System Design Document

for

Virtual Video Modeling on the Social Skills of Adults with Autism

Version 1.0 approved

For Sarah K. Howorth

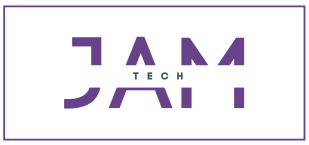
University of Maine

Nov 15, 2023

Prepared by JamTech:

Tristan Cilley, Allison Lupien, Nick Sarno,

Jacob Michaud, Maha Fazli



Virtual Video Modeling on the Social Skills of Adults with Autism

System Design Document

Table of Contents

1 . Introduction

This is a two-semester (**Fall 2023 - Spring 2024**) computer science capstone project to complete the requirements of a capstone experience at the University of Maine. This is a project for Dr. Sarah Howorth on virtual video modeling of the social skills of adults with autism.

Dr. Sarah Howorth is the director of the PEERS® Lab at UMaine and our primary client for this capstone project. The PEERS curriculum was written by Dr. Elizabeth Laugeson. The curriculum has already been converted into a PEERS mobile application on iOS. The mobile app contains information, video role-play examples, and practice questions to help users learn social skills. Dr. Howorth is interested in creating the next evolution of the PEERS app. She believes that users can learn social skills more effectively through a VR interface as it provides a more private, immersive, and engaging experience. This capstone project aims to create a proof-of-concept VR experience that can be used to solicit funding for further research.

1.1 Purpose of This Document

The purpose of this document is to describe our project's system design and architecture. This document is intended to both guide new users or laymen as well as serve as an official design document for the project. It will cover topics including system architecture, data-flows, file structures, and databases. The primary audience of this document includes our client, Dr. Sarah Howorth, and ourselves, JamTech. The secondary audience includes the professor of our capstone course, Dr. Laura Gurney, and anyone interested in learning more about the project.

1.2 References

This section includes a list of documents and other media related to this project. All of our documentation is maintained on our GitHub repository in the Deliverables directory. We used Miro to design our diagrams; there is a link below to both.

- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2023). GitHub repository, https://github.com/VoloVita/SeniorCapstone/tree/main/Deliverables
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2023). Systems Requirement Specification (SRS), version 1.0
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2023). User Interface Design Document (UIDD). **(available: Nov 29, 2023)**
- JamTech. (2023). *Miro,* https://miro.com/app/board/uXjVNShE4HY=/?share_link_id=215448921420

2 . System Architecture

This section of the system design document will cover the system's architecture, which refers to how the system is planned to be built. The system's architecture will also refer to the logical architecture of the program which is the structure of the system and decomposition description. ***See 1.2 references for link to all diagrams.***

2.1 Architectural Design

The architectural design of the system is the logical architecture of a system, which can be summarized as how the system is planned to be built. The diagrams below will represent the logical architecture of the system followed by a description for each.

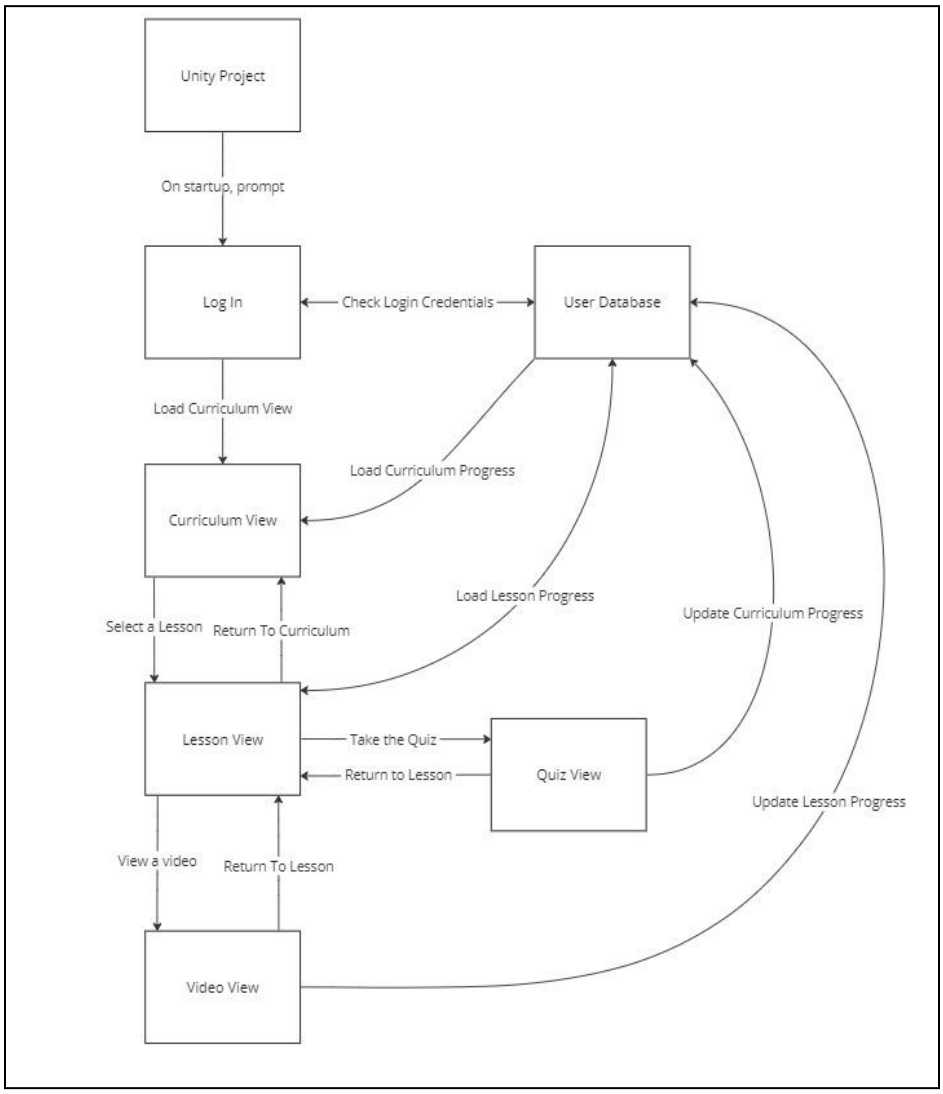


FIG 1.0 Software Architecture/Flow Diagram*: This graph showcases how the software parts interact with each other with the individual elements represented by boxes and arrows indicating interaction actions connecting everything.*

Our system flow diagram represents the relationship between different aspects of the software and their interactions with each other. The arrows are labeled with the corresponding functionality that occurs between each object. The direction of the arrow indicates the flow of information from one object to another. At most, the objects' points of reference are compared with the data within the user database in order to display the correct corresponding information to the user.

A basic data flow example using our diagram would look like this: on startup, the user is prompted to login to the system. From there, their login information is compared to that in our system's user database and if correct, the curriculum view is displayed to the user. The user can then decide to select a lesson which will bring them into the lesson view which displays a selection of videos to watch. The user can view a video which after completion will update the lesson progress of that user and return them to the lesson view. Upon completion of all of the videos in a given lesson, the user can enter the quiz view and upon completion of the quiz the curriculum progress will be updated and returned to the curriculum view.

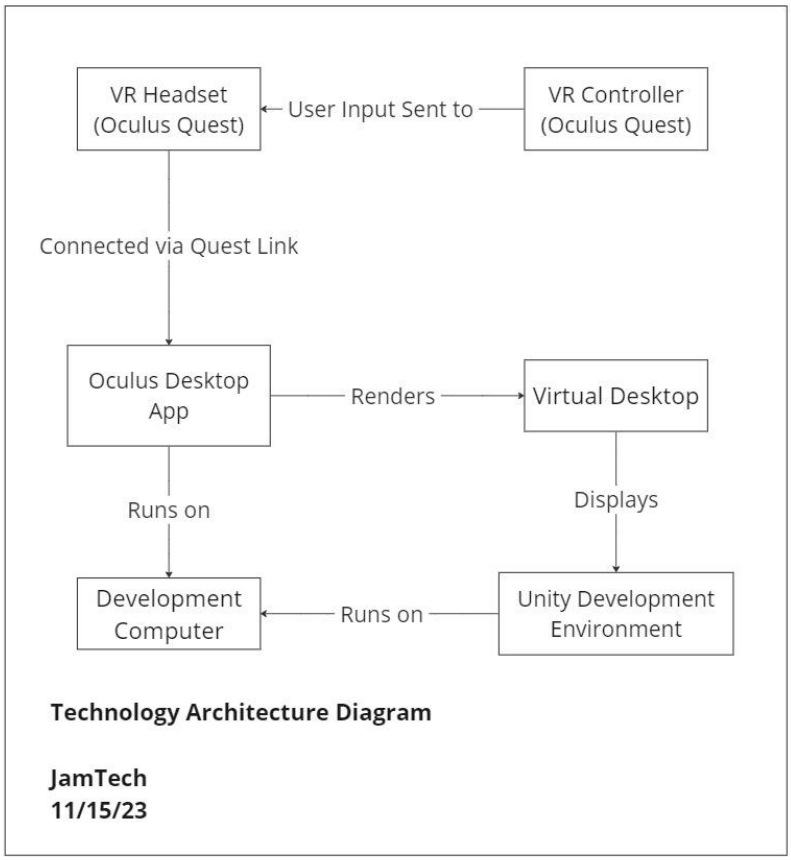


FIG 2.0 Technology Architecture Diagram*: This diagram shows the Oculus Quest connecting to our development computer via Quest Link. Once connected, we can view the Unity Development Environment in the VR headset and interact with Unity via the controllers.*

Fig 2.0 depicts the interactions between our hardware and software components. The VR equipment connects to the development computer via Quest Link which requires a USB-C 3.0 cable. Once connected, a virtual desktop environment can be accessed through the Oculus Desktop Application. The user can open a Unity Development Environment in the virtual desktop, and run the project. When played, the project is rendered in the VR headset and Unity receives input via the controllers.

2.2 Decomposition Description

This section will describe the decomposition of the system presented in section 2.1, the diagram provided will illustrate the system's design.

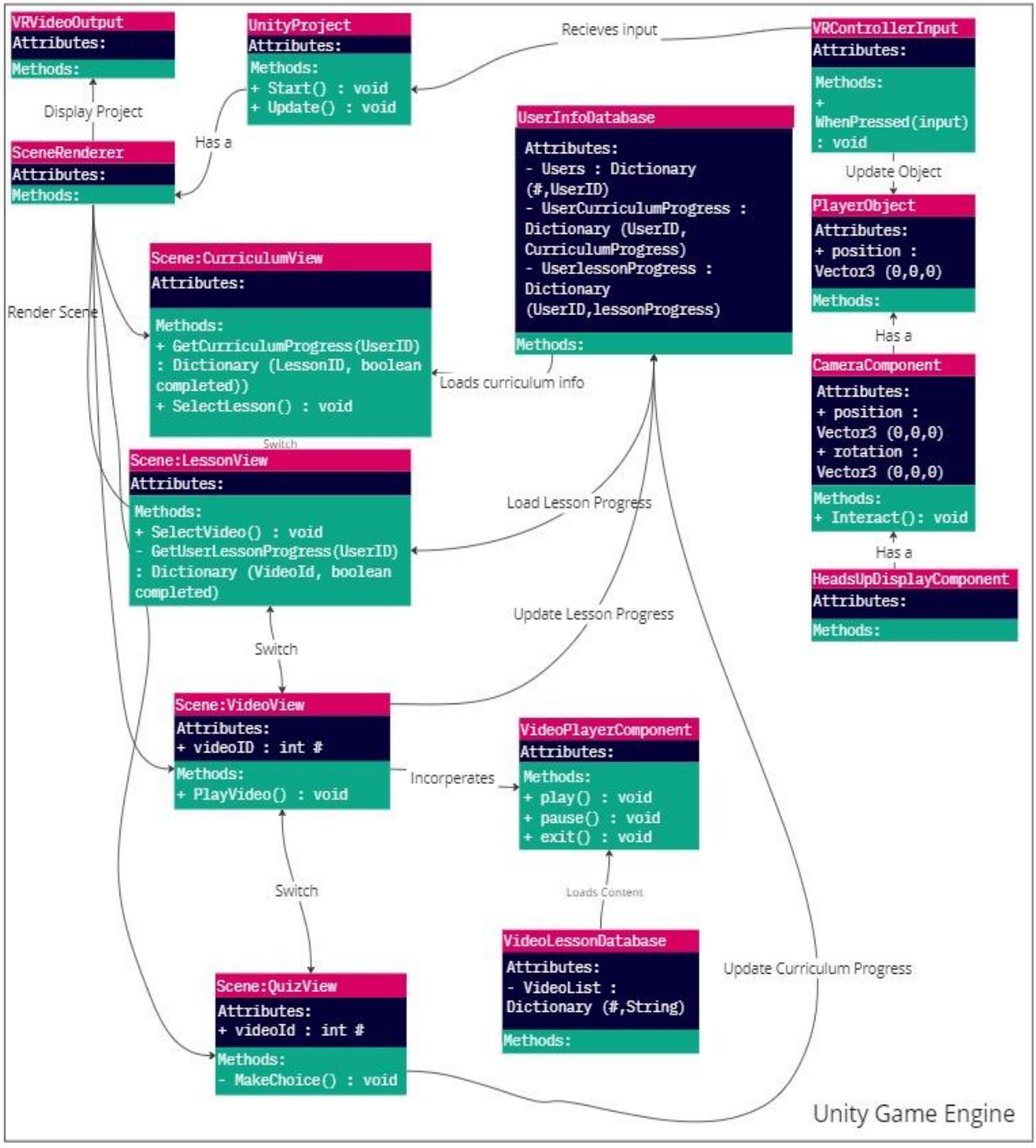


FIG 3.0 Unity Game Engine*: This diagram depicts the system's architectural design, represented by core components, their attributes, and their methods. The boxes contain this information and are connected by arrows with action words to describe how each component interacts with the system and the other components.*

The current system architecture can be represented by a context diagram and a UML diagram. The core component of this system is the Unity Project, which is the main software component represented by the UML diagram. The Unity Project is hosted on a desktop computer that has VR-ready GPU processing capabilities. The UML diagram contains multiple components that display the software's logic, including but not limited to: user interactions, video playback, logic within the application, and video/scene rendering. There are also multiple databases that are used to store information about users and their curriculum, lesson, and video progress.

Within the UML diagram are arrows indicating the interactions between different components of both the software and hardware. For example, the VRControllerInput interfaces with the unity project, and has a direct correlation to the PlayerObject, which will update the world and current view of the software to be displayed via the VRVideoOutput. In a simpler context, the desktop computer serves as the host of the application, the VR headset and controller are used to interact with the application, and the software and databases process the information needed to transfer the data between multiple systems.

3 . Persistent Data Design

This section outlines how the system will store data for this project; it is broken into sections 3.1 and 3.2.

Section 3.1 describes the databases used in this project and how they store data using entity-relationship diagrams. After each diagram there is a table that elaborates each element with its data-type, size and a brief description. Any personal identifiable information will be stored as a hash rather than plaintext. We intend to use NoSQL for the User Info and Lesson Content databases. We plan to use MongoDB to facilitate this design choice.

Section 3.2 includes a diagram of the GitHub repository for this project and a basic file structure for the Unity project.

3.1 Database Descriptions

The following diagram represents the configuration of our project's databases and their respective data. This project will include two separate databases. Figure 4.0 is a diagram of the User Info Database, which will store the user's information like username, name, password, curriculum progress, ect. Figure 5.0 is a diagram of the Lesson Content Database that includes lesson content like videos and quizzes.

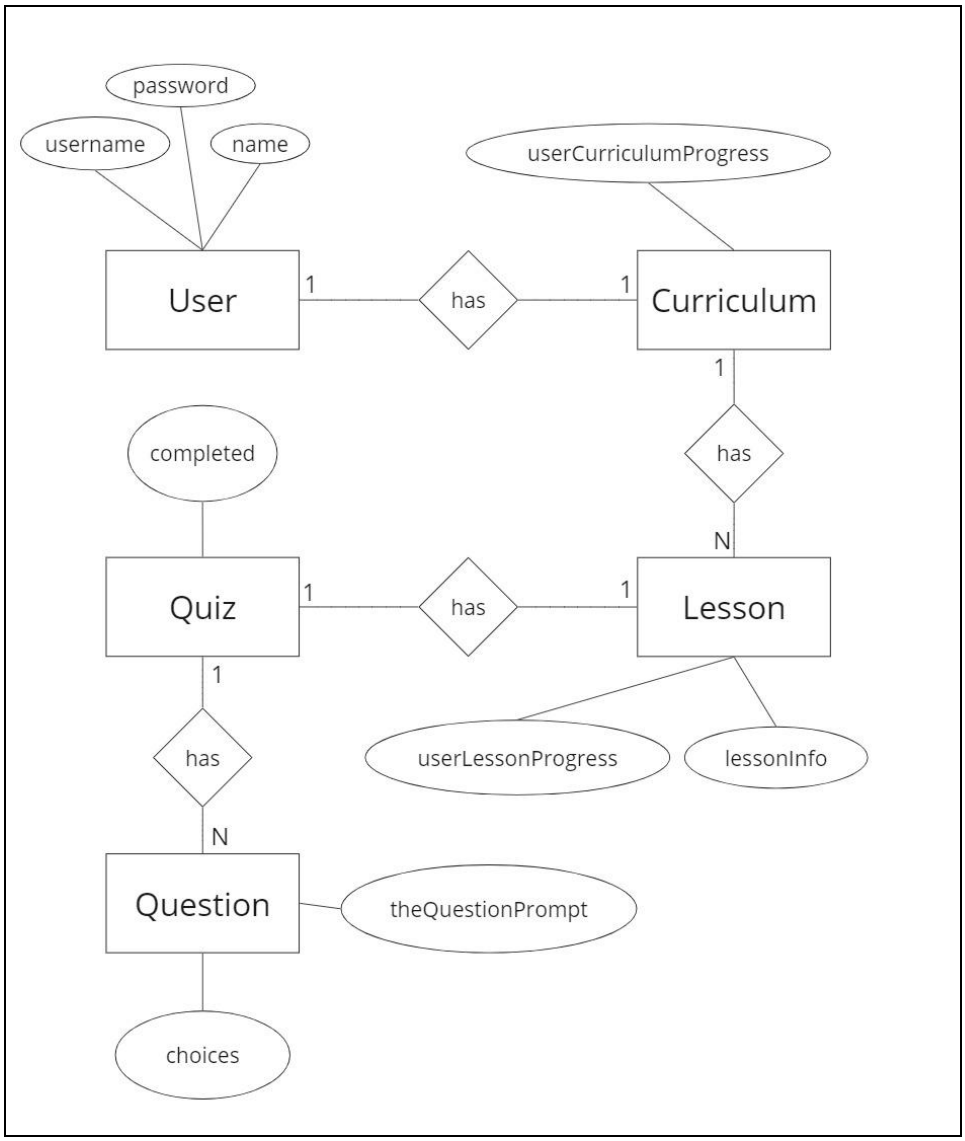


FIG 4.0 User Info Database Diagram*: The diagram above shows the layout of the User Info Database. Objects are represented by boxes, and the data those objects possess are represented by ovals. The lines represent a relationship between two components while the number or 'N' represent the ratio between objects. Object to data ratio is always one to one so a number is not used.*

Table 1: User Info Database Table

Table 1 provides more detail for each element seen in Figure 4.0 above.

Name	Data-type	Size	Description
User	object	varies	Holds the information username, password, name, and a Curriculum object that pertains to a specific user
username	string	16 bytes	The username of a user
password	hash value	16 bytes	The password of a user
name	string	35 bytes	The preferred first name of a user
Curriculum	object	varies	Holds the information userCurriculumProgress and has a Lesson object for every lesson in the PEERS curriculum
userCurriculumProgress	int	4 bytes	Indicates which lesson number the user is currently on
Lesson	object	varies	Holds in information lessonInfo, userLessonProgress, and a Quiz object
lessonInfo	string	varies	The link to the lesson data in the Lesson Content Database
userLessonProgress	dictionary	varies	Holds the information of whether a user has watched each video of a lesson. The key is the video ID and the value is a boolean representing if the video has been watched or not.
Quiz	object	varies	Holds the information completed and one or more Question objects
completed	boolean	1 bit	Indicated whether quiz has been completed or not
Question	object	varies	Holds the information theQuestionPrompt and choices
theQuestionPrompt	string	varies	The link to the question prompt in the Lesson Content Database
choices	string	varies	The link to the choices of the question in the Lesson Content Database

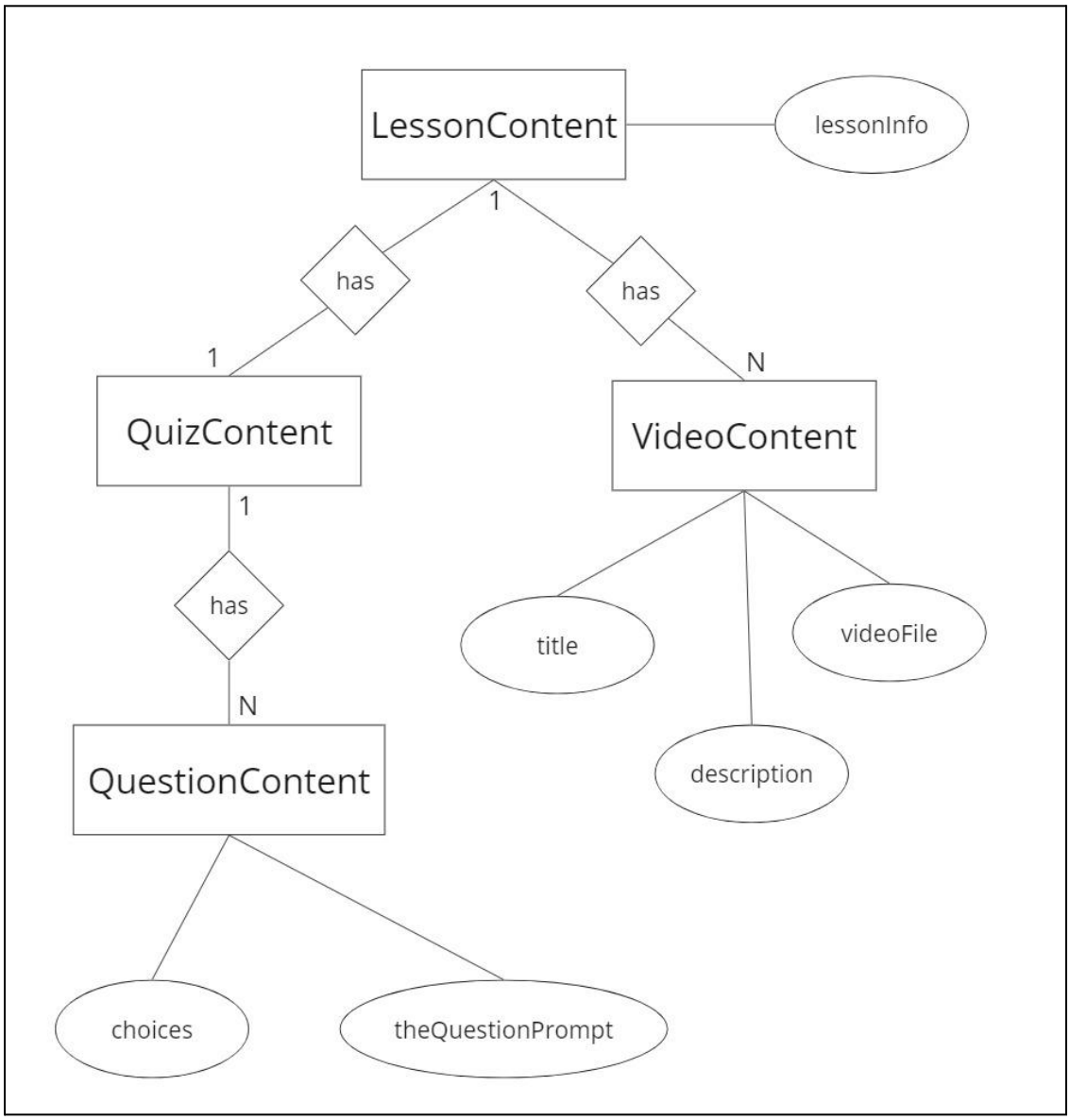


FIG 5.0 Lesson Content Database Diagram: The diagram above shows the layout of the Lesson Content Database.*Objects are represented by boxes, and the data those objects possess are represented by ovals. The lines represent a relationship between two components while the number or 'N' represent the ratio between objects. Object to data ratio is always one to one so a number is not used.*

Table 2: Lesson Content Database Table.

Table 2 provides more detail for each element seen in Figure 5.0 above.

Name	**Data-type**	**Size**	**Description**
LessonContent	object	varies	Holds the lessonInfo file, multiple VideoContent, and one QuizContent
lessonInfo	txt file	varies	Holds additional lesson information from the PEERS curriculum if necessary
VideoContent	object	varies	Holds the title, description and the video file for a specific video
title	string	varies	The title of the video
description	string	varies	A brief description of the video
videoFile	video file	varies	The video file
QuizContent	object	varies	Holds multiple QuestionContent
QuestionContent	object	varies	Holds theQuestionPrompt and the choices for a specific question
theQuestionPrompt	string	varies	The test of the question
choices	dictionary	varies	The choices are accessed by a key and holds both the string text of the choice and a boolean that indicates if the choice is the correct or incorrect answer to the pertaining question

3.2 File Descriptions

This section includes a diagram of the system's file structure followed by a detailed table describing the name, data type, and size of each and every file. Our GitHub repository holds files that are relevant to the structure of the repository, electronic records of our deliverables, a detailed inventory of our VR hardware and a directory for everything specific to our Unity project.

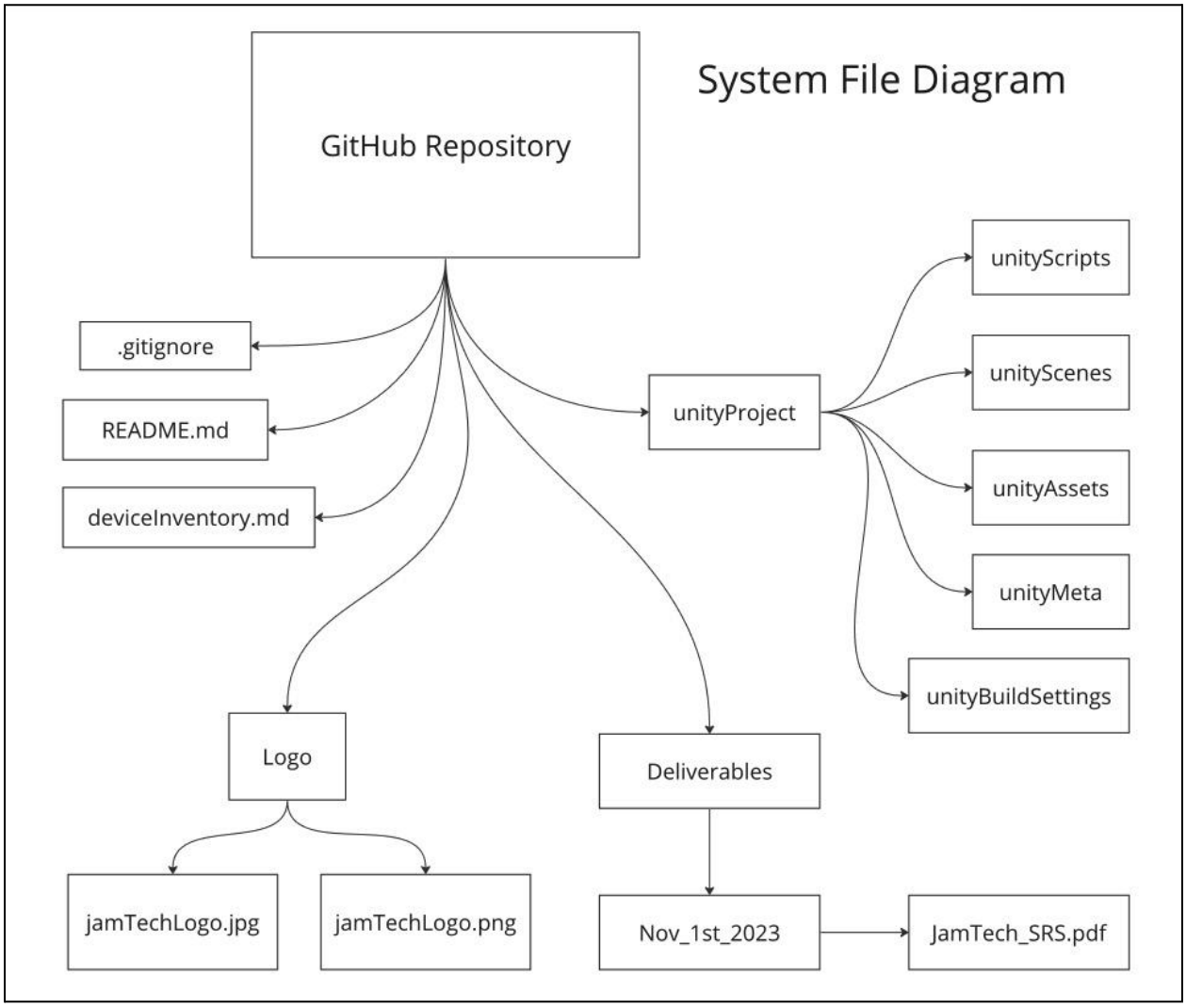


FIG 6.0 System File Diagram: The diagram above shows our system's file structure, which will be available in Github. The diagram shows the various folders and files represented by boxes, with arrows indicating what folders will contain which documents.

Table 3: File System Table

Table 3 provides more detail for each element seen in Figure 6.0 above.

File name	Data-type	Size	Description
README.md	Markdown Text	1 KB	An introduction to our team and the project overall
deviceInventoryVR.md	Markdown Text	1KB	A detailed inventory of our VR equipment
Logo	Directory	53 KB	Contains .jpg and .png images of team logo for use in documentation
Deliverables	Directory	400 KB	Contains .pdf versions of our deliverables and documentation.
.gitignore	Git Preferences	0.5 KB	Detailed list of files not to be included in our GitHub Repository
Unity Scripts	C# code	TBD	Custom scripts written for our Unity project
Scene & Asset Files	Various	TBD	Unity uses scene files and various asset files to define the game world, characters, textures, models, and other game assets
Project Settings	Text files	TBD	Unity's project settings are stored as text files in the ProjectSettings folder.
.meta files	Unity Specific	TBD	Unity generates .meta files for each asset to maintain metadata and connections between assets.
Build Settings	Unity Specific	TBD	Any custom build settings or build configurations that are critical to our project

4 . Requirements Matrix

This section refers to the functional requirements discussed in section 2 of the SRS (system requirements specification) and this matrix is represented in a tabular format to refer the functional requirements to its systems components.

Table 4: Functional Requirements to System Components

Table 4 references each functional requirement as developed in the SRS document and provides the name of the requirement, a brief description and lists all related system components.

ID	Name	Description	System components
FR-1	Log into PEERs account	Allow a user to enter an email and password to log into their PEERs account	UserInfoDatabase
FR-2	Display curriculum map	Allows the user to view the curriculum map	GetCurriculumProgress(UserID)
FR-3	Lesson selection from curriculum map	Allow the user to select a specific lesson from their curriculum map to enter the ‘lesson overview’ mode	SelectLesson( )
FR-4	Viewing role-play videos	User has selected a lesson, and now they are presented with a selection of videos from which the user can select to watch	PlayVideo( )
FR-5	Complete lesson assessment	Allow a user to complete a lesson assessment (quiz)	MakeChoice( )
FR-6	Display tutorial and safety warning	The system displays a simple tutorial and safety warnings to new users after logging in for the first time	HeadsUpDisplayComponent
FR-7	Media controls	When the system displays content videos, the media controls should follow standard conventions	HeadsUpDisplayComponent

Appendix A – Agreement Between Customer and Contractor

Upon signing off the agreement between customer and contractor, the customer (Sarah K. Howorth) and contractor (JamTech) agree on the content described in this document, mainly what the product under development is and how it will be developed. In the case of future changes to this document, the document must be re-read and reviewed then approved by all parties through updated signatures and dates.

By typing one's name under the signature column and giving the date, the individual signs this document.

Name	Signature	Date
Allison Lupien	Allison Lupien	11/15/23
Jacob Michaud	Jacob Michaud	11/15/23
Maha Fazli	Maha Fazli	11/15/23
Nick Sarno	Nick Sarno	11/15/23
Tristan Cilley	Tristan Cilley	11/15/23
Sarah K. Howorth	Sarah Howorth	11/15/2023

Customer Comments:

Appendix B – Team Review Sign-off

This is the team review sign off meaning that all current team members of JamTech (Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, and Maha Fazli) have fully reviewed and read the system design document and do agree with the content and format included in the document.

By typing one's name under the signature column and giving the date, the individual signs this document.

Name	Signature	Date
Allison Lupien	Allison Lupien	11/15/23
Comments:		
Jacob Michaud	Jacob Michaud	11/15/23
Comments:		
Maha Fazli	Maha Fazli	11/15/23
Comments:		
Nick Sarno	Nick Sarno	11/15/23
Comments:		
Tristan Cilley	Tristan Cilley	11/15/23
Comments:		

Appendix C – Document Contributions

This is the current contribution of each team member towards the system requirements specification document.

Name	% of contribution
Allison Lupien	20% [3.1: entity-relationship diagrams, editing]
Jacob Michaud	20% [2.1,2.2, uml diagram]
Maha Fazli	20% [4, intros for all sections]
Nick Sarno	20% [proofreading, editing, formatting]
Tristan Cilley	20% [3.2, uml diagram, system flow diagram, file diagram]