**Software Test Description (STD)**

**for the**

**UMBC Virtual Tour 2.0 System**

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

Version 1.0

10 December 2018

**Prepared By:**

Noah Johnson

Ronan Kaye

Tyler Little

Ryan Martin

Kristin McLaughlin

**Table of Contents**

[**Revision History**](#_14ju4wjojm2f) **4**

[**1 Scope**](#_1ke9deyixfzg) **5**

[1.1 Identification](#_3znysh7) 5

[1.2 System Overview](#_2et92p0) 5

[1.2.1 Purpose](#_tyjcwt) 5

[1.2.2. Development History](#_3dy6vkm) 5

[1.2.3 Deployment Locations](#_1t3h5sf) 5

[1.2.4 System Functions](#_4d34og8) 5

[1.3 Document Overview](#_2s8eyo1) 6

[**2 References**](#_frhlcu2agi7a) **7**

[**3 Test Preparations**](#_3rdcrjn) **7**

[3.1. Hardware Preparation](#_26in1rg) 7

[3.1.1 Server](#_823m3shns296) 7

[3.1.2 Client](#_df9ws68igztq) 7

[3.2 Software Preparation](#_3ett5hbh40sw) 8

[3.2.1 Server](#_ios4o64q97x3) 8

[3.2.2 Client](#_fptng714pt5v) 8

[**4 Test Descriptions**](#_2xprevecbn8t) **8**

[4.1.1 The user accesses http://umbcvirtualtour.com/ through a specified web browser](#_qwjnfcw94gaw) 8

[4.1.1.1 Display of VTI Main Page](#_cplr468n6wuc) 8

[4.1.1.2 Prerequisites for 4.1.1.1](#_p45r3txuzrv6) 8

[4.1.1.3 Test inputs for 4.1.1.1](#_fr830hvt9f8) 9

[4.1.1.4 Test Results for 4.1.1.1](#_nzytdajcodju) 9

[4.1.1.5 Criteria for evaluation 4.1.1.1](#_ftsvwfjyudmi) 9

[4.1.1.6 Test procedure for 4.1.1.1](#_5vmgns3kqpik) 9

[4.1.1.7 Assumptions and Constraints for 4.1.1.1](#_4y0ma65wl5jj) 9

[4.1.2 Contents of the VTI Homepage](#_7kp4cfic5ond) 9

[4.1.2.1 Functionality of VTI homepage](#_nbyk148kxabu) 9

[4.1.2.2 Prerequisites for 4.1.2.1](#_ky1ef2enbvtz) 9

[4.1.2.3 Test inputs for 4.1.2.1](#_ksxcw0ohpci) 10

[4.1.2.4 Test Results for 4.1.2.1](#_r2dz8a4lggg9) 10

[4.1.2.5 Criteria for evaluation 4.1.2.1](#_39ge41eyak1k) 10

[4.1.2.6 Test procedure for 4.1.2.1](#_ek8pzeifogb) 10

[4.1.2.7 Assumptions and Constraints for 4.1.2.1](#_qlczqk66h02n) 10

[4.1.3 The web server provides a stable hosting platform for end users](#_xgf5fnxr9dqv) 10

[4.1.3.1 Backend functionality](#_65pjhsr1gtey) 10

[4.1.3.2 Prerequisites for 4.1.3.1](#_undayjpwjze7) 11

[4.1.3.3 Test inputs for 4.1.3.1](#_yliuv11eod58) 11

[4.1.3.4 Test Results for 4.1.3.1](#_8g0zh0yr06a2) 11

[4.1.3.5 Criteria for evaluation 4.1.3.1](#_625s3ewxq5r9) 11

[4.1.3.6 Test procedure for 4.1.3.1](#_uxhvi6zf8hn5) 11

[4.1.3.7 Assumptions and Constraints for 4.1.3.1](#_rvrl2zy0j4ld) 11

[4.1.4 VUE provides realistic campus view](#_4nxb9leibyqr) 11

[4.1.4.1 VUE View from User](#_eobfpx07od2) 11

[4.1.4.2 Prerequisites for 4.1.4.1](#_hnchx8ixi21w) 12

[4.1.4.3 Test inputs for 4.1.4.1](#_mvbrczxo6672) 12

[4.1.4.4 Test Results for 4.1.4.1](#_bwanau2zhqxn) 12

[4.1.4.5 Criteria for evaluation for 4.1.4.1](#_vb2pmac0rm19) 12

[4.1.4.6 Test procedure for 4.1.4.1](#_uz7twrfjxhse) 12

[4.1.4.7 Assumptions and Constraints for 4.1.4.1](#_8avc1uf8kdud) 13

[4.1.5 VUE uses specified maps, models, textures, and movement](#_20v6h936pfmg) 13

[4.1.5.1 Backend VUE Functionality](#_hammmtkkfr42) 13

[4.1.5.2 Prerequisites for 4.1.5.1](#_r947p3e6xe9d) 13

[4.1.5.3 Test inputs for 4.1.5.1](#_db79hvrqz2xv) 14

[4.1.5.4 Test Results for 4.1.5.1](#_8sn08a9l6g7l) 14

[4.1.5.5 Criteria for evaluation 4.1.5.1](#_lv9w3kk57zm6) 14

[4.1.5.6 Test procedure for 4.1.5.1](#_lptvv7hs6nib) 14

[4.1.5.7 Assumptions and Constraints for 4.1.5.1](#_7u040c67tvsv) 14

[4.1.6 VT2 website meets minimum hardware and software requirements](#_z3k8xtan9s4v) 14

[4.1.6.1 Computer Functionality as Server](#_pjbcahm51uo0) 14

[4.1.6.2 Prerequisites for 4.1.6.1](#_u8s0jc7sw0h5) 15

[4.1.6.3 Test inputs for 4.1.6.1](#_dlx6ngoah06b) 15

[4.1.6.4 Test Results for 4.1.6.1](#_aunafro2brsu) 15

[4.1.6.5 Criteria for evaluation 4.1.6.1](#_xq8w4g4e9j1e) 15

[4.1.6.6 Test procedure for 4.1.6.1](#_vjz3hk532g37) 15

[4.1.6.7 Assumptions and Constraints for 4.1.6.1](#_y440k6ohkc0t) 15

[**5 Requirements Traceability**](#_cyvaxbpb6cvl) **16**

[**6 Notes**](#_2jxsxqh) **18**

[6.1 Background and Rationale](#_z337ya) 18

[6.2 Abbreviations, Terms, and Definitions](#_3j2qqm3) 18

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description of Changes** | **Author(s)** |
| Version 1.0 | 12/10/2018 | * UMBC VT2 system prototype build version | Noah Johnson  Ronan Kaye  Tyler Little  Ryan Martin  Kristin McLaughlin |

# 1 Scope

## 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.

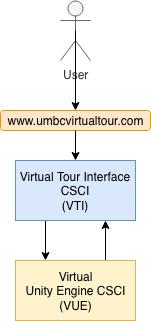
### 1.2.3 Deployment Locations

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



*Figure 1 Relationship Between VT2 System CSCIs*

## 1.3 Document Overview

This Software Test Description (STD) document describes the test preparations, test cases, and test procedures to be used to perform qualification testing of the Computer Software Configuration Item (CSCI) capabilities specified in the UMBV VT2 system Software Requirements Specification (SRS) and the System/Subsystem Design Description (SSDD) documents.

# 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 Test Preparations

Sections 3.1 and 3.2 define the hardware and software prerequisites necessary to facilitate review of the UMBC Virtual Tour 2.0 System. Hardware and software for both the server and client must be prepared to fully test the system’s functionality.

## 3.1. Hardware Preparation

### 3.1.1 Server

The server hosting the system’s website (www.umbcvirtualtour.com) should have the following hardware installed:

* Graphics card with DX10 (shader model 4.0) capabilities
* CPU: SSE2 instruction set support
* Minimum 2.4 GHz processor with 8GB of memory.

The server should be powered on and connected to the internet prior to testing.

### 3.1.2 Client

The client should be a laptop or desktop with a minimum 2.4 GHz processor and 8GB of memory. It must have either a mouse or a trackpad to access the “Explore Campus” feature of the VT2 system.

A mobile device, such as a smartphone or tablet, should also be prepared for the sole purpose of testing whether attempting to access the system from a mobile device triggers the appropriate warnings.

## 

## 

## 

## 3.2 Software Preparation

### 3.2.1 Server

The server should have the following software installed:

* Windows 7 SP1+, macOS 10.11+, Ubuntu 12.04+, or SteamOS+ operating system
* Web server software, such as Apache HTTP Server

Prior to testing, the custom Unity engine--the Virtual Unity Engine (VUE) CSCI--must be installed on the server and built for WebGL. Finally, the server should be set to listen for TCP/IP connections on port 80.

### 

### 3.2.2 Client

The client must have all of the following web browsers installed: Firefox 4+, Google Chrome 9+, Opera 12+, Safari 5.1+, Internet Explorer 11+, and Microsoft Edge build 10240+. Additionally, the client must have at least one additional browser installed that is not in the above list.

# 4 Test Descriptions

The following sections regarding test descriptions will detail all unique conditions which may arise for the client during the execution of the UMBC Virtual Tour 2.0 System. Execution of the system shall begin when the client accesses <http://umbcvirtualtour.com/> through a web browser given the specified hardware and software prerequisites.

## 4.1.1 The user accesses <http://umbcvirtualtour.com/> through a specified web browser

### 4.1.1.1 Display of VTI Main Page

* Upon successful entry, the VTI displays the homepage with a welcome banner representing UMBC Virtual Tour 2.0 system.
* The VTI includes links to UMBC home page and displays the UMBC logo.
* The VTI main page displays menu options “About this Website”, “Help”, and “Explore the Campus”.
* Each VTI web page has “Exit” or “Home” button that returns user to VTI homepage.

### 4.1.1.2 Prerequisites for 4.1.1.1

The client end system must meet the hardware and software requirements detailed in section 3.1 and 3.2.

### 4.1.1.3 Test inputs for 4.1.1.1

The client end system will be tested using all modern browsers listed in 3.2.2.

### 4.1.1.4 Test Results for 4.1.1.1

The website loads successfully on the client system, using each of the listed modern web browsers.

### 4.1.1.5 Criteria for evaluation 4.1.1.1

A proper display of the VTI Main Page is loaded within the clients web browser. These criteria are listed in section 4.1.1.1.

### 4.1.1.6 Test procedure for 4.1.1.1

This test case will be evaluated through course of demonstration.

1. Access the website “<http://umbcvirtualtour.com>” through a specified web browser from 3.2.2.
2. View that the page displays a welcome banner, links to the UMBC home page and UMBC logo.
3. View that the page displays 3 menu options: “About this Website”, “Help”, and “Explore”.
4. View that the webpage has an “Exit” or “Home” button.

### 4.1.1.7 Assumptions and Constraints for 4.1.1.1

## 4.1.2 Contents of the VTI Homepage

### 4.1.2.1 Functionality of VTI homepage

* Clicking “About this Website” provides an introduction to UMBC and explains the website’s purpose.
* Clicking “How To” provides tutorial on using “Virtual Tour” and “Free Roam” feature in 150-300 words.
* Clicking either “Virtual Tour” or “Free Roam” features launches the respective VUE CSCI.

### 4.1.2.2 Prerequisites for 4.1.2.1

The client end system must meet the hardware and software requirements detailed in section 3.1 and 3.2.

### 

### 4.1.2.3 Test inputs for 4.1.2.1

The client end system will be tested using all modern browsers listed in 3.2.2.

### 

### 4.1.2.4 Test Results for 4.1.2.1

The VTI will display the correct output that is expected based on criteria from 4.1.2.1

### 

### 4.1.2.5 Criteria for evaluation 4.1.2.1

The criteria are listed in 4.1.2.1 and the webpage will match that criteria.

### 

### 4.1.2.6 Test procedure for 4.1.2.1

This test case will be evaluated through course of demonstration.

1. From the VTI home page, click on “About this Website”.
2. View that there is an explanation and purpose for the website.
3. Return to the VTI home page by clicking “Home” or “Exit”.
4. Click on “How To”.
5. View that there is a tutorial displayed.
6. Return to the VTI home page by clicking “Home” or “Exit”.
7. Click on “Virtual Tour”.
8. View that the Virtual Tour VUE is launched.
9. Return to the VTI home page by clicking “Home” or “Exit”.
10. Click on “Free Roam”.
11. View that the Free Roam VUE is launched.

### 

### 4.1.2.7 Assumptions and Constraints for 4.1.2.1

## 4.1.3 The web server provides a stable hosting platform for end users

### 4.1.3.1 Backend functionality

The VTI is a GUI accessed via <http://umbcvirtualtour.com>, which allows end users to access the VUE CSCI using a modern web browser specified in section 3.2.2. The VUE CSCIs use WebGL Javascript API to produce a finalized build of the 3D tour which is developed using Unity. The website stores no user data other than a user agent string necessary for determining browser compatibility. If the user accesses the “Virtual Tour” or “Free Roam” feature with an incompatible browser, the VTI shall displays error message stating that the user is using an incompatible browser and listing the acceptable browser types. If user accesses the “Explore Campus” feature from a mobile device, VTI shall display an error message warning that the feature is not supported on mobile devices.

The web server which is to be implemented must satisfy the requirements stated above in order provide a stable hosting platform for end users.

### 4.1.3.2 Prerequisites for 4.1.3.1

The client and server systems must meet the hardware and software requirements detailed in section 3.1 and 3.2.

### 

### 4.1.3.3 Test inputs for 4.1.3.1

The client end system will be tested using all modern browsers listed in 3.2.2.

### 

### 4.1.3.4 Test Results for 4.1.3.1

The backend functionality works with the specified modern browsers listed in 3.2.2 and when tested with browsers that are not listed in 3.2.2 the error message is displayed.

### 

### 4.1.3.5 Criteria for evaluation 4.1.3.1

The test meets the criteria listed in 4.1.3.1.

### 

### 4.1.3.6 Test procedure for 4.1.3.1

This test case will be evaluated through course of demonstration.

1. Access the website “<http://umbcvirtualtour.com>”.
2. View that the VTI is loaded.
3. Access the website from a browser not listed in 3.2.2.
4. View that an error message is displayed notifying the user that the browser is not compatible.
5. Access the website from a mobile device.
6. View that an error message is displayed notifying the user that the browser is not compatible.

### 

### 4.1.3.7 Assumptions and Constraints for 4.1.3.1

## 4.1.4 VUE provides realistic campus view

### 4.1.4.1 VUE View from User

Addresses:

* In overhead view, VUE map displays abbreviated building name by default and full name when moused over
* In 1P and 3P views, VUE map displays full name of building when building is visible on the screen and user is within a 100 foot radius of the building
* In 1P and 3P views, camera remains behind user at all times does not pass through exterior of buildings
* VUE map displays grassy areas, roads, pavement, sidewalks, and paths matching actual UMBC campus
* VUE displays parking lot numbers and parking permit icons
* Upon launch, VUE initial view is zoomed out such that the entire campus map is displayed in overhead view
* VUE prompts user to double-click a location on the campus to start exploring
* After user double-clicks a location on VUE map, system zooms to a 3P street-level view of that part of the map
* If user clicks a non-walkable area of the campus, VUE either zooms in to closest walkable location or displays message informing user that an invalid location was chosen and prompts user to click a walkable location; system only allows user to move on walkable areas of the map. If user tries to move into any other area, no forward progress occurs
* In both 1P and 3P views, VUE displays inset overhead view map showing user’s current location

### 4.1.4.2 Prerequisites for 4.1.4.1

The client end system must meet the hardware and software requirements detailed in section 3.1 and 3.2.

### 

### 4.1.4.3 Test inputs for 4.1.4.1

The client end system will be tested using all modern browsers listed in 3.2.2.

### 

### 4.1.4.4 Test Results for 4.1.4.1

The VUE renders the view that is expected including details such as buildings, grass, parking lots and sidewalks. The view is only different from 1P to 3P in the placement of the camera but the details still look the same, but from a different angle.

### 

### 4.1.4.5 Criteria for evaluation for 4.1.4.1

The test meets the criteria listed in 4.1.4.1

### 

### 4.1.4.6 Test procedure for 4.1.4.1

This test case will be evaluated through course of demonstration.

1. Go the website “<http://umbcvirtualtour.com>” on a supported browser listed in 3.2.2.
2. Launch the VUE by clicking on “Free Roam”.
3. View that the buildings list their abbreviated name.
4. Mouse over the buildings and view that the full name is displayed.
5. View that there are grassy areas, roads, pavement, sidewalks, and paths displayed.
6. View that the parking lots are displayed with their lot numbers.
7. View that the entire map is displayed.
8. View that there is a prompt to double-click on a location on the map.
9. Double click on any location on the map.
10. View that the screen zooms to a 3P street-level view.
11. Click on an area that is non-walkable.
12. View that the VUE either zooms to a close walkable area or displays a prompt that an invalid location was chosen.
13. If a prompt was displayed, chose a walkable area.
14. Attempt to move the avatar with the arrow keys to a non-walkable area.
15. View that the avatar will not make any forward progress.
16. View that there is an overhead map displaying the user’s current location.

### 

### 4.1.4.7 Assumptions and Constraints for 4.1.4.1

## 4.1.5 VUE uses specified maps, models, textures, and movement

### 4.1.5.1 Backend VUE Functionality

Addresses:

* VUE uses UMBC campus map data from Open Street Map (OSM) as the basis for its map
* VUE enhances basic OSM map with models and textures from UMBC IRC for 25 campus buildings listed in SRS Table 3
* VUE provides method of shifting between 1P and 3P views
* In VUE 3P view, user changes camera perspective by moving the mouse or trackpad up, down, left, and right; perspective changes in direction opposite of mouse or trackpad movement
* In VUE 1P view, user changes camera perspective by moving the mouse or trackpad up, down, left, and right; perspective changes in same direction of mouse or trackpad movement
* User advances through VUE map by using the *w*, *a*, *s,* and *d* keyboard keys or the up, left, right, and down arrow keys; direction user faces is not changed

### 4.1.5.2 Prerequisites for 4.1.5.1

The client end system must meet the hardware and software requirements detailed in section 3.1 and 3.2.

### 

### 4.1.5.3 Test inputs for 4.1.5.1

The client end system will be tested using all modern browsers listed in 3.2.2.

### 

### 4.1.5.4 Test Results for 4.1.5.1

The user is able to advance throughout the simulation with the correct view as well as able to switch between 1P and 3P views.

### 

### 4.1.5.5 Criteria for evaluation 4.1.5.1

The test meets the criteria listed in 4.1.5.1.

### 

### 4.1.5.6 Test procedure for 4.1.5.1

This test case will be evaluated through course of inspection and demonstration.

1. Inspect the build files to view that OSM is used for the map.
2. Access the website “<http://umbcvirtualtour.com>”.
3. Launch the VUE by clicking on “Free Roam”.
4. View that there is an option to switch between 1P and 3P.
5. Choose 3P.
6. Move the mouse around and view that the screen view changes opposite of the mouse, i.e, if the mouse moves right, the view moves left.
7. Choose 1P.
8. Move the mouse around and view that the screen view changes in the same direction of the mouse, i.e, if the mouse moves right, the view moves right.
9. Move the avatar by using the *w*, *a*, *s,* and *d* keyboard keys or the up, left, right, and down arrow keys.
10. View that the avatar goes in the expected direction.

### 

### 4.1.5.7 Assumptions and Constraints for 4.1.5.1

## 4.1.6 VT2 website meets minimum hardware and software requirements

### 4.1.6.1 Computer Functionality as Server

Addresses:

* Computer acting as the server and host for the VT2 website meets requirements specified on the “System Requirements for Unity 2018.2” webpage; specifically, computer uses Windows 7 SP1+, macOS 10.11+, Ubuntu 12.04+, or SteamOS+
* Computer acting as the server and host for the VT2 website has graphics card with DX10 (shader model 4.0) capabilities
* Computer acting as the server and host for the VT2 website has CPU with SSE2 instruction set support
* Computer acting as the server and host for the VT2 website has minimum 2.4 GHz processor with 8 GB of memory

### 4.1.6.2 Prerequisites for 4.1.6.1

The client end system must meet the hardware and software requirements detailed in section 3.1 and 3.2.

### 

### 4.1.6.3 Test inputs for 4.1.6.1

The client end system will be tested using all modern browsers listed in 3.2.2.

### 

### 4.1.6.4 Test Results for 4.1.6.1

The computer has the correct functionality listed in 4.1.6.1.

### 

### 4.1.6.5 Criteria for evaluation 4.1.6.1

The test meets the criteria listed in 4.1.6.1.

### 

### 4.1.6.6 Test procedure for 4.1.6.1

This test case will be evaluated through course of inspection.

1. Inspect the computer used to as server to find that it meets the criteria in 4.1.6.1.

### 

### 4.1.6.7 Assumptions and Constraints for 4.1.6.1

# 5 Requirements Traceability

Table 1: VT2 System Requirements Traceability Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID No.** | **SRS Section** | **Requirement** | **SSDD Section** | **Test Case** |
| 1 | 3.2.1.1 | VTI is a GUI accessed via www.umbcvirtualtour.com | 4.1.1.1, 4.1.2.2 | 4.1.3 |
| 2 | 3.2.1.2 | Use WebGL JavaScript API for VUE | 4.1.2.2 | 4.1.3 |
| 3 | 3.2.1.3 | Website home page has welcome banner for VT2 system | 4.1.2.2 | 4.1.1 |
| 4 | 3.2.1.4 | VTI includes link to UMBC home page, displays UMBC logo | 4.1.2.2 | 4.1.1 |
| 5 | 3.2.1.5 | VTI main page displays menu with “About this Website”, “Help”, “Explore the Campus” options | 4.1.2.2 | 4.1.1 |
| 6 | 3.2.1.6 | Clicking “About this Website” provides intro to UMBC, explains website purpose | 4.1.2.2 | 4.1.2 |
| 7 | 3.2.1.7 | Clicking “Help” provides tutorial on using “Explore the Campus” feature in 150-300 words | 4.1.2.2  4.1.2.2.1 | 4.1.2 |
| 8 | 3.2.1.8 | Clicking “Explore the Campus” launches the VUE CSCI | 4.1.2.2.34.1.2.1.1 | 4.1.2 |
| 9 | 3.2.1.19 | Each VTI web page has “Exit” or “Home” button that returns user to VTI main page | 4.1.2.2 | 4.1.1 |
| 10 | 3.2.1.20 | Website stores no user data other than a user agent string necessary for determining browser compatibility | 4.1.2.2 | 4.1.3 |
| 11 | 3.2.1.21 | Website is compatible compatible with Firefox 4+, Google Chrome 9+, Opera 12+, Safari 5.1+, Internet Explorer 11+, and Microsoft Edge build 10240+ | 4.1.2.2.34.1.2.2.4 | 4.1.3 |
| 12 | 3.2.1.22 | If user accesses “Explore Campus” feature with a browser not listed in requirement 3.2.1.21, VTI displays error message stating that the user is using an incompatible browser and listing the acceptable browser types | 4.1.2.2.34.1.2.2.4 | 4.1.3 |
| 13 | 3.2.1.23 | If user accesses the “Explore Campus” feature from a mobile device, VTI shall display an error message warning that the feature is not supported on mobile devices | 4.1.2.2.34.1.2.2.4 | 4.1.3 |
| 14 | 3.2.2.1 | VUE uses UMBC campus map data from Open Street Map (OSM) as the basis for its map | 4.1.2.1.3 | 4.1.5 |
| 15 | 3.2.2.2 | VUE enhances basic OSM map with models and textures from UMBC IRC for 25 campus buildings listed in SRS Table 3 | 4.1.2.1.4 | 4.1.5 |
| 16 | 3.2.2.3 | In overhead view, VUE map displays abbreviated building name by default and full name when moused over | 4.1.2.1.14.1.2.1.2 | 4.1.4 |
| 17 | 3.2.2.4 | In 1P and 3P views, VUE map displays full name of building when building is visible on the screen and user is within a 100 foot radius of the building | 4.1.2.1.1 | 4.1.4 |
| 18 | 3.2.2.5 | VUE accurately renders relative heights of campus buildings | 4.1.2.1.6 |  |
| 19 | 3.2.2.6 | In 1P and 3P views, camera remains behind user at all times does not pass through exterior of buildings | 4.1.2.1.5 | 4.1.4 |
| 20 | 3.2.2.7 | VUE map displays grassy areas, roads, pavement, sidewalks, and paths matching actual UMBC campus | 4.1.2.1.7 | 4.1.4 |
| 21 | 3.2.2.8 | VUE displays parking lot numbers and parking permit icons | 4.1.2.1.14.1.2.1.2 | 4.1.4 |
| 22 | 3.2.2.9 | VUE loads within 15 seconds and displays progress bar indicating the time remaining until loading is complete | 4.1.1.1.  4.1.2.2 |  |
| 23 | 3.2.2.10 | Upon launch, VUE initial view is zoomed out such that the entire campus map is displayed in overhead view | 4.1.2.1.1 | 4.1.4 |
| 24 | 3.2.2.11 | VUE prompts user to double-click a location on the campus to start exploring | 4.1.2.1.1 | 4.1.4 |
| 25 | 3.2.2.12 | After user double-clicks a location on VUE map, system zooms to a 3P street-level view of that part of the map | 4.1.2.1.1 | 4.1.4 |
| 26 | 3.2.2.13 | If user clicks a non-walkable area of the campus, VUE either zooms in to closest walkable location or displays message informing user that an invalid location was chosen and prompts user to click a walkable location; system only allows user to move on walkable areas of the map. If user tries to move into any other area, no forward progress occurs | 4.1.2.1.1 | 4.1.4 |
| 27 | 3.2.2.15 | VUE provides method of shifting between 1P and 3P views | 4.1.2.1.5 | 4.1.5 |
| 28 | 3.2.2.16 | In VUE 3P view, user changes camera perspective by moving the mouse or trackpad up, down, left, and right; perspective changes in direction opposite of mouse or trackpad movement | 4.1.2.1.5 | 4.1.5 |
| 29 | 3.2.2.17 | In VUE 1P view, user changes camera perspective by moving the mouse or trackpad up, down, left, and right; perspective changes in same direction of mouse or trackpad movement | 4.1.2.1.5 | 4.1.5 |
| 30 | 3.2.2.18 | User advances through VUE map by using the *w*, *a*, *s,* and *d* keyboard keys or the up, left, right, and down arrow ke*ys*; direction user faces is not changed | 4.1.2.1.5 | 4.1.5 |
| 31 | 3.2.2.19 | In both 1P and 3P views, VUE displays inset overhead view map showing user’s current location | 4.1.2.1.1 | 4.1.4 |
| 32 | 3.9.1 | Computer acting as the server and host for the VT2 website meets requirements specified on the “System Requirements for Unity 2018.2” webpage; specifically, computer uses Windows 7 SP1+, macOS 10.11+, Ubuntu 12.04+, or SteamOS+ | 4.1.1.1 | 4.1.6 |
| 33 | 3.10.1.1 | Computer acting as the server and host for the VT2 website has graphics card with DX10 (shader model 4.0) capabilities | 4.1.1.1 | 4.1.6 |
| 34 | 3.10.1.2 | Computer acting as the server and host for the VT2 website has CPU with SSE2 instruction set support | 4.1.1.1 | 4.1.6 |
| 35 | 3.10.1.3 | Computer acting as the server and host for the VT2 website has minimum 2.4 GHz processor with 8 GB of memory | 4.1.1.1 | 4.1.6 |

# 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