**Software Requirements Specification**

**for the**

**UMBC Virtual Tour 2.0 System**

**Document # CMSC447-05-FA2018-G03-SRS-01A**

Revision A

26 October 2018

Table of Contents

[**0.1 26 October 2018: CMSC447-05-FA2018-G03-SRS-01A** 3](#_Toc528037613)

[1 Scope 3](#_Toc528037614)

[**1.1 Identification** 4](#_Toc528037615)

[**1.2 Definitions, Acronyms and Abbreviations** 4](#_Toc528037616)

[**1.3 References** 4](#_Toc528037617)

[**1.4 Document Overview** 4](#_Toc528037618)

[2 Overall Description 5](#_Toc528037619)

[**2.1 System Purpose and Perspective** 5](#_Toc528037620)

[**2.1 System Functions** 5](#_Toc528037621)

[**2.2 User Characteristics** 6](#_Toc528037622)

[**2.3 Constraints** 7](#_Toc528037623)

[**2.4 Assumptions and Dependencies** 7](#_Toc528037624)

[3 Requirements 7](#_Toc528037625)

[**3.1 Required states and modes** 8](#_Toc528037626)

[**3.2 CSCI Functional Requirements** 8](#_Toc528037627)

[**3.2.1 Virtual Tour Interface CSCI (VTI)** 8](#_Toc528037628)

[**3.2.2 Virtual Campus Explorer CSCI (VCE)** 8](#_Toc528037629)

[**3.2.4 Virtual Parking Finder CSCI (VPF)** 8](#_Toc528037630)

[**3.2.5 Virtual Unity Engine CSCI (VUE)** 8](#_Toc528037631)

[**3.3 CSCI external interface requirements** 8](#_Toc528037632)

[**3.3.1 Interface identification and diagrams** 8](#_Toc528037633)

[**3.3.x (Project unique identifier of interface)** 8](#_Toc528037634)

[**3.4 CSCI internal interface requirements** 8](#_Toc528037635)

[**3.5 CSCI internal data requirements** 8](#_Toc528037636)

[**3.6 Adaptation requirements** 8](#_Toc528037637)

[**3.7 Safety requirements** 8](#_Toc528037638)

[**3.8 Security and privacy requirements** 8](#_Toc528037639)

[**3.9 CSCI environment requirements** 8](#_Toc528037640)

[**3.10 Computer resource requirements** 8](#_Toc528037641)

[**3.10.1 Computer hardware requirements** 8](#_Toc528037642)

[**3.10.2 Computer hardware resource utilization requirements** 8](#_Toc528037643)

[**3.10.3 Computer software requirements** 8](#_Toc528037644)

[**3.10.4 Computer communications requirements** 8](#_Toc528037645)

[**3.11 Software quality factors** 9](#_Toc528037646)

[**3.12 Design and implementation constraints** 9](#_Toc528037647)

[**3.13 Personnel-related requirements** 9](#_Toc528037648)

[**3.14 Training-related requirements** 9](#_Toc528037649)

[**3.15 Logistics-related requirements** 9](#_Toc528037650)

[**3.16 Other requirements** 9](#_Toc528037651)

[**3.17 Packaging requirements** 9](#_Toc528037652)

[**3.18 Precedence and criticality of requirements** 9](#_Toc528037653)

[4 Qualification provisions 9](#_Toc528037654)

[5 Requirements traceability 9](#_Toc528037655)

[6 Notes 9](#_Toc528037656)

[A Appendixes 9](#_Toc528037657)

**0 Revision History**

**0.1 26 October 2018: CMSC447-05-FA2018-G03-SRS-01A**

Release A contains the basic specification for the UMBC Virtual Tour 2.0 System.

**1 Scope**

This specification establishes the fuctional, performance, and development requirements for version 1.0 of a software application enabling virtual tours of the UMBC campus.

**1.1 Identification**

Title: UMBC Virtual Tour 2.0

Abbreviation: VT2

Version Number: 1.0

**1.2 Definitions, Acronyms and Abbreviations**

CSCI Computer Software Configuration Item

GIS Geographic Information Systems

SRS Software Requirements Specification

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

**1.3 References**

The following standards apply:

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications

MIL-STD-498 Military Standard Software Development and Documentation

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

Unity User Manual

(2018.2) <https://docs.unity3d.com/Manual/index.html>

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

**1.4 Document Overview**

This document is organized as follows: Section 1 identifies the scope of this document and lists the definitions, abbreviations, acronyms, and references used therein. Section 2 provides an overview of the system and a brief description of its architecture. Section 3 lists all of the functional requirements for the system, including the four primary CSCIs. Section 4 provides a matrix with the qualification methods for the CSCIs from section 3. Section 5 provides traceability between the system requirements and the CSCIs , and Section 6 provides important notes about the project.

**2 Overall Description**

**2.1 System Purpose and Perspective**

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

**2.1 System Functions**

The system has three primary functions that correspond to the CSCIs specified in section 3:

1. **Virtual Campus Explorer (VCE):** This CSCI shall allow the user to freely explore the virtual UMBC campus map from a chosen starting point.
2. **Virtual Parking Finder (VPF):** This CSCI shall identify parking lots where the user is allowed to park based on the user’s status (faculty member, commuter student, resdiential student, visitor, etc.).
3. **Virtual Tour Interface (VTI):** This CSCI shall provide a browser-based interface for the VT2 system based on the WebGL framework. The interface shall provide the main menu for the system, allowing the user to select either the VCE function or the VPF function.
4. **Virtual Unity Engine (VUE):** This CSCI shall provide a customized version of the Unity Engine that includes accurate 3D renderings and textures of the UMBC campus buildings. It shall allow natural movement around the campus’ exterior spaces with motion and camera effects generally expected of first and third-person point of view video games.

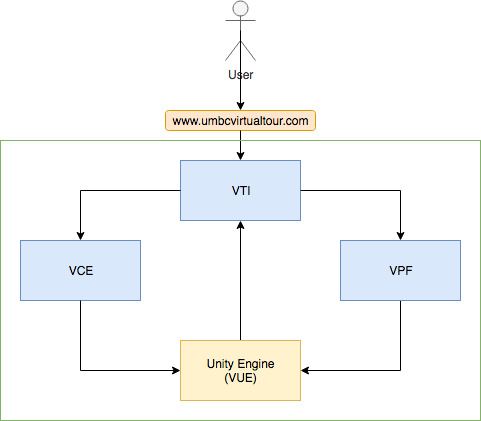


Figure 1 System Architecture

**2.2 User Characteristics**

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.

**2.3 Constraints**

The system relies on UMBC’s GIS office to provide the object files necessary for creating three-dimensional renderings of the campus buildings. As of late October 2018, the GIS department 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.

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. 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.

Finally, version 1 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.

**2.4 Assumptions and Dependencies**

It is assumed that the CSCIs identifed in this version of the SRS are the basic CSCIs necessary to meet customer requirements. Once the software engineering team has successfully implemented these CSCIs, the team may proceed with implementation of “reach goals” such as direction-finding algorithms and mobile support upon agreement with the customer.

Additionally, the engineering team’s ability to host the application at www.umbcvirtualtour.com is dependent on continued access to low-cost domain registration and

No other special assumptions or dependencies have been identified.

**3 Requirements**

**3.1 Required states and modes**

**3.2 CSCI Functional Requirements**

This section describes the functional requirements for each of the four CSCIs: the Virtual Tour Interface CSCI (VTI), the Virtual Campus Explorer CSCI (VCE), the Virtual Parking Finder CSCI (VPF), and the Virtual Unity Engine CSCI (VUE).

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

**3.2.2 Virtual Campus Explorer CSCI (VCE)**

**3.2.4 Virtual Parking Finder CSCI (VPF)**

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

**3.3 CSCI external interface requirements**

**3.3.1 Interface identification and diagrams**

**3.3.x (Project unique identifier of interface)**

**3.4 CSCI internal interface requirements**

**3.5 CSCI internal data requirements**

**3.6 Adaptation requirements**

**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.

**3.9 CSCI environment requirements**

**3.10 Computer resource requirements**

**3.10.1 Computer hardware requirements**

**3.10.2 Computer hardware resource utilization requirements**

**3.10.3 Computer software requirements**

**3.10.4 Computer communications requirements**

**3.11 Software quality factors**

**3.12 Design and implementation constraints**

**3.13 Personnel-related requirements**

**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**

**3.16 Other requirements**

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.

**3.17 Packaging requirements**

The system shall be packaged along with…

**3.18 Precedence and criticality of requirements**

**4 Qualification provisions**

**5 Requirements traceability**

**6 Notes**

**A Appendixes**