Critical Design Review Document

for

Virtual Video Modeling on the Social Skills of Adults with Autism

Version 2.0 approved

For Sarah K. Howorth

University of Maine

Dec 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

Critical Design Review

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. 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 following description comes from the website for the UCLA PEERS® Clinic,

> "The Program for the Education and Enrichment of Relational Skills (PEERS®) is world-renowned for providing evidence-based social skills treatment to preschoolers, adolescents, and young adults with autism spectrum disorder (ASD), attention deficit/hyperactivity disorder (ADHD), anxiety, depression, and other socio-emotional problems." (PEERS®, 2023)

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 (eg; communication, humor, dating etiquette, etc) 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

This document shall encapsulate all of our design and planning thus far and include a plan of implementation for the next semester. This document is split into five sections. Section 1 - Introduction introduces the project, including its general scope, purpose, background, constraints, and objectives. Section 2 - Project Management outlines our team roles and management strategies. Section 3 - Implementation Plan goes into detail about all of the deliverables for this project, GitHub and version control, a risk assessment, plans for the spring semester, and our CI/CD pipeline. Section 4 - System & Database Design goes over general aspects of the project included in the SRS and SDD such as use cases, system requirements, architecture, the UML

diagram, and database design. Section 5 - User Interface & Accessibility quickly summarizes our UI flow diagram and the user interface aspect of our project.

Included in the document are also Appendices B, C, and D, which correspond to the SRS, SDD, and UIDD respectively. These appendices include the full documents and all information and diagrams contained within them.

1.2 Product Scope

The scope of the product is to teach social skills i.e. (conversational skills, dating, handling teasing and bullying). Additionally, the system is designed to be able to provide users with a virtual reality (VR) interface to be able to learn social skills in an engaging way.

1.3 Project Goals

For this project, JamTech plans to develop a proof of concept. To do so, we will develop the login system, the curriculum map and 2-3 lessons including the overview, the video player and the quiz within a VR environment. This development will take place between January and April 2024 during the Spring semester. The proof of concept will be available by the Student Symposium on April 12th, 2024 where students present their final capstone projects. Section 3 provides a more detailed description of our planning for the Spring semester.

1.3.1 Completion Requirements

In order to complete this project over the next several months, we will need to complete the following features of our project in order: A Video View, Lesson View, Quiz View, Content Database, Curriculum View, User Info Database, and Login features. In order to complete this project, all of these tasks must be completed by the end of the spring semester. Each task will be completed in consecutive 2 week sprints where the first third of the sprint is dedicated to planning, then ⅛ coding, followed by ¼ component testing and ¼ system testing. Allowing this portion of the 2 weeks to be spent ensures that the aspect of the project to be completed is well

thought out and properly tested. If time permits once all sprints have been completed, we will focus on implementing additional features pertaining to accessibility and inclusivity.

1.4 References

This section includes a list of documents and other media related to this project.

- Gurney, L. (2023). Lecture 4. SDLC. Orono; Maine.
- JamTech.(2023) . Figma, https://www.figma.com/file/Iosmhnt2IlNk7ggfHybAt9/PEERS?type=design&node-id=0%3A1&mode=design&t=sEC0nLgmgZfr62xM-1
- JamTech. (2023). Miro, https://miro.com/app/board/uXjVNShE4HY=/?share_link_id=215448921420
- Laugeson E. A. (2014) The PEERS Curriculum for School-Based Professionals, Table 1.1 (1st Edition) https://www.routledge.com/The-PEERS-Curriculum-for-School-Based-Professionals-Social-Skills-Training/Laugeson/p/book/9780415626965
- PEERS®. (2021). PEERS® (version 1.1.0) [Mobile app]. Apple Store OR Google Play. https://play.google.com/store/apps/details?id=com.peersclinic.peers&hl=en_US&gl=US
- PEERS (2023) UCLA PEERS® Clinic, Semel Institute for Neuroscience and Human Behavior. https://www.semel.ucla.edu/peers
- 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). System Design Document (SDD), version 1.0
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2023). Systems Requirement Specification (SRS), version 2.0
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2023). User Interface Design Document (UIDD), version 1.0

2. Project Management

This section will provide an overview of our team's project management strategy, and will cover topics such as: roles and responsibilities, and team management strategies. These topics will further expand on the responsibilities of the team, as well as the successful strategies that have been implemented throughout the semester.

2.1 Roles & Responsibilities

Each member of the team has volunteered for and decided on a specific role in order to delegate responsibilities to everyone on the team. All members have had equal participation in the documentation of the SRS, SDD, and UIDD. The general responsibilities for the team are as follows:

Tristan Cilley	- Meeting leader, scrum master, and DevOps, documentation
Maha Fazli	- Scheduling, documentation, and UI designer
Jacob Michaud	- Lead developer and UML designer, documentation
Allison Lupien	- Documentation and database management/design
Nick Sarno	- Client communications, presentations, and documentation

2.2 Team Management Strategies

Our team has had meetings every Monday and Saturday, where we discuss work that we need to work on in the future, as well as doing the work that has been assigned to us together. This has been very effective and it has made sure that our lines of communication have been strong throughout the creation of this project. We have a biweekly review for each of our team members and the group as a whole in order to gauge how well the group is doing and make suggestions to members that need to step up in order to meet expectations, however there has been no issues thus far.

3. Implementation Plan

This section will give an overview of JamTech's plan and strategies to develop this system during the spring semester.

3.1 Deliverables

This section will contain information about the deliverables we have completed this semester (Fall 2023), and upcoming deliverables for the Spring 2024 semester. Physical copies of important documents will be provided as a hard copy, and digital copies will be provided as a pdf. Most deliverables will be subject to updates and modifications over the semester.

Our deliverables this semester included the System Specification Requirements (SRS), the System Design Document (SDD), and the User Interface Design Document (UIDD). Each document was created in succession, using information from the previous document to compose the next. The SRS was created first to present an overview of the entire project, detailing function requirements, use cases, system diagrams, and acceptance tests for the proposed system. Using the information from the SRS, the SDD was then created to describe the system's design and architecture, system flows, filesystems, and databases. Within this document, system components from the UML diagram were paired with functional requirements from the SRS. Lastly, using the flow diagrams and layouts stated in the SDD, the UIDD was created as a guide for the system's user interface and visual design complete with user interface standards, a user interface walkthrough, and data validation.

Deliverables for Fall 2023 - [COS397]

An electronic file containing the following:

** Documents are available on Github **

- Systems Requirement Specification, Version 2.0 **(available: Nov 1, 2023)**
- System Design Document, Version 1.0 **(available: Nov 15, 2023)**
- User Interface Design Document, Version 1.0 **(available: Dec 5, 2023)**
- Critical Design Review Document, Version 1.0 **(available: Dec 15, 2023)**

Deliverables for Spring 2024 - [COS497]

Hard copies of each of the following:

- User Manual
- Administrator Manual

An electronic file containing the following:

** Documents hosted on Github will be updated regularly **

- Critical Design Review Document **(Estimated availability: TBD)**
- Code Inspection Report **(Estimated availability: TBD)**
- Final Project Report **(Estimated availability: TBD)**
- User Manual **(Estimated availability: TBD)**
- Administrator Manual **(Estimated availability: TBD)**
- All source code **(Estimated availability: TBD)**
- The executable program **(Estimated availability: TBD)**
- Any other software required for installation and execution of the delivered program. **(Estimated availability: TBD)**

3.2 GitHub & Version Control

JamTech has chosen GitHub as our version control solution. Here is a link to our Github repository: https://github.com/VoloVita/SeniorCapstone

The main branch of the repository contains the most stable working version of our code and the latest copies of our deliverables. The deliverables are stored in individual directories labeled by the date we submitted them to the professor for grading. On the next page, there is a

diagram (Figure 1) of our GitHub repository as well a table (Table 1) detailing the specifics of its contents. These are evolutions of the file system diagram (figure C6.0) and the accompanying table (Table C3) from section 3.2 of the SDD. (See Appendix C)

Once the repository has been cloned, anyone can open it as a Unity Project using Unity Hub. The first time the project is opened, Unity will generate the Library, Logs, and User Settings directories. These directories are excluded by the .gitignore file and will not be included in pushes to any branch. This is done to reduce the size of the repository. Only the Assets, Packages, and ProductSetting directories will be stored on Github. This is sufficient to build the entire development environment for individual contributors.

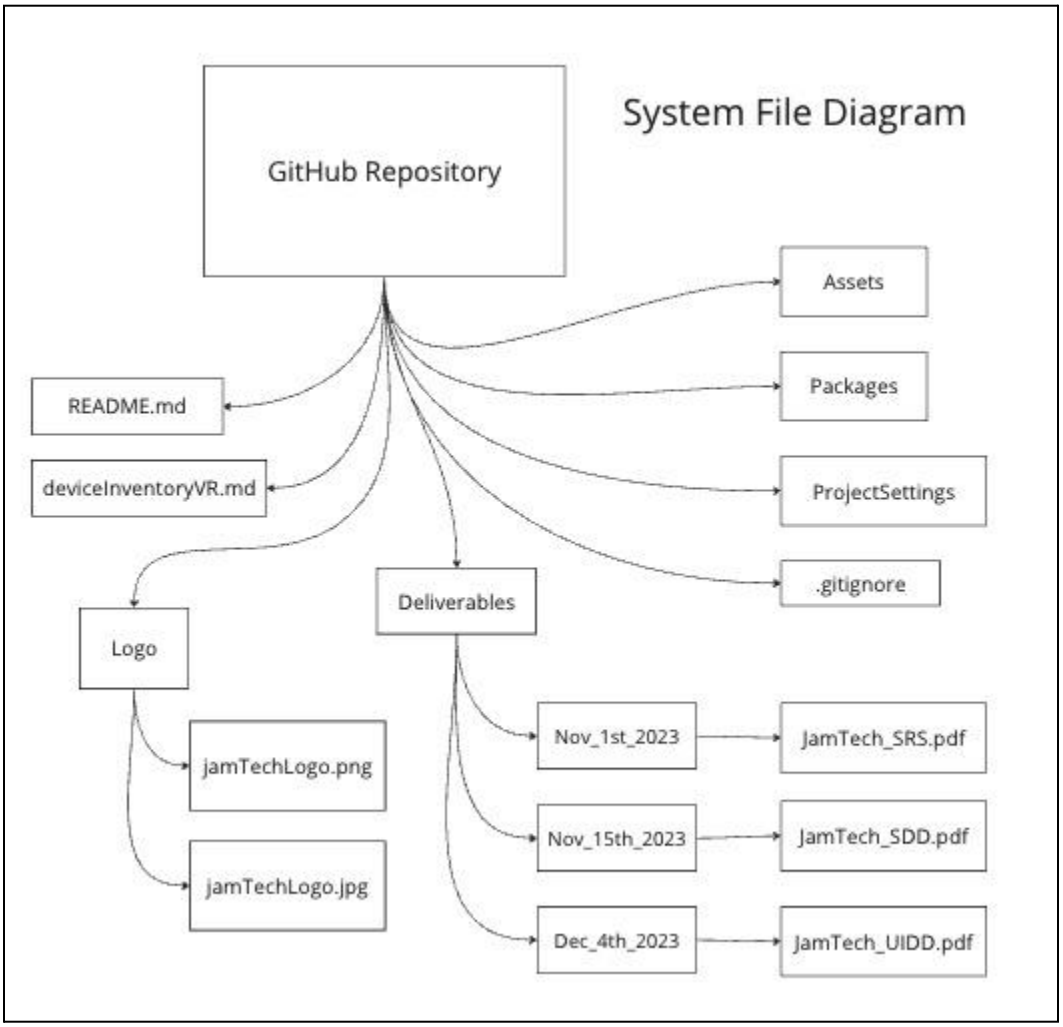


Figure 1 File System: *The file system organization on the JamTech GitHub repository.*

Table 1: File System Table

Table 1 provides more detail for each element seen in Figure 1.0 above.

File name	Data-type	Size	Description
README.md	Markdown Text	1 KB	An overall introduction to JamTech and the project.
deviceInventoryVR.md	Markdown Text	1KB	A detailed inventory of VR equipment made available to JamTech by Dr. Sarah Howorth
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
Packages	Various	TBD	Contains the Unity Package Manager (UPM) packages. These packages are assets, libraries, or tools that you can add to a Unity project to enhance functionality or include third-party features.
Asset Files	Various	TBD	Unity uses scene and asset files to define the game world, characters, textures, models, and other game assets.
ProjectSettings	Text files	TBD	Unity's project settings are stored as text files in the ProjectSettings folder.

3.3 Initial Risk Assessment

JamTech conducted a risk analysis for this project and categorized the potential risks on a scale from low to high risk. The risk analysis showed that the highest risk for this project is the user getting motion sick or falling over due to the VR headset. The consequences for this would have the greatest impact on our project, and to mitigate this risk we have decided to require the user to be seated while using our software.

The second highest risk includes damages to the VR equipment which would then need to be replaced. The consequences for this risk would be financial losses and time to replace the VR headset. As a response to mitigate this risk we created our functional requirements (FR) for creating a safety tutorial.

The third risk assessed was determining if our github repository had broken or unusable code that made the project nonfunctional. We plan to mitigate this by creating our DevOps workflow and by using GitHub Actions to test and lint our code, as well as maintain consistent formatting.

A risk that was proposed to our scheduling was having delayed deliverables for our product. This has a low consequence due to it only impacting our class grades. To mitigate this, we have been engaging in consistent scrum meetings and schedules made, as well as having direct communication with each team member consistently. Work can be redistributed as needed between team members to meet scheduled deadlines.

Finally our last risk is the possibility of our databases being breached and our users' account credentials being compromised. This is also a low risk due to the nature of our databases, and that we don't keep track of much if any sensitive information. To mitigate this we designed our databases to encrypt sensitive information.

3.4 Spring Semester Plan for Code Development

This section identifies the implementation plan for coding in the spring semester. It explains our development strategy using a mix of spiral and incremental and the backbone of the development timeline.

3.4.1 Spiral Incremental Strategy:

Our spiral iterative development strategy divides the spring semester into seven discrete sprint cycles. Each sprint begins by determining our objectives to accomplish that sprint and updating our risk assessment by identifying and resolving any risks. Once properly oriented, we create a new feature branch in our repository and begin programming our project's next iteration. During the development phase, we will divide our labor between writing source code and writing tests for the new feature. Before progressing to the next sprint, we will meet to discuss the successes and challenges of the previous sprint and plan our intentions for the upcoming sprint. This process is summarized in Figure 2.

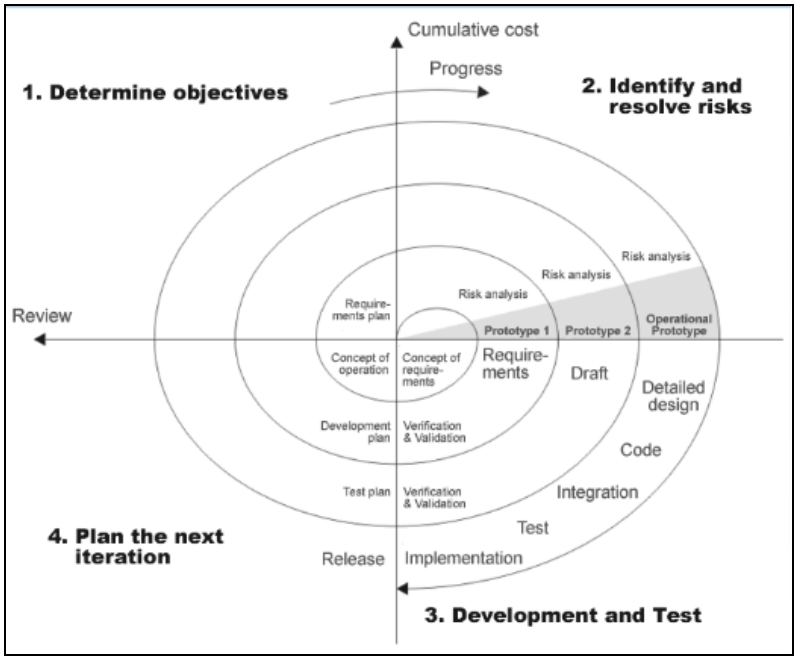


Figure 2 Spiral Strategy*: Beohem's spiral model from Lecture 4 SDLC by Dr.Gurney.*

3.4.2 Development Timeline:

There are seven main development milestones identified for this project as shown in Figure 3. Each one will be the focus of a two week sprint for a total of 14 weeks in the semester. In addition to the main parts, more functionality may be addressed or assigned in a sprint. This includes functionality like a tutorial for first time users or more accessibility elements.

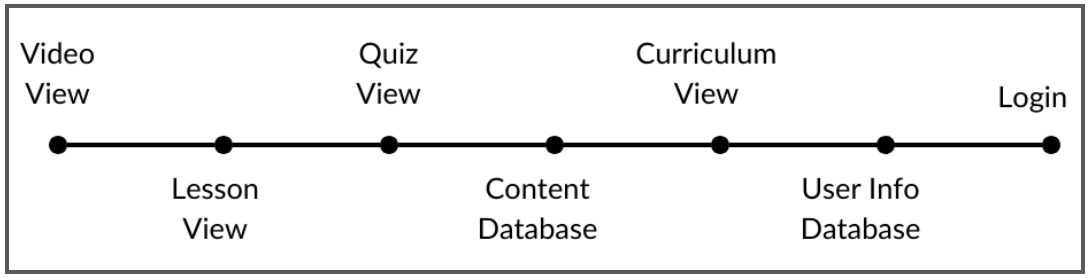


Figure 3 Timeline*: This timeline identifies the seven main parts of the system that need to be developed.*

3.4.3 Technical Difficulty:

In this section, we will be evaluating each of the milestones in the development timeline above based on our expectations of their technical complexity. Each milestone will be rated on a scale from 1-5 where a rating of 1 indicates an expectation of the milestone being fairly simple while a rating of 5 indicates an expectation of the milestone being exceedingly difficult.

1. **Video View** **3/5**

   Implementing the video view should be a straightforward design process. The additional setup required to display video content in a scene should also be simple.

2. **Lesson View** **3/5**

   Designing the lesson view should also be a straightforward design process, however displaying multiple video thumbnails in a scrollable carousel will require more time.

3. **Quiz View** **2/5**

   The quiz view will be similar to the video view in difficulty and features, once we have the video view it should be trivial to extend its features to the quiz view.

4. **Content Database** **4/5**

   Creating the content database for this project is one of the harder features we need to make. We still require more planning before we decide how we will store and transfer our content to the project. This will take the majority of this sprint's effort.

5. **Curriculum View** **2/5**

   This milestone should be able to be completed quickly as the functionality is not complex, the only struggles could come with figuring out how to set up the project to integrate the User Info Database in the future.

6. **User Info Database** **5/5**

   This will be the most difficult part of the project to implement. It is also one of the most important in order to ensure that each user has their personal information and content progress saved, which will require work on all other aspects of the project made thus far. It will require a lot more testing than the other sections.

7. **Login** **3/5**

   After implementing the database, the login should be somewhat difficult to implement, but with the framework so far we should be able to create an interface to login that works with the whole project properly.

8. **Extra Features** **2/5**

   Finally, the additional features that may be added include inclusivity and accessibility features that are easy to implement and shouldn't require much testing or time to complete.

3.5 CI/CD Pipeline

Our continuous integration and continuous delivery (CI/CD) pipeline will be a combination of the Unity Test Framework package and Github Action.

The Unity Test Framework allows for Edit Mode tests, similar to standard unit tests, and Play Mode tests, which are similar to integration tests. Edit Mode tests allow developers to send specific input to functions and then check the corresponding output for validity. Play Mode tests are run as coroutines during the runtime of the Unity project, simulating complex actions or series of user inputs. All tests are run by the Unity Test Runner which uses an integration of an open-source unit testing library for .Net languages called the NUnit library. The test script files also need to be referenced in either the Edit Mode or Play Mode assembly definitions.

Once the tests have been written, we can modify the YAML file that defines our Github workflows to automatically run our Unity Test Runner and build the project. Additionally we can add linting and format checking jobs to the workflow to ensure code quality and readability.

4. System & Database Design

This section will describe elements of the system detailed in the SRS and SDD such as use cases, functional and nonfunctional requirements, architecture, and important diagrams containing relevant information to the construction and design of the system.

4.1 Use Cases

The use cases in this section describe what the system will be capable of doing. Each use case is based on a function our client requested be implemented into the system, put in simple terms and expressed as a desire to have the system do something from the perspective of the client. There are 7 use cases included in our system which appear in Figure B2 in section 1.3 of Appendix B - SRS. That figure depicts how the user interacts with the system in the context of the use cases.

Use cases below take the form UC-# of the use case: Name - a use case statement.

UC-1: Login - As a user, I want to be able to login to my PEERS account.

UC-2: See Curriculum Map - As a user, I want to be able to see my progress by viewing my curriculum map.

UC-3: Select Lesson - As a user, I want to be able to select a lesson to work on.

UC-4: Watch Video - As a user, I want to be able to watch lesson videos.

UC-5: Complete Quiz - As a user, I want to be able to complete a lesson quiz.

UC-6: Watch Tutorial - As a user, I want to be able to watch a tutorial video on how to use the system the first time I login.

UC-7: Interact with Video Controls - As a user, I want to be able to interact with standard media controls while playing a video.

4.2 Functional Requirements

The functional requirements below are written with the intended purpose of what the system will offer, as well as the requirements that the system will follow to provide an interface for the user to interact with. Each functional requirement is followed by a more detailed use-case table in section 2 of Appendix B - SRS. The priority number for each functional requirement is included in its corresponding table.

Functional Requirements follow the format: FR-(unique identification number): requirement description.

FR-1: The system shall allow users to log into their PEERs account by entering their email and password.

FR-2: The system displays the curriculum map to the user.

FR-3: The system shall allow the user to select a lesson from the curriculum map.

FR-4: The system shall allow the user to view lesson-related video role-play examples in the VR environment.

FR-5: The system shall allow users to complete lesson assessments in the VR environment.

FR-6: The system shall offer a simple VR tutorial and safety warnings to new users.

FR-7: The system shall allow standard media controls for the content videos.

4.2.1 Acceptance Testing

The purpose of acceptance testing is to thoroughly test all of the functional requirements listed above. Each acceptance test will test one or multiple of the functional requirements, ensuring that every aspect of the program will function as expected.

There are seven acceptance tests created for our system to match the number of functional requirements. Included in the acceptance tests are several scenarios where the system is both used as intended and not used as intended, to test how the system will function under such circumstances. To see the acceptance tests for our system, please see Appendix B - SRS section 2.1.

4.3 Non-Functional Requirements

The non-functional requirements below are written to describe characteristics of the system, such as constraints or capabilities. NFR's 1-6 are performance requirements, NFR's 7-9 are safety and privacy requirements, NFR-10 is security, and NFR's 11-13 are software quality attributes.

Non-Functional Requirements follow the format: NFR-unique identification number: requirement description. Priority numbers for each requirement can be found in section 3 of Appendix B - The SRS.

NFR-1: The system shall display the lesson to the VR headset within 0.5 seconds, 99% of the time.

NFR-2: The system shall accurately keep track of the user's curriculum progress 99% of the time.

NFR-3: The system shall be able to handle up to 100 active users 95% of the time.

NFR-4: The system shall be able to handle 100 login requests every 5 seconds 95% of the time.

NFR-5: The system shall be able to handle the user input within 0.5 seconds 95% of the time.

NFR-6: The system shall be able to update based on user actions within 0.5 seconds 95% of the time.

NFR-7: The system shall comply with the California Consumer Privacy Act. (California)

NFR-8: The system shall comply with the General Data Protection Regulation. (European Union)

NFR-9: The system shall protect user's data from unauthorized access.

NFR-10: The system shall be able to protect user data from attacks 99% of the time.

NFR-11: The system shall be able to connect to internet services 95% of the time.

NFR-12: The system shall be available to all users 24 hours a day, 95% of the time.

NFR-13: The system shall not exceed a size of 20GB.

4.4 System Architecture

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. Appendix C - SDD contains a full copy of our SDD. A software architecture/flow diagram, which can be found as Figure 4 below or Figure C1.0 in Appendix C - SDD, depicts how the system is planned to flow in terms of interacting with it as a user. The technology architecture diagram, which can be found as figure C2.0, depicts interactions between the hardware and software components.

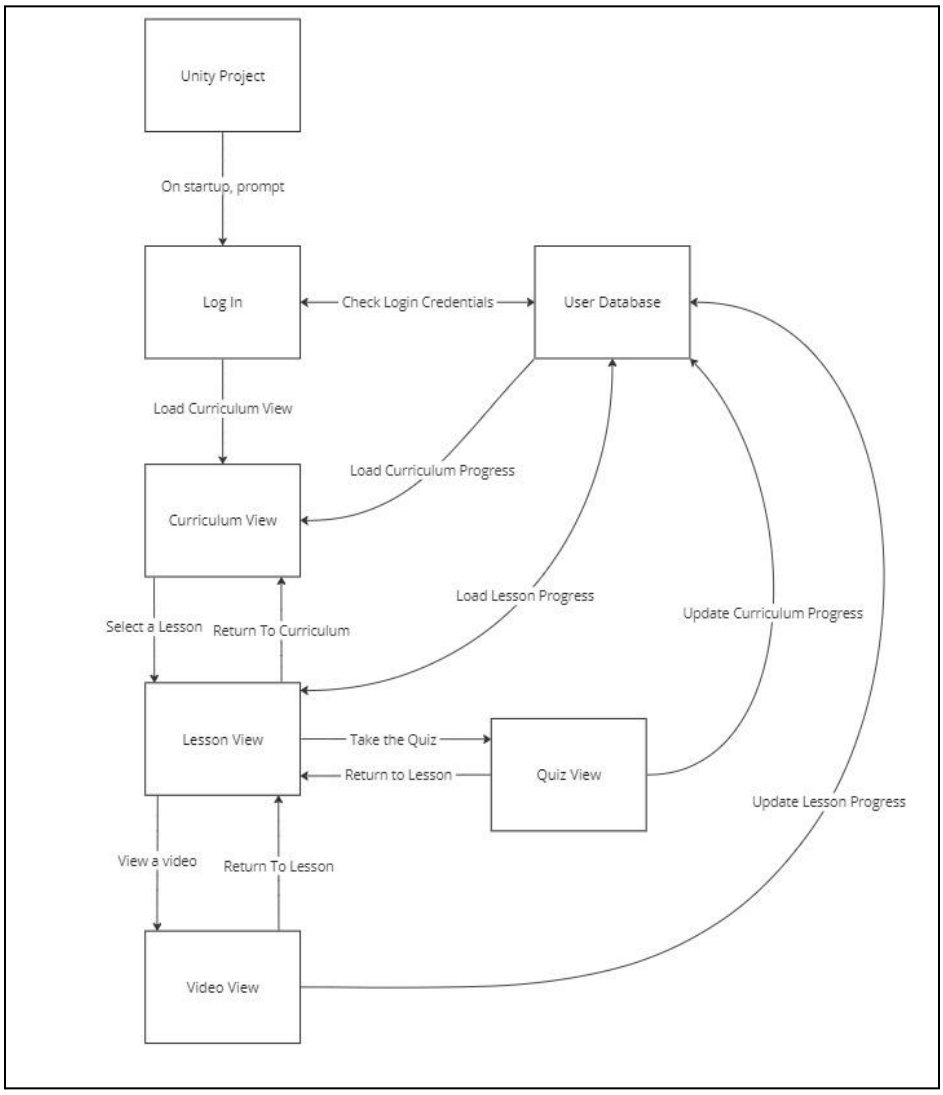


Figure 4 Software Architecture/Flow Diagram*: This graph showcases how the software parts interact with each other with the individual elements represented by boxes and arrows indicating interaction actions connecting everything.*

4.5 UML Diagrams

The current system architecture of the system can be represented by a context diagram and a UML diagram, which can be found as figure C3.0 in Appendix C - SDD. This 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.

4.6 Database Design

Detailed within section 3.1 of Appendix C - SDD are two separate database diagrams. Figure C4 is the User Info Database diagram which will store the user's information like username, name, password, curriculum progress, etc. Figure C5 is the Lesson Content Database which includes lesson content such as videos and quizzes. Both figures are followed by tables detailing the specifics of each diagram. They are Figure C1 and Figure C2 respectively.

5. User Interface & Accessibility

Appendix D - UIDD includes all information pertaining to the UI design and presentation of the system, including user interface standards such as logical, common, and visual components; accessible and inclusive design elements; and a full user interface walkthrough. The user interface walkthrough is a comprehensive guide through our program including visual diagrams and explanations of what is included in each screen. Below is Figure 5 UI Navigation (Figure D1 in Appendix D - UIDD) which is the detailed flow diagram. This is a navigation diagram for the various UI views incorporated in the project.

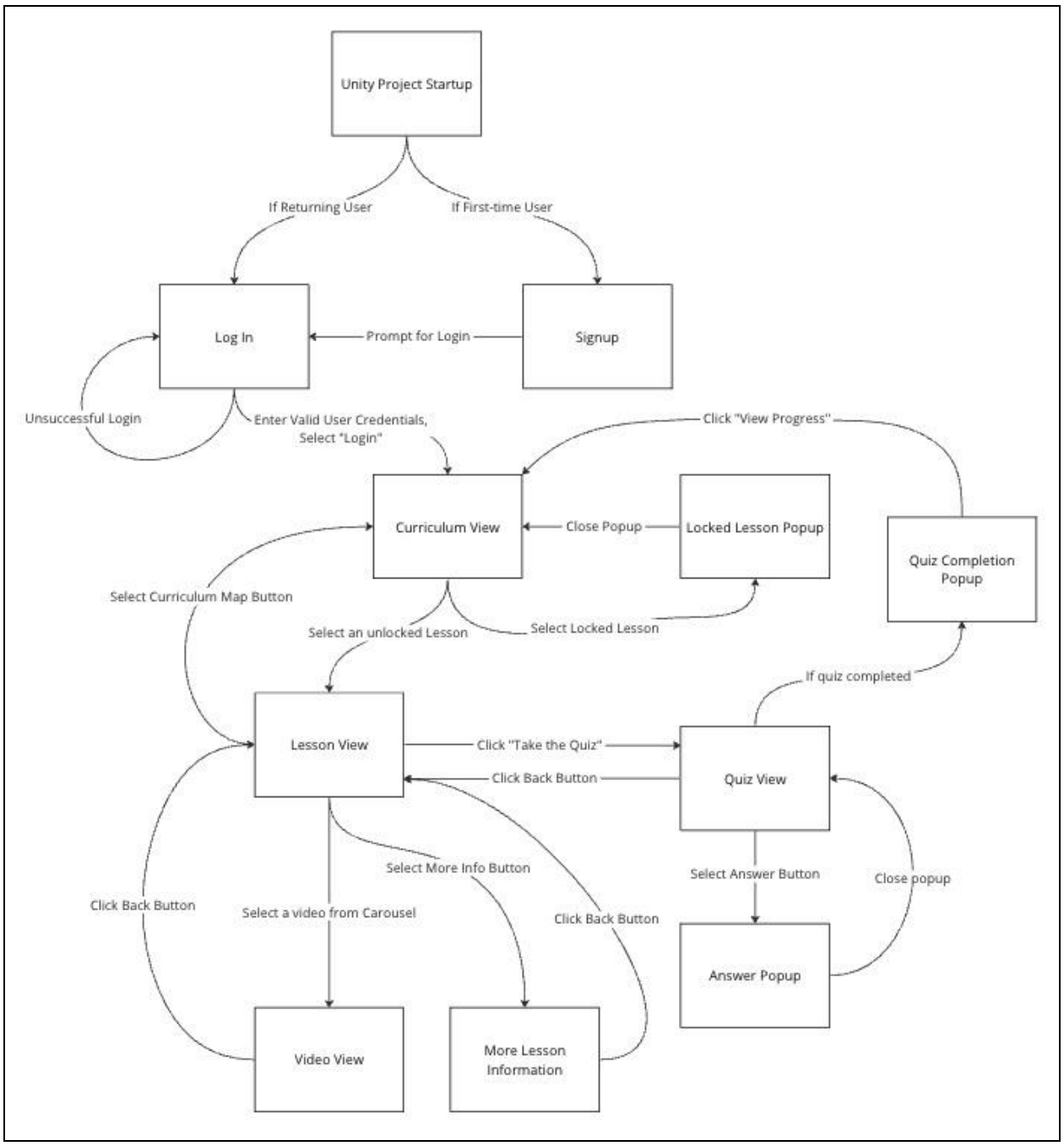


Figure 5 UI Navigation: *A detailed flow diagram depicting how a user is expected to use UI elements to navigate through the project.*

Appendix A - CDR

A.1 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, as well as what the team plans to achieve with this project given the current time frame January - May.

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	12/14/23
Jacob Michaud	Jacob Michaud	12/14/23
Maha Fazli	Maha Fazli	12/14/23
Nick Sarno	Nick Sarno	12/14/23
Tristan Cilley	Tristan Cilley	12/14/23
Sarah K. Howorth	Sarah K. Howorth	12/14/23

Customer Comments:

A.2 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 agreed upon what will be required to complete the proof of concept for this project.

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	12/14/23
Comments:		
Jacob Michaud	Jacob Michaud	12/14/23
Comments:		
Maha Fazli	Maha Fazli	12/14/23
Comments:		
Nick Sarno	Nick Sarno	12/14/23
Comments:		
Tristan Cilley	Tristan Cilley	12/14/23
Comments:		

A.3 Document Contributions

This is the current contribution of each team member towards the CDR.

Name	% of contribution
Allison Lupien	20% [Introduction, References, Appendix D, Implantation Plan, Revisions]
Jacob Michaud	20% [2 Project Management, completion requirements, Appendix C, revisions]
Maha Fazli	20% [UIDD, project management intro, revisions]
Nick Sarno	20% [4.0 System & Database Design - 5.0 User Interface & Accessibility, Rewriting intro paragraphs, Revisions]
Tristan Cilley	20% [Introduction, Purpose of Document, DevOps, Version Control, CI/CD, Appendices]

System Requirements Specification

for

Virtual Video Modeling on the Social Skills of Adults with Autism

Version 2.0 approved

For Sarah K. Howorth

University of Maine

Nov 27, 2023

Prepared by JamTech:

Tristan Cilley, Allison Lupien, Nick Sarno,

Jacob Michaud, Maha Fazli

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.

Dr. Sarah Howorth, is an associate professor of special education and program coordinator for the special education graduate programs at the University of Maine and she is also 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

This document provides an initial overview of the entire project. Key areas of design such as functional requirements, non-functional requirements, use-case modeling, system diagrams, and acceptance testing shall be covered in greater detail. 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. Our intention is to provide a clear, detailed summarization of this project's requirements and to establish a schedule for upcoming deliverables.

1.2 Purpose of the Product

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.

1.3 Product Scope

The scope of the product, described by Figure 1, is to teach social skills i.e. (conversational skills, handling teasing and bullying). Additionally, the system is designed to be able to provide users with a virtual reality (VR) interface to be able to learn social skills in an engaging way. Figure 2 provides more information on how a user can interact with the system. The requirements of the system describe more precisely what the system will offer to the users. (***See section 2. Functional requirements)***

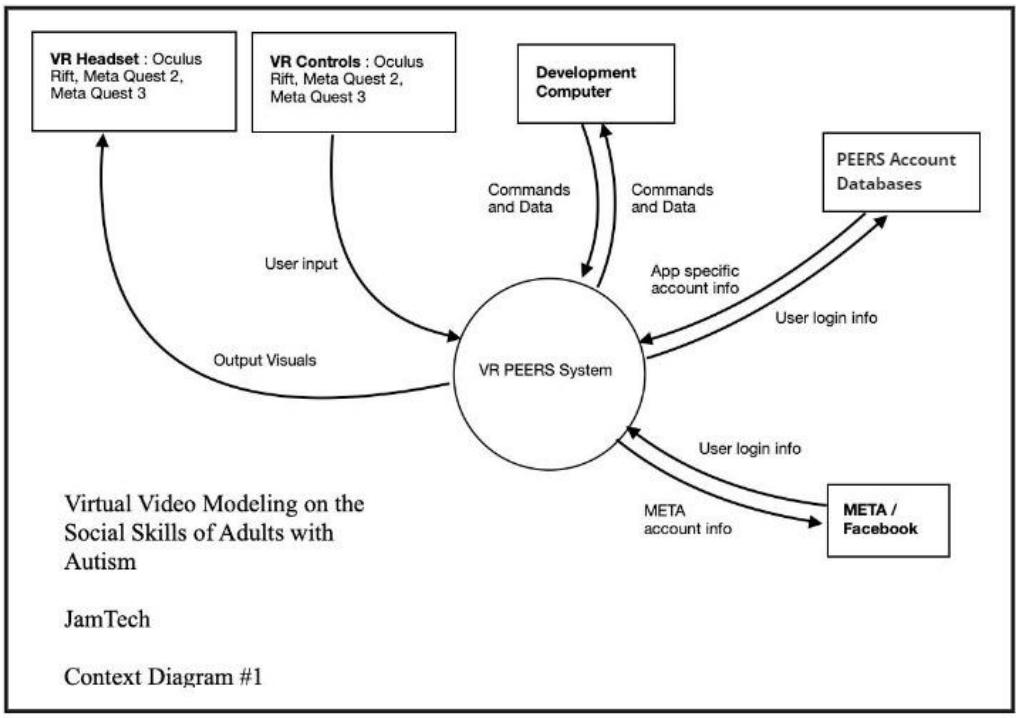


Figure B1 System Context Diagram: *The VR PEERS System will interact with META to log the user into the VR headset, with the PEERS databases to log the user in and get user information, with the development computer to transfer commands and data as well as the headset to display the scene and the controls to get user input.*

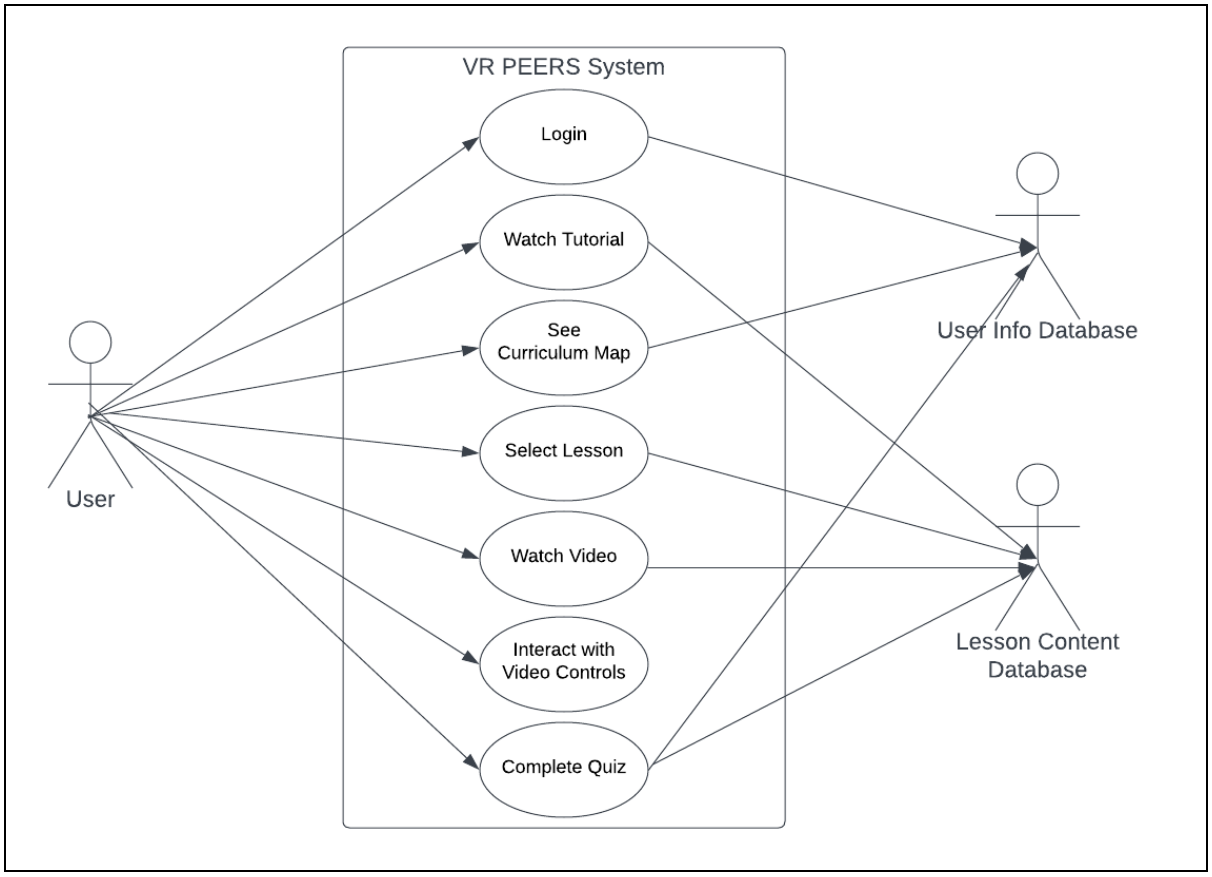


Figure B2 Top Level Use Case Diagram*: This use case diagram depicts the different ways a user would want to interact with the VR PEERS system as well as the additional actors that system interacts with, namely the User Info Database and the Lesson Content Database.*

There are 7 use cases depicted in Figure 2 Use Case Diagram. In general terms, a user of a system can log into the system, watch a tutorial, look at their curriculum map, select a lesson to learn from the map, watch lesson videos with standard media controls and complete lesson quizzes. Use cases take the form UC-# of the use case: Name - a use case statement. Each use case is reflected in the functional requirements. (***See section 2. Functional requirements***)

UC-1: Login - As a user, I want to be able to login to my PEERS account.
UC-2: See Curriculum Map - As a user, I want to be able to see my progress by viewing my curriculum map.
UC-3: Select Lesson - As a user, I want to be able to select a lesson to work on.
UC-4: Watch Video - As a user, I want to be able to watch lesson videos.
UC-5: Complete Quiz - As a user, I want to be able to complete a lesson quiz.
UC-6: Watch Tutorial - As a user, I want to be able to watch a tutorial video on how to use the system the first time I login.
UC-7: Interact with Video Controls - As a user, I want to be able to interact with standard media controls while playing a video.

2. Functional Requirements

The functional requirements below are written with the intended purpose of what the system will offer, as well as the requirements that the system will follow to provide an interface for the user to interact with. The functional requirements are each followed by a use-case table which displays detailed information about the requirement.

Functional Requirements follow the format: FR-(unique identification number): requirement description. The priority number for each functional requirement is included in its corresponding table. The priority number 1 represents the lowest priority and 5 the highest priority.

FR-1: The system shall allow users to log into their PEERs account by entering their email and password.

FR-2: The system displays the curriculum map to the user

FR-3: The system shall allow the user to select a lesson from the curriculum map.

FR-4: The system shall allow the user to view lesson-related video role-play examples in the VR environment.

FR-5: The system shall allow users to complete lesson assessments in the VR environment.

FR-6: The system shall offer a simple VR tutorial and safety warnings to new users.

FR-7: The system shall allow standard media controls for the content videos.

FR-1: The system shall allow users to log into their PEERs account by entering their email and password.

Number	1	
Name	Log into PEERs account	
Summary	Allow a user to enter an email and password to log into their PEERs account	
Priority	5	
Preconditions	User must have a PEERs account	
Postconditions	User is logged in and is granted access to the rest of the application's functions	
Primary Actor	User	
Secondary Actors	Account Database	
Trigger	User selects the "Login" button	
Main Scenario	**Step**	**Action**
	1	A log-in window is displayed
	2	User enters their email and password into the text boxes
	3	User clicks the "Login" button to submit their information
	4	User is logged in
	5	User is brought to curriculum map
Extensions	**Step**	**Branching Action**
	4a	The information entered by the user does not match an account: - A pop-up message alerts the user that the information entered does not match an existing account, prompting them to try again
	5a	The user is new: - Instead of being brought to the curriculum map, the user is shown the tutorial videos first.

FR-2: The system displays the curriculum map to the user

Number	2	
Name	Display Curriculum Map	
Summary	Allows the user to view the curriculum map	
Priority	5	
Preconditions	User must be logged in	
Postconditions	User is presented with their current curriculum map	
Primary Actor	User	
Secondary Actors	Curriculum database	
Trigger	User logs in successfully, or User backs out of a lesson	
Main Scenario	**Step**	**Action**
	1	System queries database for user’s current curriculum map
	2	User is shown the curriculum map

FR-3: The system shall allow the user to select a lesson from the curriculum map.

Number	3	
Name	Lesson Selection from Curriculum Map	
Summary	Allow the user to select a specific lesson from their curriculum map to enter the 'lesson overview' mode.	
Priority	4	
Preconditions	User must be logged in. User is on the Curriculum Map view.	
Postconditions	User is able to view the role-play videos within the lesson	
Primary Actor	User	
Secondary Actors	Curriculum database	
Trigger	User selects a lesson from the curriculum map.	
Main Scenario	**Step**	**Action**
	1	User selects a specific unlocked lesson: lesson overview mode is entered
	2	The lesson overview is displayed
Extensions	**Step**	**Branching Action**
	1a	User selects a specific locked lesson on the curriculum map: - A pop-up window displays a message alerting the user that they have not yet unlocked this lesson

FR-4: The system shall allow the user to view lesson-related video role-play examples in the VR environment.

Number	4	
Name	Viewing Role-play Videos	
Summary	User has selected a lesson, and now they are presented with a selection of videos from which the user can select to watch.	
Priority	3	
Preconditions	User is currently in ‘lesson overview’ mode	
Postconditions	Once all video content has been viewed, the VR quiz is unlocked.	
Primary Actor	User	
Secondary Actors	VR Peers System	
Trigger	User enters ‘lesson overview’ mode	
Main Scenario	**Step**	**Action**
	1	User selects a video
	2	The systems enters the 3D video scene
	3	The user watches the content of the video
	4	The user returns to the lesson display

FR-5: The system shall allow users to complete lesson assessments in the VR environment.

Number	5	
Name	Complete Lesson Assessment	
Summary	Allow a user to complete a lesson assessment (quiz)	
Priority	4	
Preconditions	User must have completed a video lesson material and be in the lesson overview mode	
Postconditions	Lesson marked complete and progress is reflected in the curriculum map	
Primary Actor	User	
Secondary Actors	Curriculum Database	
Trigger	User clicks “Take the quiz” button	
Main Scenario	**Step**	**Action**
	1	Display VR immersive video
	2	Video pauses at the end
	3	Simple binary choice quiz is displayed for the user
	4	User selects answer
	5	Correct response video is displayed
	6	Assessment ends and user is returned to the lesson overview mode
Extensions	**Step**	**Branching Action**
	5a	If incorrect response is chosen, incorrect response video is displayed
	5b	User is returned to step 3

FR-6: The system shall offer a simple VR tutorial and safety warnings to new users.

Number	6	
Name	Display tutorial and safety warnings	
Summary	The system displays a simple tutorial and safety warnings to new users after logging in for the first time.	
Priority	5	
Preconditions	New user must be logged in	
Postconditions	User is brought to the curriculum map	
Primary Actor	User	
Secondary Actors	Account Database	
Trigger	New user successfully logs in	
Main Scenario	**Step**	**Action**
	1	The system displays safety warnings for the user to read before proceeding
	2	User clicks the “Continue” button
	3	The system displays a tutorial for the user to watch before proceeding
	4	User clicks the “Continue” button

FR-7: The system shall allow standard media controls for the content videos.

Number	7	
Name	Media controls	
Summary	When the system displays content videos, the media controls should follow standard conventions	
Priority	2	
Preconditions	User has selected a video	
Postconditions	User is able to interact with content videos	
Primary Actor	User	
Secondary Actors	Media content	
Trigger	Content video is played	
Main Scenario	**Step**	**Action**
	1	User clicks the video to bring up the media controls
	2	User interacts with the video using standard media controls
	3	The corresponding media control response is applied to the video
	4	The video ends
Extensions	**Step**	**Branching Action**
	1a	The user does not click the video: - Nothing happens and the video plays through uninterrupted.

2.1 Acceptance Testing

The following acceptance tests are created with the intention of thoroughly testing all of the functional requirements listed above. Each acceptance test will test one or multiple of the functional requirements, ensuring that every aspect of the program will function as expected.

The acceptance tests will be written in the following format:
Acceptance Test #ID (list of all functional requirements that will be tested):
Overview
[In depth description]

Acceptance Test 1 (1, 6, 7):
The user logs into the application using their PEERs account as a new user.
This test will ensure that users are able to successfully log into an account while also ensuring that new users will be shown safety warnings and tutorial videos upon logging in. This will also test if the standard media controls are functioning for the tutorial videos.

Acceptance Test 2 (2):
The user opens the curriculum map.
This test will ensure that the curriculum map functions and displays properly.

Acceptance Test 3 (1, 2, 6):
The user logs into the application as an existing user.
This test will ensure that users are able to successfully log into an account that has tracked their progress from previous sessions. This will also ensure that the safety warnings and tutorial videos are only displayed for new users and not existing videos. Aside from this, logging in as an existing user and checking the curriculum map will confirm whether the map has tracked the user's previous progress or not.

Acceptance Test 4 (2, 3):
The user chooses a lesson.
This test will ensure that the user is able to move between the curriculum map and the lesson overview mode easily by selecting a lesson from the curriculum map. This will also ensure that the lesson overview mode displays as expected.

Acceptance Test 5 (3, 4, 5, 7):

The user completes a lesson.

This test will ensure that the user is able to enter a new lesson and complete it while testing all of the features associated with completing a lesson. First, it will ensure that the role-play videos display properly. This test will also ensure that content videos can be interacted with using standard media controls. As for the assessment portion, this test will ensure that the assessment environment functions as expected, and that the user is able to answer the assessment questions as expected.

Acceptance Test 6 (5):

The user intentionally selects the wrong answer in a role-play scenario.

This test will ensure that the program proceeds the way it's intended to and plays the correct corresponding videos when the incorrect answer is selected.

Acceptance Test 7 (2, 3, 5):

The user exits the lesson overview to return to the curriculum map after completing an assessment.

This test will make sure the progress from completing the assessment is properly saved, and that the assessment ends when it's supposed to. This will also test to ensure that the lesson overview mode still displays properly after the assessment, and that the user is able to return to the updated curriculum map from the lesson overview mode.

3. Non-Functional Requirements

The non-functional requirements (**NFR**) are meant to display requirements that are related to the characteristics of the system. NFRs describe constraints and capabilities of the system (performance requirements, safety and privacy requirements, security requirements, and software quality attributes).

Non-Functional Requirements follow the format: NFR-unique identification number: requirement description (Priority number). The priority number 1 represents the lowest priority and 5 the highest priority.

3.1 Performance Requirements

NFR-1: The system shall display the lesson to the VR headset within 0.5 seconds, 99% of the time. (Priority 5)

NFR-2: The system shall accurately keep track of the user's curriculum progress 99% of the time. (Priority 3)

NFR-3: The system shall be able to handle up to 100 active users 95% of the time. (Priority 4)

NFR-4: The system shall be able to handle 100 login requests every 5 seconds 95% of the time. (Priority 3)

NFR-5: The system shall be able to handle the user input within 0.5 seconds 95% of the time. (Priority 4)

NFR-6: The system shall be able to update based on user actions within 0.5 seconds 95% of the time. (Priority 4)

3.2 Safety and Privacy Requirements

NFR-7: The system shall comply with the California Consumer Privacy Act (California) (Priority 4)

NFR-8: The system shall comply with the General Data Protection Regulation (European Union) (Priority 4)

NFR-9: The system shall protect user's data from unauthorized access. (Priority 4)

3.3 Security Requirements

NFR-10: The system shall be able to protect user data from attacks 99% of the time. (Priority 4)

3.4 Software Quality Attributes

NFR-11: The system shall be able to connect to internet services 95% of the time. (Priority 3)

NFR-12: The system shall be available to all users 24 hours a day, 95% of the time. (Priority 3)

NFR-13: The system shall not exceed a size of 20GB. (Priority 1)

4. User Interface

See "User Interface Design Document (**UIDD**) for Virtual Video Modeling on the Social Skills of Adults with Autism" for user interface designs. Estimated availability on December 4th, 2023.

5. Deliverables

The deliverables for this project will be maintained on our central GitHub repository. Physical copies of important documents will be provided in person. In the section below, all deliverables are separated by the semester in which they will be first published. Almost all deliverables will be subject to updates and modifications over the entire year. Our GitHub repository will contain the most current versions of all deliverables; additional physical copies will be provided as requested.

Deliverables for Fall 2023 - [COS397]

Hard copies of each of the following:

- Systems Requirement Specification
- System Design Document
- User Interface Design Document

An electronic file containing the following:

- Systems Requirement Specification **(available: Nov 1, 2023)**
- System Design Document **(available: Nov 15, 2023)**
- User Interface Design Document **(available: Nov 29, 2023)**

Deliverables for Spring 2024 - [COS497]

Hard copies of each of the following:

- User Manual
- Administrator Manual

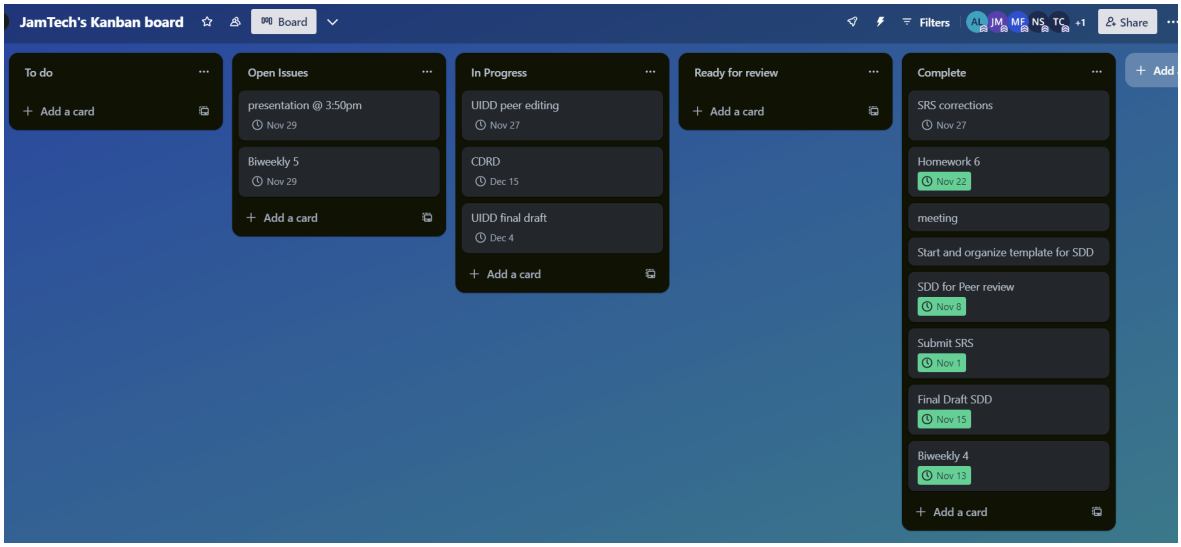
An electronic file containing the following:
** Documents hosted on Github will be updated regularly **

- Critical Design Review Document **(Estimated availability: to be announced)**
- Code Inspection Report **(Estimated availability: to be announced)**
- Final Project Report **(Estimated availability: to be announced)**
- User Manual **(Estimated availability: to be announced)**
- Administrator Manual **(Estimated availability: to be announced)**
- All source code **(Estimated availability: to be announced)**
- The executable program **(Estimated availability: to be announced)**
- Any other software required for installation and execution of the delivered program. **(Estimated availability: to be announced)**

6. Open Issues

This section will cover any issues that are currently open that have not reached a conclusion, a detailed description of the issue will be available. The issues will be addressed by their target completion date.

Open issues will be tracked in our kanban board:
https://trello.com/invite/b/erGa4KSJ/ATTI3a00e58a77e188a54783a2636c467fb372962312/jamtechs-kanban-board



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

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.

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.

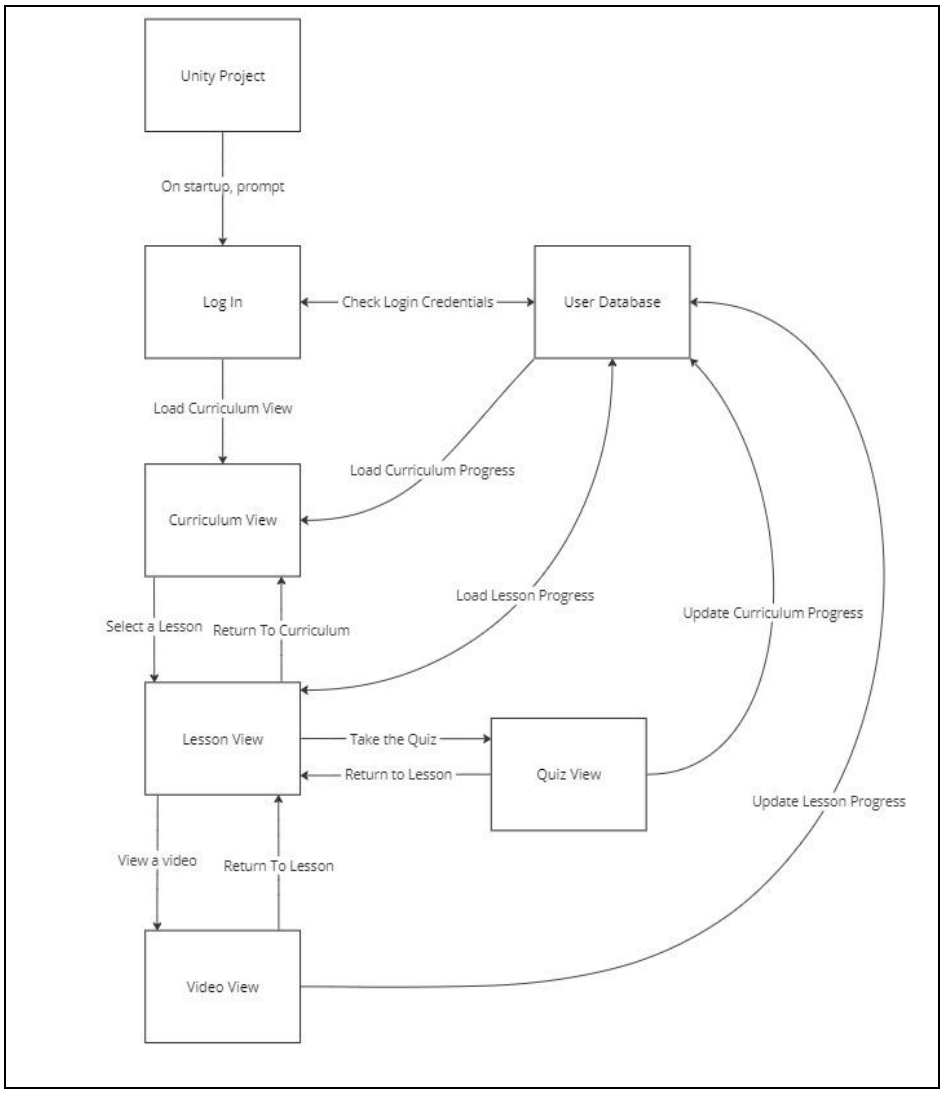


FIG C1.0 Software Architecture/Flow Diagram*: This graph showcases how the software parts interact with each other with the individual elements represented by boxes and arrows indicating interaction actions connecting everything.*

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.

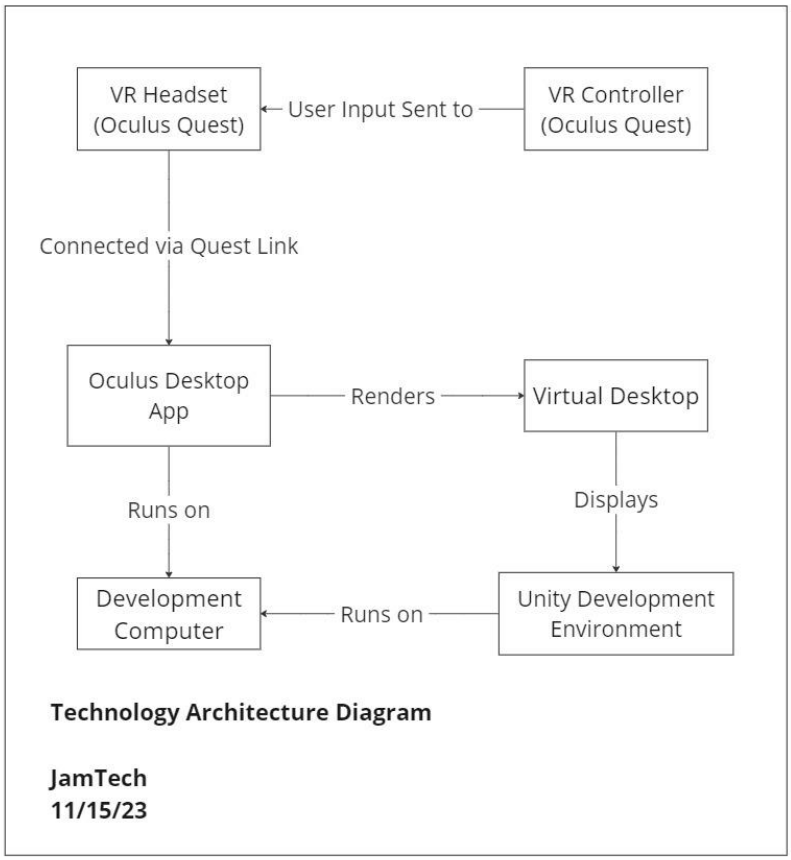


FIG C2.0 Technology Architecture Diagram*: This diagram shows the Oculus Quest connecting to our development computer via Quest Link. Once connected, we can view the Unity Development Environment in the VR headset and interact with Unity via the controllers.*

Fig C2.0 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.

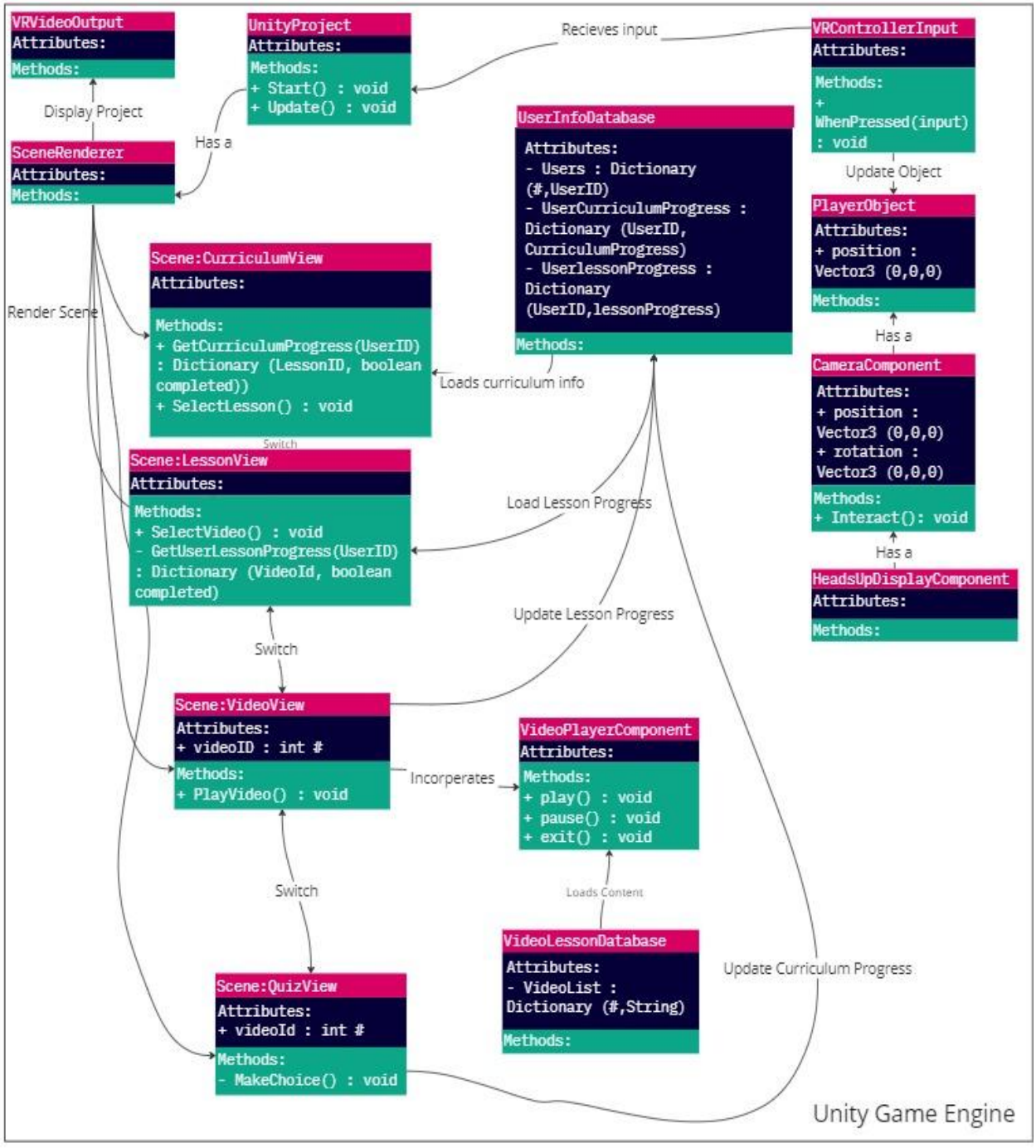


FIG C3.0 Unity Game Engine*: This diagram depicts the system's architectural design, represented by core components, their attributes, and their methods. The boxes contain this information and are connected by arrows with action words to describe how each component interacts with the system and the other 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 C4.0 is a diagram of the User Info Database, which will store the user's information like username, name, password, curriculum progress, ect. Figure C5.0 is a diagram of the Lesson Content Database that includes lesson content like videos and quizzes.

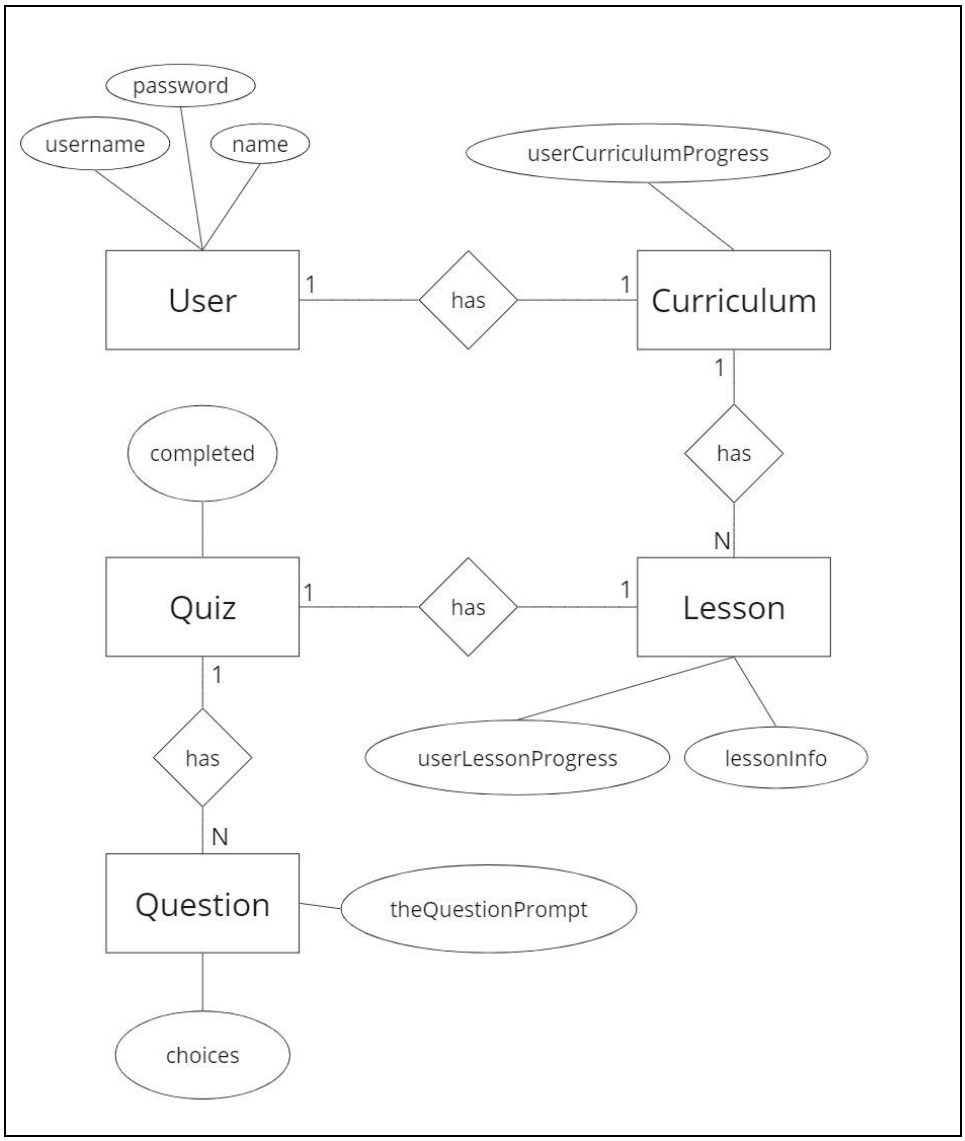


FIG C4.0 User Info Database Diagram*: The diagram above shows the layout of the User Info Database. Objects are represented by boxes, and the data those objects possess are represented by ovals. The lines represent a relationship between two components while the number or 'N' represent the ratio between objects. Object to data ratio is always one to one so a number is not used.*

Table C1: User Info Database Table

Table C1 provides more detail for each element seen in Figure C4.0 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	64 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

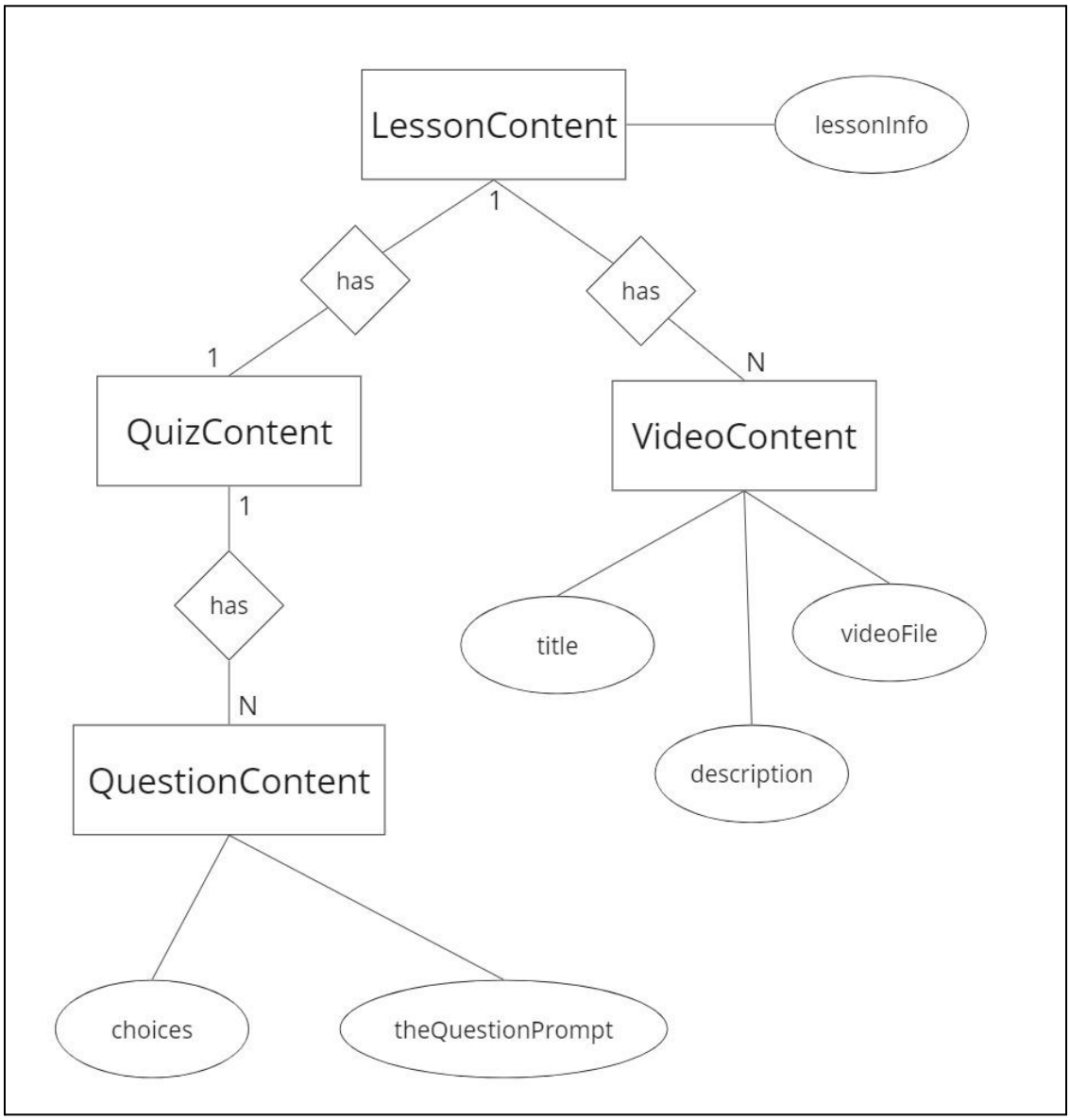


FIG C5.0 Lesson Content Database Diagram: The diagram above shows the layout of the Lesson Content Database.*Objects are represented by boxes, and the data those objects possess are represented by ovals. The lines represent a relationship between two components while the number or 'N' represent the ratio between objects. Object to data ratio is always one to one so a number is not used.*

Table C2: Lesson Content Database Table.

Table C2 provides more detail for each element seen in Figure C5.0 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.

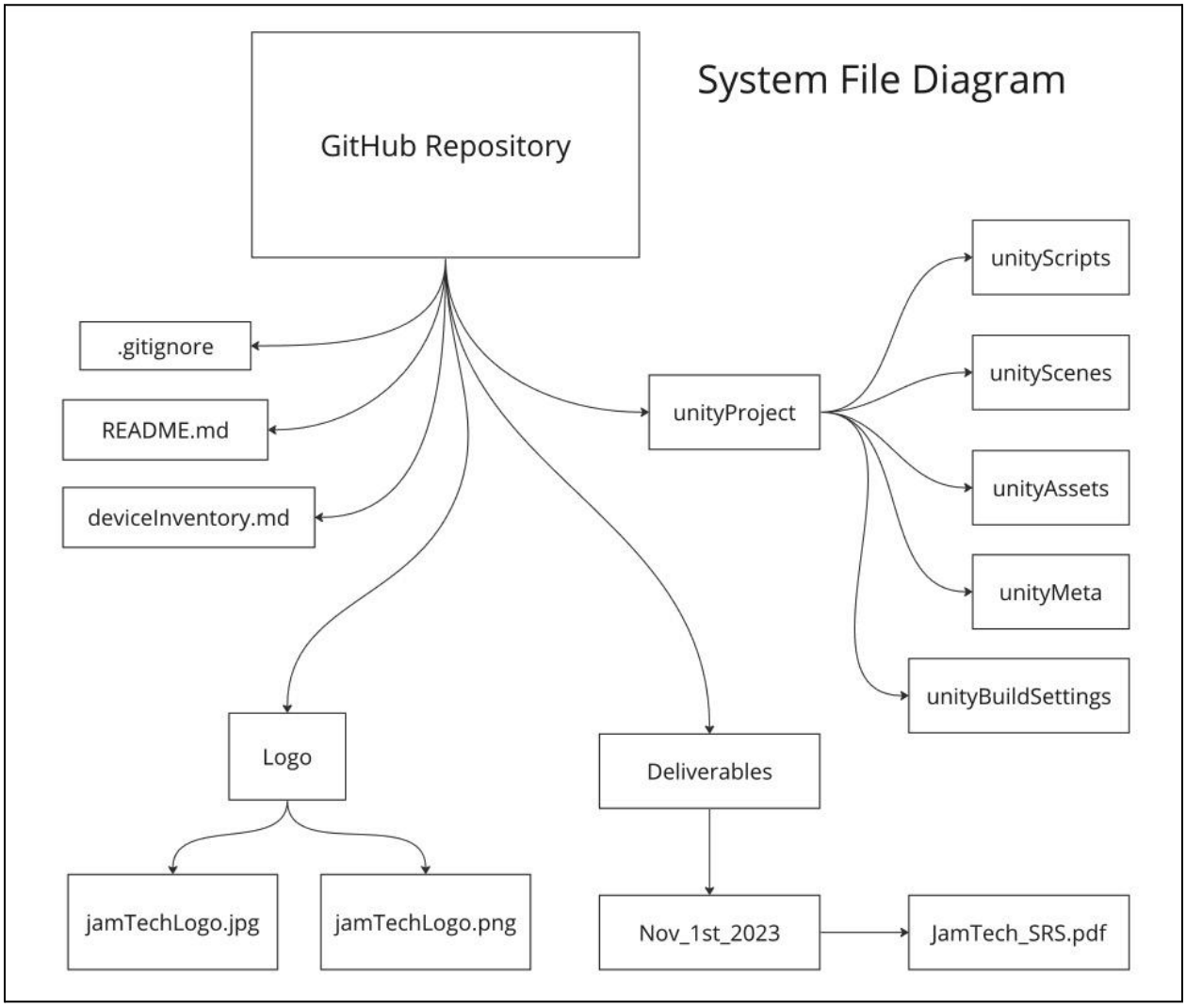


FIG C6.0 System File Diagram: The diagram above shows our system's file structure, which will be available in Github. The diagram shows the various folders and files represented by boxes, with arrows indicating what folders will contain which documents.

Table C3: File System Table

Table C3 provides more detail for each element seen in Figure C6.0 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 C4: Functional Requirements to System Components

Table C4 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

User Interface 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

Dec 4, 2023

Prepared by JamTech:

Tristan Cilley, Allison Lupien, Nick Sarno,

Jacob Michaud, Maha Fazli

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. 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 following description comes from the website for the UCLA PEERS® Clinic,

> "The Program for the Education and Enrichment of Relational Skills (PEERS®) is world-renowned for providing evidence-based social skills treatment to preschoolers, adolescents, and young adults with autism spectrum disorder (ASD), attention deficit/hyperactivity disorder (ADHD), anxiety, depression, and other socio-emotional problems." (PEERS®, 2023)

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 (eg; communication, humor, dating etiquette, etc) 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

This document shall serve as a guide to the application's user interface and visual design. The document will cover the following topics: user interface standards, user interface walkthrough, and data validation. 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.

2 . User Interface Standards

This section focuses on the standards chosen for the design of this application, encompassing logical choices such as layout and design principles, common components like menus and screens, and visual aspects including colors and fonts. These standards will be consistently applied throughout the user interface design of the entire application.

To see the flow from one screen to another, please see ***Figure D1.0*** in the JamTech SDD, ***Software Architecture/Flow Diagram***, which details the different screen areas. For visual aids for each of the screen areas, as well as explanations of each, please see ***Section 3: User Interface Walkthrough***, below.

Logical Choices:

1. Keeping titles and pop up displays centered.

2. The curriculum map is on the home screen because the curriculum map in the original PEERS® app was also the home screen, and because it's the largest navigation menu in the program. The curriculum map will be horizontal because there is more horizontal space in VR.
3. The screens will not be cluttered in order to avoid overstimulating the user, especially because it's easier to overstimulate users with a crowded screen in virtual reality, so we took that into account.

Common Components:

1. The white bar at the top with the title will remain the same, except the title will change as needed.
    a. The map icon will be there if the user is not in the curriculum map to return the user to that screen. This was taken from the original PEERS® app.
4. All of the lesson layouts will be the same
    a. Popup window prompting for more information, video carousel, questions in the quizzes, etc.

Visual Aspects:

1. The default background color for the application will be [Linear gradient (FFFFFF, B03EBA, 5F2065)] and the color for the buttons will be [BF3CAA].
2. Each lesson in the curriculum map will be assigned a unique color.
3. Within the lessons, the background color will vary depending on the color assigned to the lesson being viewed
4. Within the lessons, the color for the buttons will vary depending on the color assigned to the lesson being viewed.
5. The pop-up windows displayed when viewing lessons on the curriculum map will be [Linear gradient ((7D7EF0,2F2F94)]
6. The default text color throughout the program will be white
    a. The text in the login input boxes will be gray for contrast
    b. The text in the screen titles will match the pop-up windows when viewing lessons.
7. We will use the font [San-serif font "Inter"] for all of the text.
8. The map icon to return to the curriculum map in the top left-hand corner of the screen will be shaped like a zig-zag to match the design shape of the curriculum map that shows on the curriculum map screen.
9. The curriculum map will be shaped in a zig-zag to make better use of the space, and also in reference to the curriculum map designed in the original PEERS® app.

2.1 Accessibility and Inclusive Design Elements

Our application design prioritizes accessibility and inclusivity through diverse video content and by addressing the needs of users with visual, hearing, physical, or mental impairments. The following section outlines the features we intend to implement to meet our goals of accessibility and inclusion.

Visual impairment: (dyslexia / color blindness)

1. High contrast colors will be utilized throughout the project, especially with text against a background with color.
2. Color blindness mode
3. Text-to-Speech will be available for textual information such that a screen reader would be able to be used

Hearing impairment:

1. Video will have high quality clear audio
2. The user will be able to adjust the audio up or down for those who are sensitive or hard of hearing
3. An option for subtitles will be available on video content for those hard of hearing

Physical impairment:

1. The user will be able to navigate throughout the application with only one remote for anyone with one hand or limited mobility in their hands
2. The user will be able to use the software while sitting down for users with mobility issues
3. Haptic feedback will be used to confirm a click

Intellectual impairment:

1. The app is designed for people with diverse cognitive abilities.
2. Text-to-speech will be included for users who are non-readers.

Other functionality for different communities:

1. The system will eventually be available in different languages
2. Content will not be gender specific, sexual-identity specific, or race specific
3. Universal icons will be utilized for ease of use and any icon that is unique will be explained in the tutorial.
4. A tutorial will be provided for users who do not know how to use the technology

3 . User Interface Walkthrough

This section of the document serves as a comprehensive introduction to the project's user interface (UI) design. It begins with a general overview of the various UI views incorporated in the project, accompanied by a navigation diagram (***see Figure D1***). Each mock-up screenshot is individually presented with a summary, navigation details, and explanations of buttons, text-fields, or other UI objects.

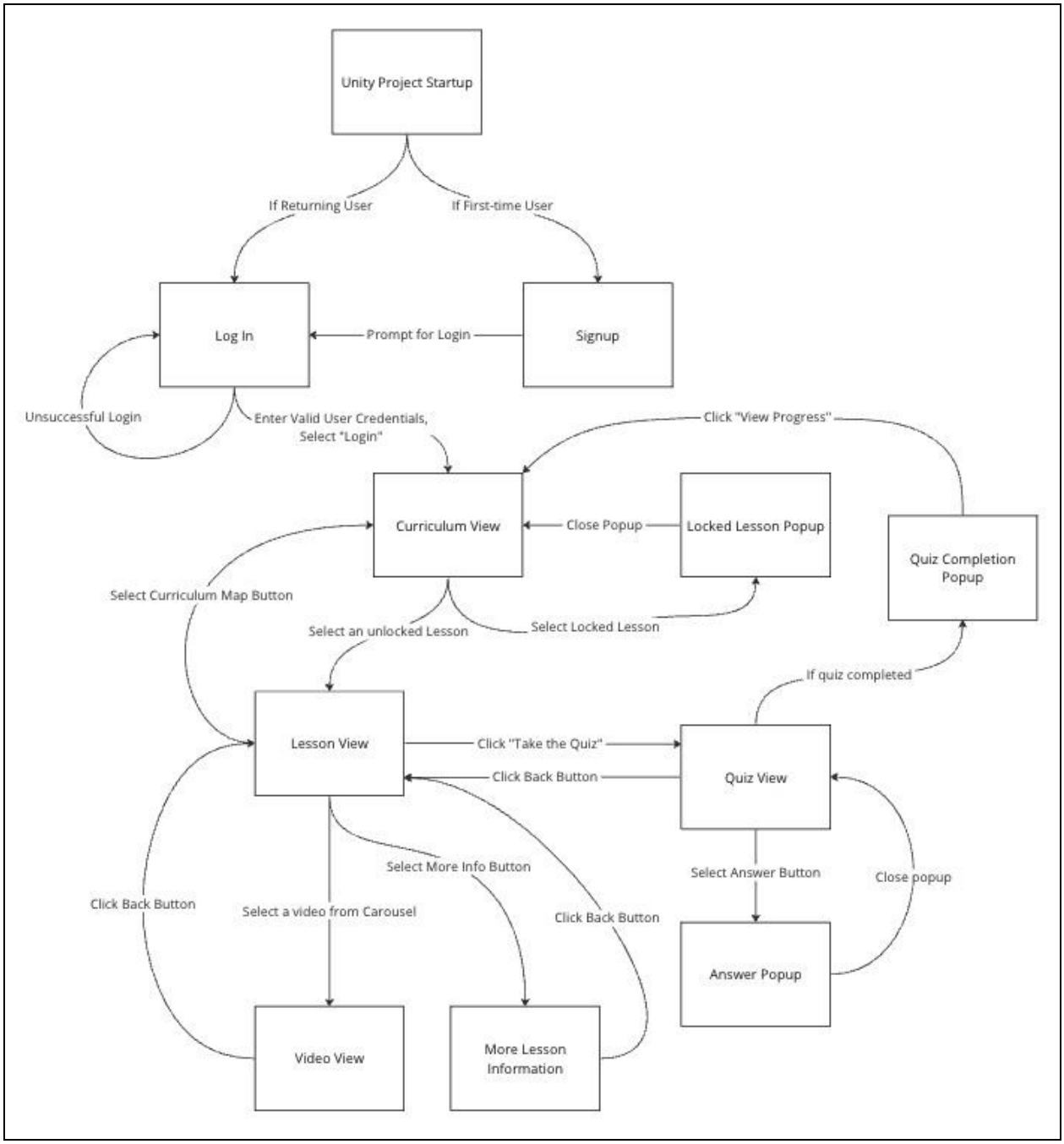


Figure D1 UI Navigation: *A detailed flow diagram depicting how a user is expected to use UI elements to navigate through the project.*

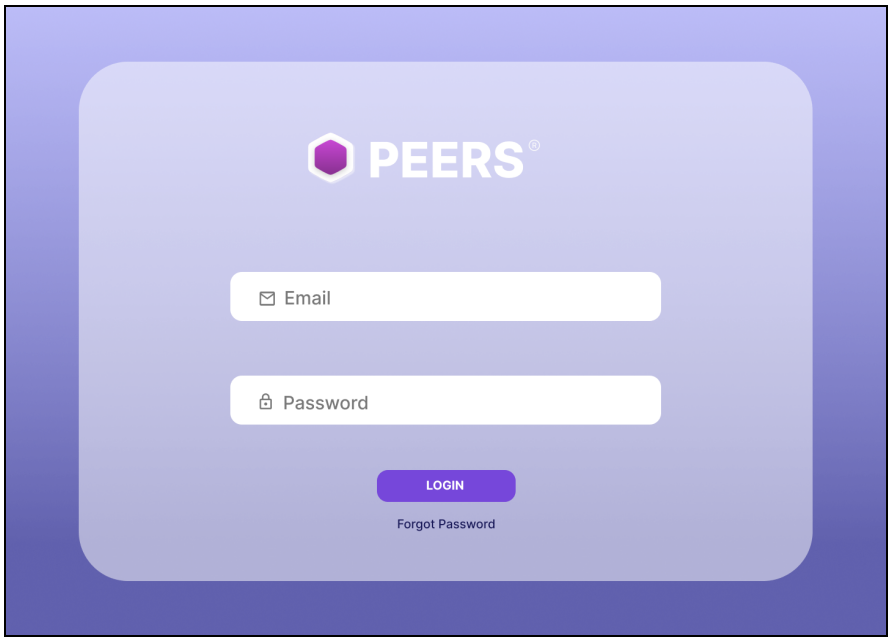


Figure D2 Login Page: *The User opens the application, they view the login page and enter their email and password.*

Summary:

The login page is the first view the user is presented with after opening the application. In order to proceed to the curriculum map, the user must first enter their account information and log in.

Navigation guide:

Once the user enters their account information and logs in by clicking the 'login' button, they will be brought to the tutorials section and then to the curriculum map, alternatively they can go straight to the curriculum map depending on if they are a new or returning user . If the user has forgotten their password, they can click the "forgot password" link and enter their email address associated with their account and instructions on how to reset their password.

Breakdown of components:

The login page contains a distinct box with the PEERS® logo at the top, two input boxes for text and a login button. The two text boxes have 'email' and 'password' written inside to indicate what information they require. There is also a "forgot password" link below the "login" button.

Database:

User login information, including emails and passwords, is stored in the user info database. Emails are stored as strings and passwords are stored as hash values.

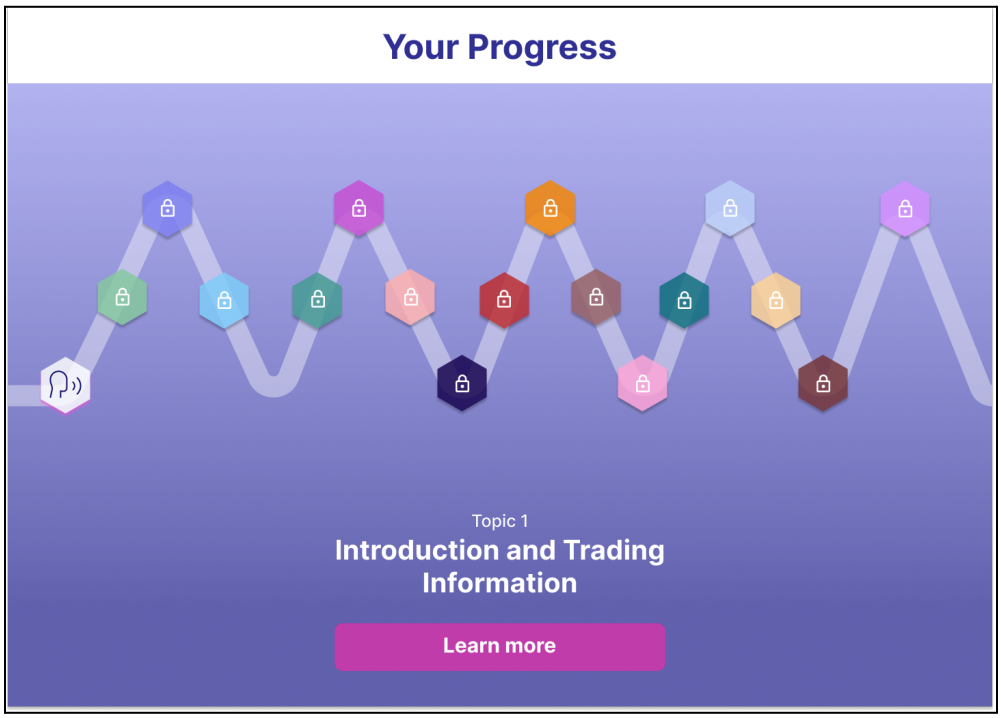


Figure D3 Curriculum map: *The home screen of the application depicting a map in the shape of a zig-zag with the user's current lesson progress.*

Summary:

The curriculum map is an extended head's up display that allows the user to look around themselves in the virtual reality environment to see the map. This will be the home screen of the application, allowing the user to access lessons and information while also telling them how much progress they have made.

Navigation guide:

Since this curriculum map functions as the home page, the user will be able to reach this screen in a multitude of ways. The user may enter their account information on the login page and click the 'login' button to reach this page or they may click the curriculum-map-shaped icon in the top left-hand corner displayed in any other screen to return. By clicking on any of the lessons visible within the screen, the user will bring up a pop-up window for that lesson, personalized to whether or not it has been unlocked yet. The user can choose which lesson they would like to interact with by clicking it, and then by clicking on the 'Learn More' button at the bottom of the screen, the user can access the lesson display page for that particular lesson.

Breakdown of components:

This screen includes a zig-zag shaped trail going from left to right with several multi-colored hexagons at the peaks, troughs, and middles of every zig or zag to represent lessons. Each hexagon has either a lock symbol in the middle to indicate the lesson is still locked or a personalized icon to represent the content of the lesson in a visual. Below the map, the topic number and lesson title of the last lesson interacted with by the user is displayed. If the lesson is unlocked, there is a 'Learn More' button, and if the lesson is locked, there is a lock symbol and text that reads 'Not yet unlocked' in place of that button.

Database:

The user's progress through the curriculum map is tracked by the data element userCurriculumProgress, which is an integer stored in the User Info Database. The integer represents the lesson number the logged in user is currently on, in other words the most recent unlocked lesson. All lessons before this number are displayed as unlocked and all lessons after are displayed as locked.

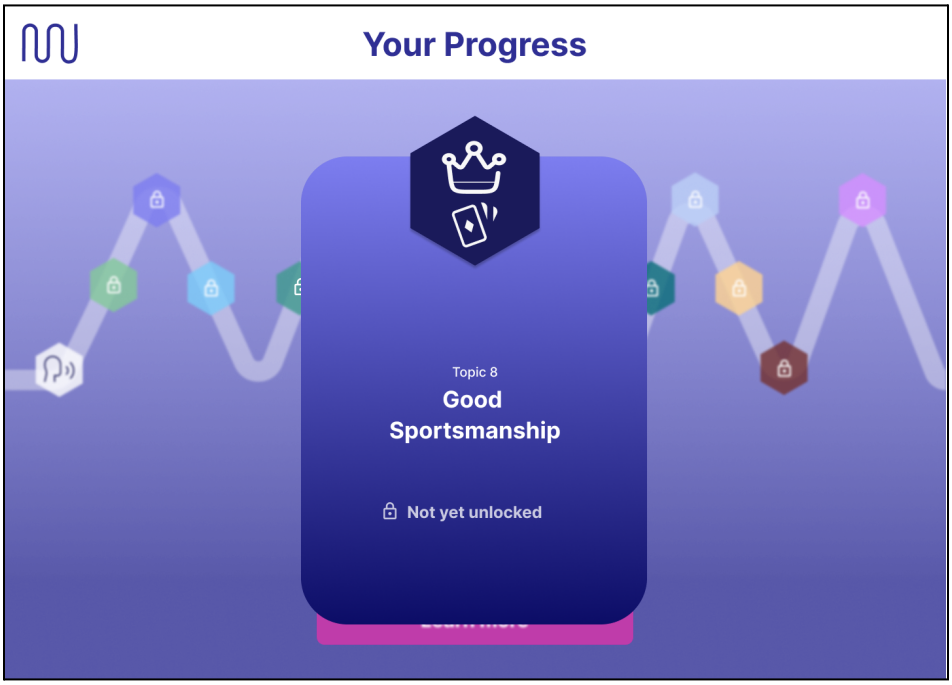


Figure D4 Locked Lesson Selection: *If the user clicks on a locked lesson this screen will pop up notifying the user the name and number of the topic along with a message that shows the lesson has not been unlocked.*

Summary:

From the curriculum map, the user can click on any lesson to be brought to the current lesson display. If this lesson is still locked the display will indicate that the lesson is 'not

yet unlocked'. The user cannot enter the lesson view when it is locked, but can return to the curriculum map.

Navigation guide:

To navigate to a locked lesson the user can point the cursor at a locked lesson on the curriculum map and click. Once the locked lesson is displayed the user can return to the curriculum map by using the cursor to click on the curriculum map icon that looks like a curled zig-zag in the upper left hand corner.

Breakdown of components:

In the locked lesson display, the icon, topic number and title of the lesson are displayed as well as a lock icon and text to indicate that the lesson is locked. The user cannot continue to the lesson view. The only navigation options available from this view is to return to the curriculum map by clicking on the curriculum map icon in the upper left corner.

Database:

This UI view does not involve a query to the database. The locked/unlocked status of a lesson is established by querying the User Info Database for the userCurriculumProgress data element. This happens each time the curriculum map is rendered. The type of pop-up, locked or unlocked, depends on the current state of the curriculum map.

(UI Walkthrough continues on following page)

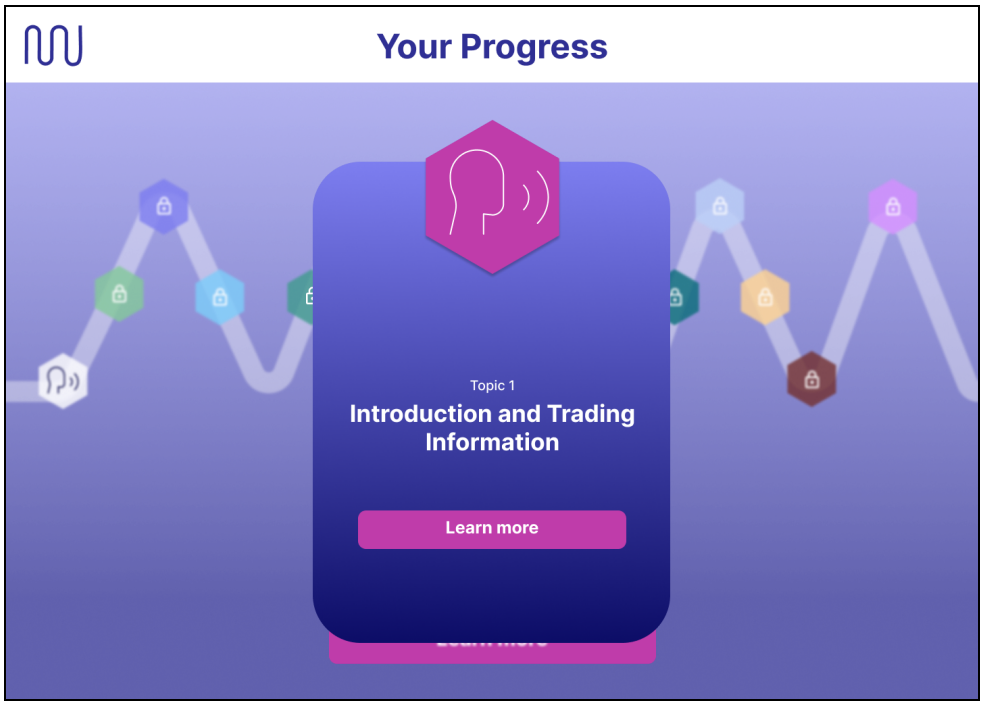


Figure D5 Current Lesson: *If the user clicks on the current lesson they are on, this screen will pop up displaying the lesson they have clicked on followed by a learn more button.*

Summary:

From the curriculum map, the user can click on any lesson to be brought to the current lesson display. If this lesson is unlocked the display will include a button reading 'Learn more' that when clicked allows the user to access the lesson content.

Navigation guide:

To navigate to an unlocked lesson display the user can point the cursor at an unlocked lesson on the curriculum map and click. Once the unlocked current lesson is displayed, the user can return to the curriculum map by using the cursor to click on the map icon in the upper left hand corner, or continue to that lesson's lesson display view by using the cursor to click on the Learn more button.

Breakdown of components:

In the unlocked lesson display, the icon, topic number, and title of the lesson are displayed as well as a button that reads 'Learn more'. The user can navigate to the lesson view via the learn more button. Additionally, the user can navigate back to the curriculum map by clicking on the curriculum map icon in the upper left corner.

Database:

This UI view does not involve a query to the database. The locked/unlocked status of a lesson is established by querying the User Info Database for the userCurriculumProgress data element. This happens each time the curriculum map is rendered. The type of pop-up, locked or unlocked, depends on the current state of the curriculum map.

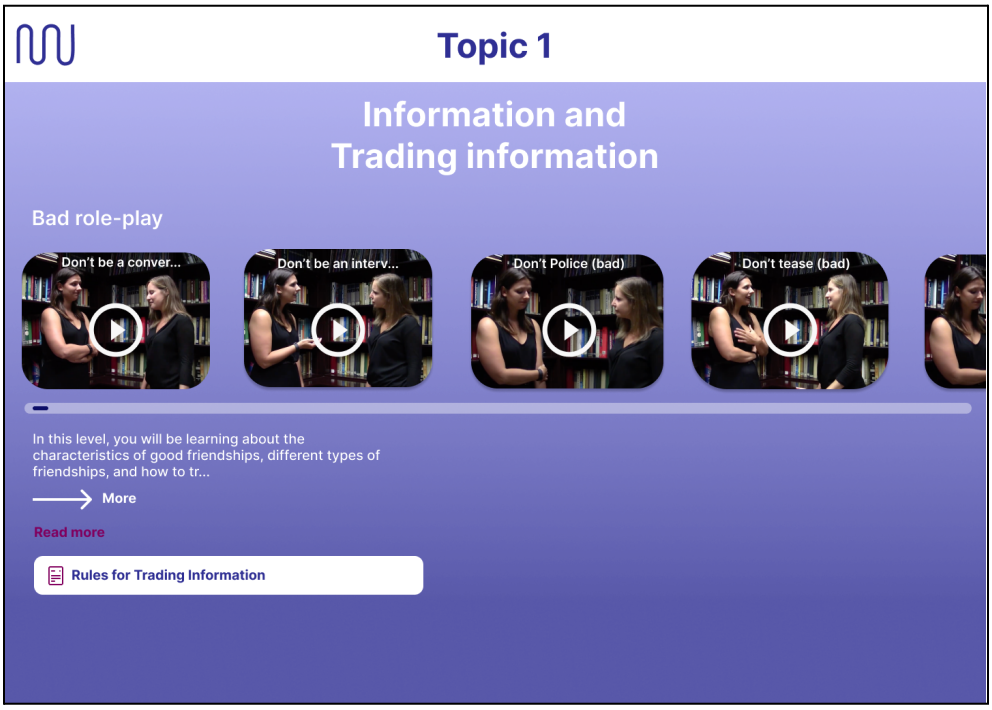


Figure D6 Lesson Display: *The lesson displays the content of the specific lesson including videos and a general description.*

Summary:

When a user clicks on a 'Learn more' button for a lesson, they are directed to the lesson display that originally shows bad-role play videos, a general description that can be expanded, and a button to see more content rules. The user can look down to see more lesson content.

Navigation guide:

The user can navigate to a lesson display by using the cursor to click on a 'Learn More' button from either a current lesson view or from the current lesson displayed on the curriculum map. The user can navigate back to the curriculum map by using the cursor to click on the curriculum map icon in the upper left corner. The user can look down to see more content like more videos and the quiz.

Breakdown of components:

On a lesson display the user is shown the title bad-role play is followed by a series of videos that showcase different examples of bad-role play in the following topic, followed by a description of what will be addressed in this lesson. The word “more” when clicked provides an extended description of the lesson. The read more section has a rules button that when clicked provides a more detailed rules and explanation of the topic. Lastly, the user can look down with the headset to see more lesson content and navigation options.

Database:

The title, labels, text information, and video thumbnails displayed in this view are all stored in the Lesson Content Database in a LessonContent object. The text information is stored in the data element lessonInfo which is formatted text. The thumbnail images for each video are contained in the data object, VideoContent. All lesson content is queried from the database each time the user enters the lesson view.

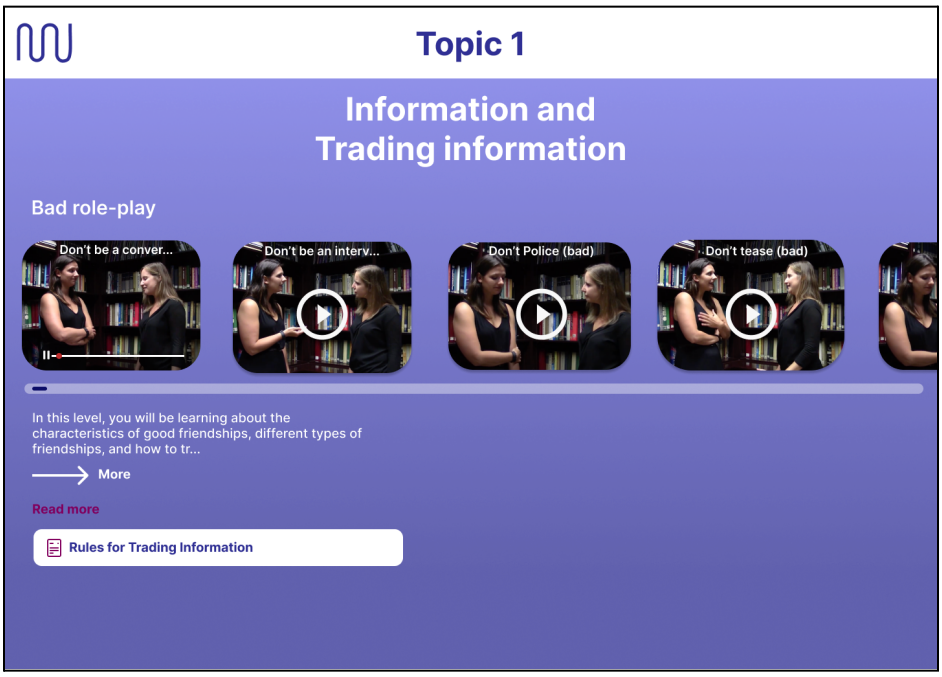


Figure D7 **Video Being Played**: *The user can click on a video in the lesson content to play the video.*

Summary:

The user has a variety of videos that they can choose to watch being displayed on the lesson view. Once the user clicks the video they want to watch, the YouTube video begins to play. The user then is provided with multiple options to enlarge the video, play/pause, and adjust the volume. The user exits back to the lesson view after the video is finished.

Navigation guide:

To navigate this view, the user must first put their cursor on the video they would like to watch, and then click the rear trigger button on the VR controller to play the video. They then have the option to pause the video by putting their cursor on the pause button in the bottom left and clicking the rear trigger button. The user can also place their cursor on the video slider, click with the rear trigger, and drag to skip to a specific time in the video. The user may also navigate their cursor to the volume slider and click the rear trigger button and drag to the desired volume.

Breakdown of components:

The video view has a pause/play button, a volume slider, a video slider, and a fullscreen button. The title of the video is displayed at the top in white text if the user is not in fullscreen mode.

Database:

When a thumbnail is clicked, the associated video file is queried from the VideoContent object in the Lesson Content Database.

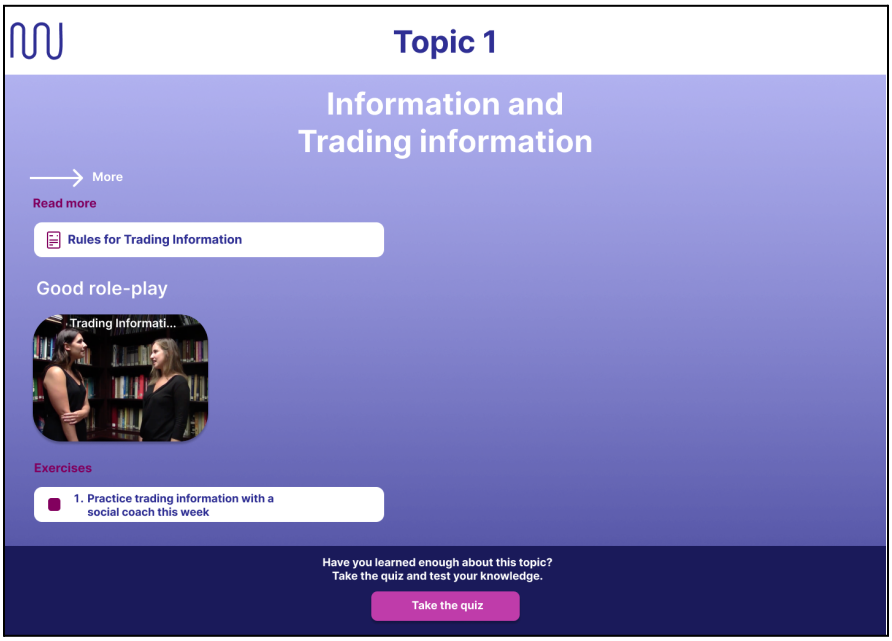


Figure D8 Additional Content: *If the user looks down they will be able to see the rest of the lesson view and lesson content including the good role-play videos, exercise and button to start quiz.*

Summary:

The exercises portion of the lesson view has multiple activities that the user can do in real life in order to improve their skills related to the current lesson. The exercises tab contains check boxes that can be filled in once the user completes the corresponding task. Once all exercises for a lesson have been completed, the “Take the Quiz” button lights up and is able to be clicked on.

Navigation guide:

In order to navigate this section of the app, the user must place their cursor on the exercises tab and click the rear trigger button, this will bring up all of the exercises for the lesson. Once in this view, the user can move their cursor over any checkbox they have completed, and again click the rear trigger button to fill in the check box. Once the user is done checking the desired boxes, they can move their cursor to the back button that is in the top left corner of the exercises tab. Once all exercises are completed the user can move their cursor to the “Take the Quiz" button and click the rear trigger button to enter the quiz view.

Breakdown of components:

There are varying buttons for the check boxes, and purple text fields to the right of them for the corresponding amount of exercises a given lesson has. There is a white“Take the Quiz” text field within a pink button located at the bottom of the page. Before completing all exercises the “Take the Quiz” text field and associated button are a darker more transparent color, and are non interactable.

Database:

The data used to create this view is already available from the first query to the Lesson Content Database which occurred when the user first navigated to the lesson view. (See Figure 6)

(UI Walkthrough continues on following page)

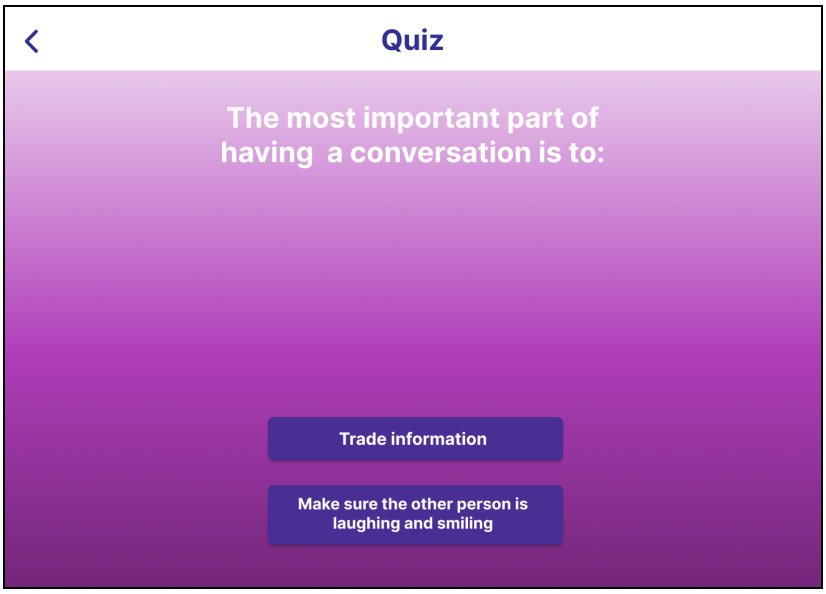


Figure D9 Quiz: *When the user clicks on take quiz they will be taken to the quiz view and they are displayed with a prompt and two choices. If the user has decided they are not ready to take this quiz the arrow on the top left will take them to the lesson view.*

Summary:

The quiz view displays a question in white text at the top of the screen, and two purple boxes that have answers in white text. The boxes can be clicked on to answer the question, and a resulting prompt will appear displaying either “correct answer” or “incorrect answer”. There is also a back arrow to navigate back outside of the quiz view.

Navigation guide:

To navigate this view, the user must aim their cursor at the button that contains the answer they would like to choose, and click down the rear trigger on the VR controller. Once the user has done this for both questions a prompt will appear for them to return to the curriculum view, which they will again press the rear trigger button to activate. If the user does not want to take the test, they may navigate their cursor to the back arrow in the top left corner of the view and press the rear trigger button to return to the lesson view.

Breakdown of components:

There are two buttons on the bottom of the view that can be clicked to answer the question, and one button in the top left that can be clicked to return the user to the lesson view. There is one text field at the top of the screen reading “Quiz”, one text field below that reading the question, and two different text fields within the two answer buttons that display each potential answer.

Database:

The quiz prompts, answers, and results are all stored in a QuizContent object in the Lesson Content Database. Upon successful completion, the userCurriculumProgress integer is incremented in the User Info Database to reflect the user's progress.

4 . Data Validation

The following offers a detailed breakdown of each data item that can be entered into the project. Table D1 provides specific information about data items, including data types, size limits, boundary cases, and allowable formats. Notably, this project requires very few data items from the user, aside from those necessary for login and authentication.

Table D1: Data Validation

Data item	Data type	Limits	Allowable format(s)
username	string	64 bytes	Must be an email Cannot exceed 64 characters
password	string/hash	128 bits / 16 bytes	Must be at least 8 characters long Must include - At least 1 number - At least 1 uppercase letter - At least one special character Cannot exceed 16 characters
name	string	35 bytes	Cannot exceed 35 characters Only include letters