide the user with directions to or from parking lots or buildings. It is intended that the system will provide this capability in a future release.

**3.13 Personnel-related requirements**

Users of this system are assumed to possess basic familiarity with internet browsers and websites, but no other special knowledge or skills are required. All user interaction with the system will take place through browser-based menus, and the system will prompt the user to take action with clear and simple instructions when necessary. The VTI CSCI shall also provide a help menu.

**3.14 Training-related requirements**

No special training is required to use the system. The VT2 website shall provide prompts and a help section to assist the user in navigating the site.

**3.15 Logistics-related requirements**

While the customer shall have responsibility for maintaining the server on which the software resides after testing and acceptance, the development team shall provide technical support for problems encountered with the software via the Jira ticketing system, email, and live chat. The technical support team shall respond to all requests within one business day.

**3.16 Other requirements**

Use of the Unity engine within the VT2 system shall comply with the requirements of Unity Technologies’ licensing policy. Specifically, the VT2 system shall initially use a free unity license for students and in-class instruction as described on the Unity website at https://store.unity.com/education. Once testing and acceptance is complete and UMBC incorporates the system into its official website, the VT2 system shall convert to an appropriate paid license.

**3.17 Packaging requirements**

The system shall be packaged as a single compressed file containing binaries, executables, and source code for the web application and all files necessary to install the customized Unity engine. Additionally, the file shall contain a README with information about how to install and run the software on a web server. Initially, the development team shall have responsibility for maintaining the server on which the software runs. After completion of acceptance and verification, the development team shall transfer responsibility for the server to the customer.

**3.18 Precedence and criticality of requirements**

The development team and customer have determined that the most critical requirements are implementing the VTI CSCI and the VUE\_CampusExplorer CSC. Therefore, development of these components shall take precedence over development of the VUE\_ParkingFinder CSC and any other features proposed in the future.

**4 Qualifications Provisions**

The below table provides the qualification provisions for each requirement from section 3.

Table 4 VT2 System Qualification Provisions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement**  **Number** | **Demonstration** | **Testing** | **Analysis** | **Inspection** | **Special** |
| 3.2.1.1 | ● |  |  |  |  |
| 3.2.1.2 |  |  |  | ● |  |
| 3.2.1.3 | ● |  |  |  |  |
| 3.2.1.4 | ● |  |  |  |  |
| 3.2.1.5 | ● |  |  |  |  |
| 3.2.1.6 | ● |  |  |  |  |
| 3.2.1.7 | ● |  |  |  |  |
| 3.2.1.8 | ● |  |  |  |  |
| 3.2.1.9 | ● |  |  |  |  |
| 3.2.1.10 | ● |  |  |  |  |
| 3.2.1.11 | ● |  |  |  |  |
| 3.2.1.12 | ● |  |  |  |  |
| 3.2.1.13 | ● |  |  |  |  |
| 3.2.1.14 | ● |  |  |  |  |
| 3.2.1.15 | ● |  |  |  |  |
| 3.2.1.16 | ● |  |  |  |  |
| 3.2.1.17 | ● |  |  |  |  |
| 3.2.1.18 | ● |  |  |  |  |
| 3.2.1.19 | ● |  |  |  |  |
| 3.2.1.20 |  |  |  | ● |  |
| 3.2.1.21 |  |  |  |  |  |
| 3.2.1.22 | ● |  |  |  |  |
| 3.2.1.23 | ● |  |  |  |  |
| 3.2.2.1 |  |  |  | ● |  |
| 3.2.2.2 |  |  |  | ● |  |
| 3.2.2.3 | ● |  |  |  |  |
| 3.2.2.4 | ● |  |  |  |  |
| 3.2.2.5 |  |  |  | ● |  |
| 3.2.2.6 | ● |  |  |  |  |
| 3.2.2.7 | ● |  |  | ● |  |
| 3.2.2.8 | ● |  |  |  |  |
| 3.2.2.9 | ● |  |  |  |  |
| 3.2.2.10 | ● |  |  |  |  |
| 3.2.2.11 | ● |  |  |  |  |
| 3.2.2.12 | ● |  |  |  |  |
| 3.2.2.13 | ● |  |  |  |  |
| 3.2.2.14 | ● |  |  |  |  |
| 3.2.2.15 | ● |  |  |  |  |
| 3.2.2.16 | ● |  |  |  |  |
| 3.2.2.17 | ● |  |  |  |  |
| 3.2.2.18 | ● |  |  |  |  |
| 3.2.2.19 | ● |  |  |  |  |
| 3.2.2.20 | ● |  |  |  |  |
| 3.2.2.21 |  |  |  | ● |  |
| 3.2.2.22 | ● |  |  |  |  |
| 3.2.2.23 | ● |  |  |  |  |
| 3.2.2.24 | ● |  |  |  |  |
| 3.2.2.25 | ● |  |  |  |  |

**5 Requirements Traceability**

Each functional requirement listed in section 3 is labeled with a unique requirement number. All future documentation for the VT2 system shall be traceable back to a requirement in section 3 using this requirement number.

**6 Notes**

## 6.1 Background and Rationale

A virtual campus tour is an important component of a university’s strategy for recruiting students. In an environment of intense competition for students of all types—domestic and international, in-state and out-of-state, and undergraduate and graduate—a strong virtual tour application can convince a prospective student to apply or visit the campus in person. Additionally, virtual campus tours can help current students and visitors navigate their way to their classes or special events.

UMBC currently has several websites that nominally offer virtual tours of the campus. The Undergraduate Admissions UMBC Virtual Tour (located at undergraduate.umbc.edu/visit/virtual-tour.php) provides 9 panoramic views of the campus, though it claims to offer 25 views. A virtual tour site for the graduate school (gradschool.umbc.edu/discover/vtour/) simply provides a link to the same site that hosts the panoramic campus views noted above. Additionally, a UMBC undergraduate student created a basic virtual tour mobile application for Android devices in 2014 titled, “Introducing UMBC Tours - A Virtual Campus Tour Experience for Android” (www.youtube.com/watch?v=zRI61jkUDT4). However, this implementation had extremely limited functionality and did not represent a significant improvement on the applications offered on the UMBC website.

The purpose of the UMBC VT2 software described in this specification is to dramatically improve the currently available UMBC virtual tour applications by importing the UMBC campus map and building information into the Unity game engine. It will allow users to select any location on a three-dimensional map of the campus and allow them to explore it freely. It will provide browser-based access to this system through a web application. Moreover, the new system will provide several other useful features, including the ability to highlight valid parking locations based on user status. The system will primarily benefit prospective students seeking to familiarize themselves with the campus environment and current students, faculty, and visitors trying to find their classes or event venues and seeking the best place to park.

## 6.2 Abbreviations, Terms, and Definitions

1P First-person point of view

3D Three dimensional

3P Third-person point of view

API Application Programming Interface

CSC Computer Software Component

CSCI Computer Software Configuration Item

GUI Graphical User Interface

HWCI Hardware Configuration Item

HTML Hyper Text Markup Language

HTTP Hyper Text Transfer Protocol

IL2CPP Intermediate Language to C++

IRC The Imaging Research Center, a digital media laboratory at UMBC and the source of the building models and textures used in the VT2 system.

MTL File A Material Library (.mtl) file contains one or more material definitions, each of which includes the color, texture, and reflection map of individual materials. These are applied to the surfaces and vertices of objects and are stored in ASCII format.

OBJ File An object (.obj) file is a standard 3D image format that can be exported and opened by various 3D image editing programs. It contains a three-dimensional object including 3D coordinates, texture maps, polygonal faces, and other object information.

OSM Open Street Map

SIMD Single Instruction, Multiple Data

SRS Software Requirements Specification

SSE2 Streaming SIMD Extensions 2

TCP Transmission Control Protocol

TGA File A Truevision Graphics Adapter (.tga) file is a raster graphics file format that can store raw or compressed images.

UMBC University of Maryland, Baltimore County

Unity The Unity cross-platform game engine

VCE Virtual Campus Explorer CSCI

VPF Virtual Parking Finder CSCI

VTI Virtual Tour Interface CSCI

VT2 UMBC Virtual Tour 2.0

VUE Virtual Customized Unity Engine CSCI

WebGL The Web Graphics Library, a cross platform JavaScript API for rendering 2D and 3D graphics in a web browser

XML Extensible Markup Language

UDP User Datagram Protocol