



Co-funded by the
Erasmus+ Programme
of the European Union



Erasmus+

D2.2

First prototype

2020-1-CY01-KA226-VET-082750
Remote Class System



Authors	Virtual Solutions
Abstract	This document focuses on the First Prototype application.



THE CONSORTIUM

No.	Partner Name	Logo
1	GeoImaging Geoimaging ltd	
2	AUTH Aristotle University of Thessaloniki	
3	VS Virtual Solutions	
4	SVR SchooVR	

DISCLAIMER

The information in this document is subject to change without notice. Company or product names mentioned in this document may be trademarks or registered trademarks of their respective companies.

All rights reserved

The document is proprietary of the **Classy** consortium members. No copying, distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights.

This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein.



Table of Contents

Table of Contents	3
Introduction	4
Prototype	4
Frontend	4
Home page	4
Registration page	6
Experience pages before and after login	6
Profile page	7
My experiences page	7
Experience Editor	8
Scene settings tab	9
Models tab	10
My Models tab	11
Hotspots tab	11
Save Experience Button	12
My Sessions page	12
Live Session page	13
Teacher's version	13
Student's version	13
Backend	16
Implemented Subsystems	16
Future subsystems	17



Introduction

Classy is about a digital classroom in which the user can attend classes, have interational meetings, seminars, play games or even get virtual tours in a 3D environment.

This document describes what the first prototype delivers

Prototype

A Prototype is a piece of application delivering initial functionalities based on a concept. It is the basis of the development process that follows and its creation allows the evaluation of the concept and the project by the stakeholders.

Frontend

The frontend application that was created is exactly the same as the mock-ups (Deliverable 2.1) , since for the creation of the mockups we created an actual frontend application that was using dummy data to render the pages.

The frontend of the first prototype includes the following pages:

Home page

The home page is the main entrance to our application a page where the visitor gets an idea about the project and how he / she can use it. For the needs of the prototype this page didn't require any functioning content yet. This page will be one of the last to be filled with content , since we need to finalize the development process and our pilots in order to present the platform users with the full Classy functionality (as this is a product of all the steps in the project).

Currently the Home page is split onto the following sections :

- ◆ Top section (a simple button to allow the students to quickly go to join page where they enter the pin shared by a teacher)

Classy Dev 3

Lorem ipsum dolor sit amet.

Join Class



◆ Section 2 (a section to show basic information about what classy is)

CLASSY

CLASSY is a platform that simulates a class with students and an instructor via 3D virtual environment, in which everyone will be able to interact with each other, a video-wall for the instructor to make presentations, pose quizzes, share screen etc. The "classroom" is the starting place from where the operator-teacher will be able to transfer the class to virtual tours, with the option of VR glasses support, depending on the subject

◆ Section 3 (more information about the project and its goals)



The goal of this project is to bring remote 3D guided lessons into the education. We will develop a platform that offers a novel, technically advanced and appealing e-learning method, offering better learning potentials by watching a virtual model rather than an image in a book or a conventional image of video in a screen. Our solution will offer numerous features appropriate for lessons, like group quizzes in gamified form, personal repository for videos, images etc as well as the option of attending custom-made classes with tours using VR glasses.

◆ Footer section (showing dummy data for now, but will display real info on the final version of the amount of users, experiences and sessions in the platform)

8235 Users 92751 Sessions 80 Experiences

Login page

The login page is the page where users either login to their accounts , or they register in the application

Log In

[» Register](#)

Email/Username
myemail@email.com

Password
.....

Stay Logged In

Log In



Based on the concept this page includes a fully working form that authenticates and logs in users against our system's backend. This form also allows for the remember me function that when checked, it remembers the user in the current browser and auto-logs him/her in the app on next visit.

Registration page

Register

✕ Register

Username

Name

Email
myemail@email.com

Password
.....

Retype Password

Register

Similar to the previous page, this one allows for the registration of new users (teachers). The users created via this form get the proper authorization to use the platform. The form includes a retype password field to make sure the user enters the correct password (one they will be able to remember)

Experience pages before and after login

Both of these pages are left blank for the purpose of the prototype. At a later stage these pages will include a set of pre-made experiences created by the development team based on the feedback and concept of the platform as it evolves.



For the non-logged-in users (visitors) this page will allow a quick preview of the experiences, in order for them to get an understanding of the application without having to register.

For the logged in users this page will allow preview as previously, but also the ability to create a copy of the premade experience to their list of experiences (my experiences) which they can fine tune through our editor, or directly create a session out of.

Profile page

Profile Settings

The Profile page is the page where the registered users can edit their personal information and their passwords as well as editing their system avatar.

The information entered here 'sticks' with the registered user during his usage of the system. The Avatar creation and manipulation was implemented through our WebGL implementation. For the first prototype stage the avatars were kept as simple as possible but more settings will be added along the way. These avatars represent exactly the way these users will look like in our virtual worlds.

My experiences page

My Experiences

Experience Photo	Experience Title	Create Session	Delete	Edit
	Dfvdfvef	→	🗑️	✎️

Records per page: 5 ▾ 1-1 of 1

Create new Experience

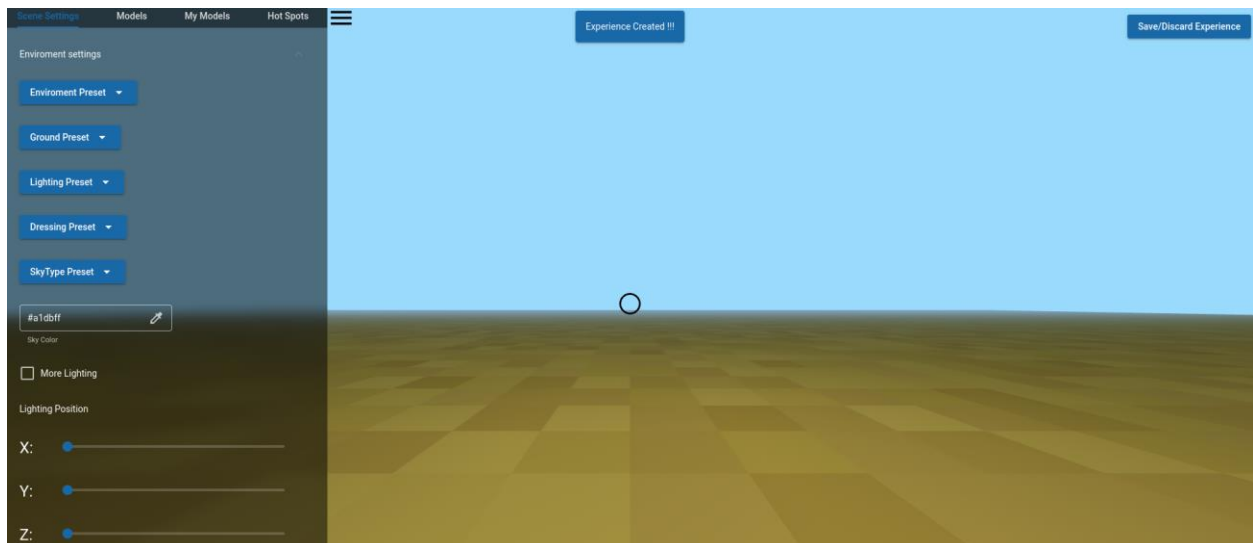


This page holds a list of the experiences the user has created in the system (at a later stage will also include the copied pre-made experiences as mentioned earlier) through our editor. From here the user has the option to edit an existing experience (or delete it) and create new ones.

After the user selects to create a new experience, he/she is presented with a form to enter the basic information (Title, Description, Image)

After entering the information he/she is taken to the Experience Editor

Experience Editor



The experience editor is one of the 2 most important parts of our first prototype (the other is the Live Session page). This Subsystem is a powerful editor that allows for the easy creation of




virtual worlds in 3D.

In this page the user is presented with a left-side drawer with 4 basic tabs :

Scene settings tab

Environment settings

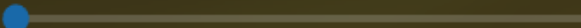
- Environment Preset ▾
- Ground Preset ▾
- Lighting Preset ▾
- Dressing Preset ▾
- SkyType Preset ▾

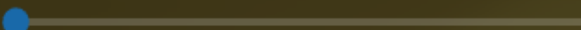
#a1dbff 

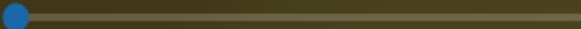
Sky Color

More Lighting

Lighting Position

X: 

Y: 

Z: 



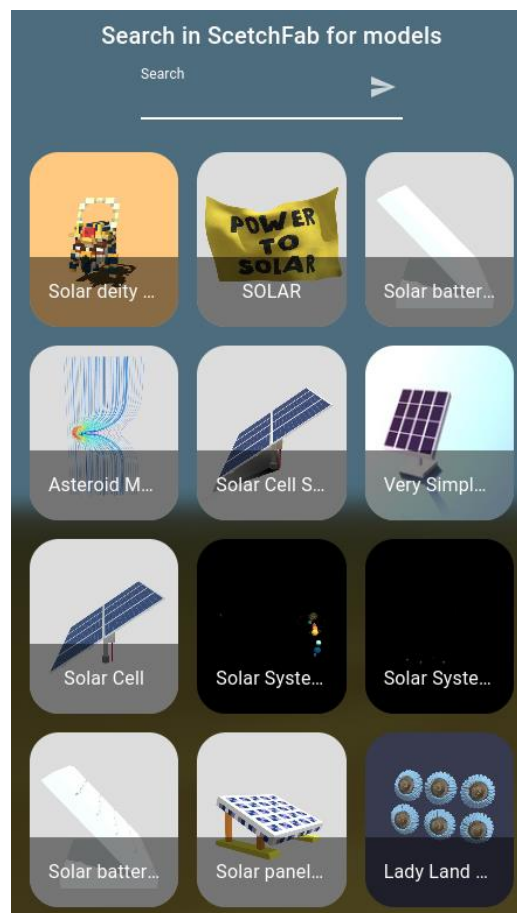
The Scene settings tab holds User interface to control the environment settings as well as the scene lighting. The environment settings can be changed based on presets. We have created presets for:

- Environment
- Ground
- Lighting
- Dressing
- Type of sky

By changing these selections the 3d worlds gets updated in real time and the user can create unique-looking environments.

Extra controls were added for the lighting of the Scene as well for the Sky color.

Models tab



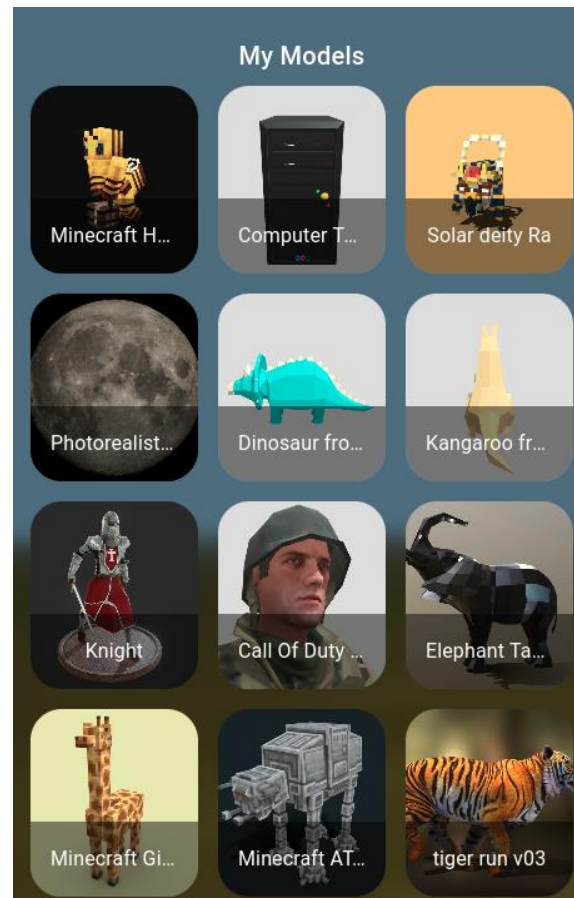
The models tab allows for the users to connect to Sketchfab and import free models to the scene , as well as import their own models (if they have them in sketchfab). This feature requires that the users register an account on Sketchfab.

By selecting a model, the users are presented with the model basic information such as the number of faces and vertices , the license of the model and a link to view more info directly in Sketchfab. They also get an import button that once pressed , brings the model inside the 3d space.



We created a sub system in the overall application to be able to handle importing of the models. When a model gets imported, we use the Sketchfab's API to fetch the actual model and save it onto our server. We do that in order to make the application faster and more responsive and to be able to minify the models (at a later development stage) through an algorithm called Draco compression, which will minimize the size of these models to make them faster to load on all devices.

My Models tab



This tab holds all the models the current user has used in the past as part of his library. This allows him to not have to refetch models from sketchfab , but serve them directly from our system.

Hotspots tab

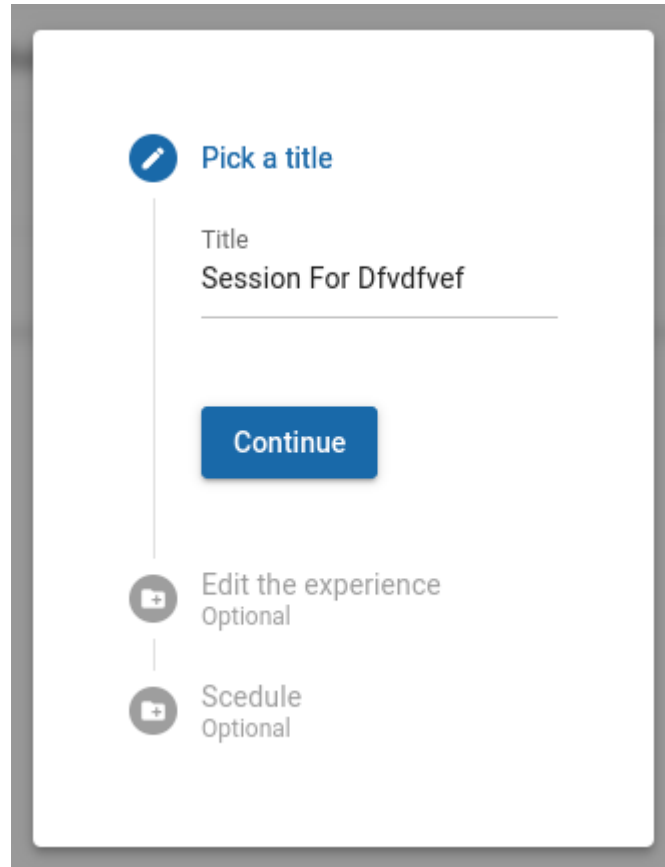
This is kept blank for this phase, but for the next phases it will include the functionality to add teacher resources to the environment.



Save Experience Button

This button allows the user to save / update the experience based on the changes he/she did through the editor. For newly created experiences a new entry is created in the My experiences list.

After the My experiences list gets populated by using our editor, the user is able to create sessions for any experience.



My Sessions page

My sessions

Experience Photo	Session Code	Session Title	Experience Based	Date/Time	Start	Delete	Edit
		Session For Test	test	2022-07-28 10:07	→		
Records per page: 5 ▾ 1-1 of 1							



The sessions the user has created appear in a list through which the user is able to either edit/delete sessions, copy the session code (which they share with their students) and finally they can Launch the session.

Live Session page

The Live Session page is the heart of our system where the real time events happen. It is comprised by a teachers and a students' version

Teacher's version



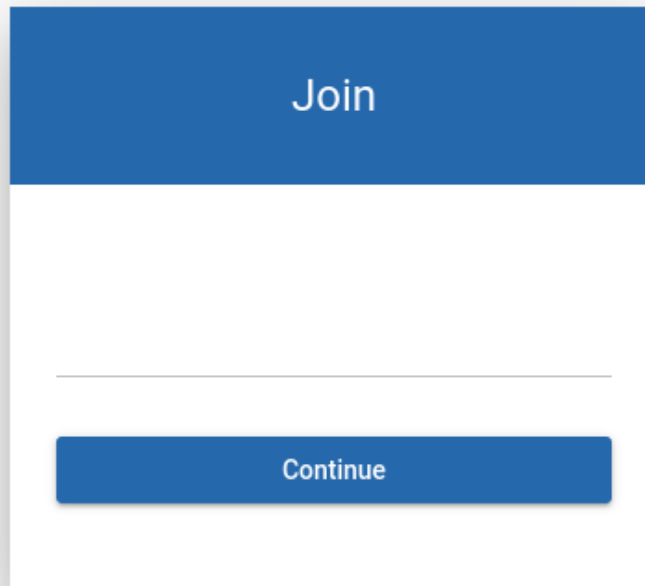
After selecting the microphone that he/she wants to use (we have added the ability to switch through microphones present at your hardware setup) the teacher flies into the virtual environment and is presented with a basic (for now) User interface that allows him/her to view the participants of the session (once a student joins the list gets auto populated), to mute student's microphones and to kick users out of the session.

At a later stage this toolset will include the chatting subsystem as well as on-the-fly actions for the teacher (share a resource, add a quiz etc).

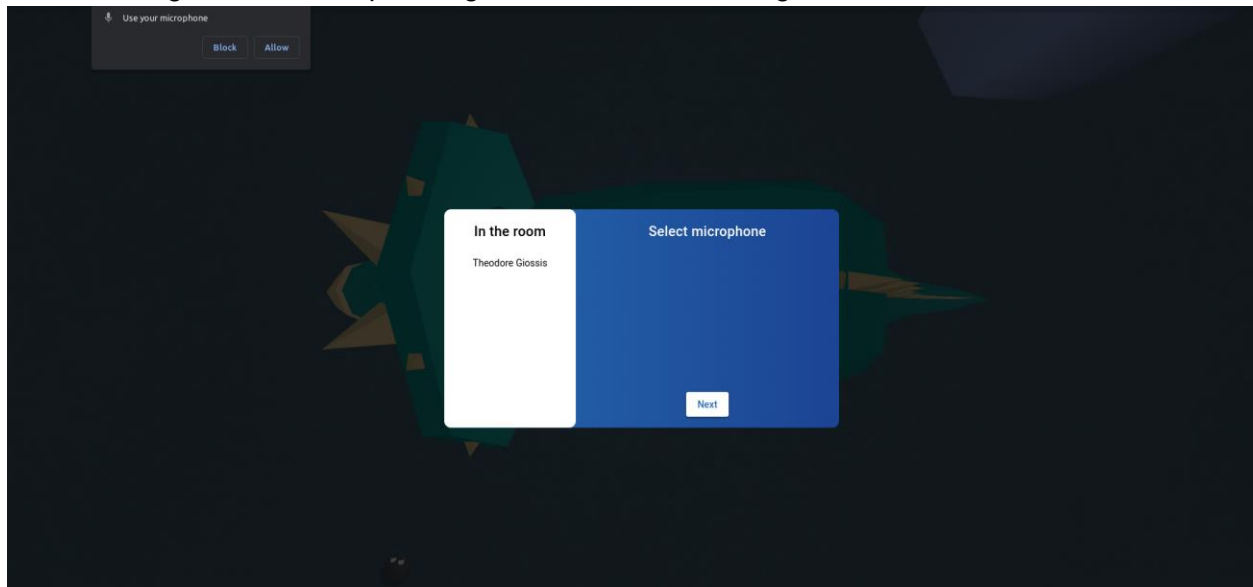
The teacher is able to walk inside the environment by using FPS (First Person Shooter) Game mechanics (Keyboard/ mouse navigation) The teacher's microphone is transmitted to all participants through our SFU (Selective Forward Unit) Subsystem.

Student's version

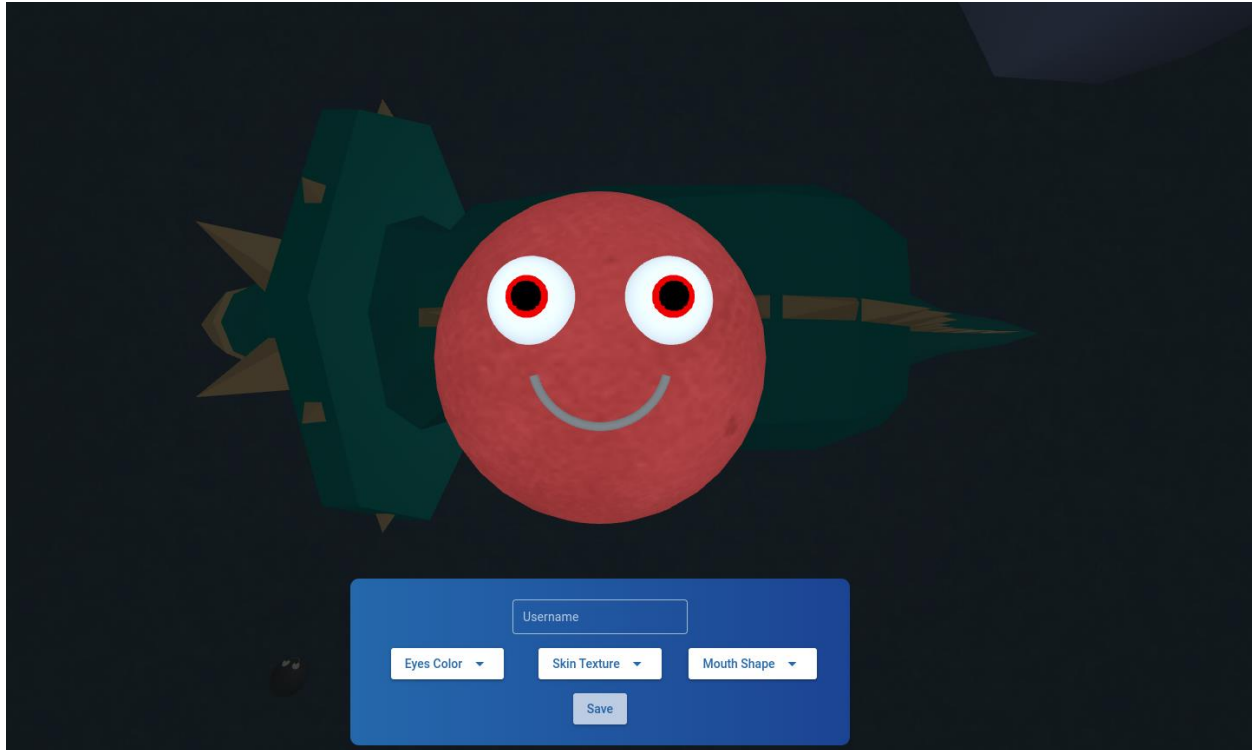
To enter the session , the student first needs to go onto the join page where he/she , simply needs to enter the session code (shared by the teacher on an earlier stage)



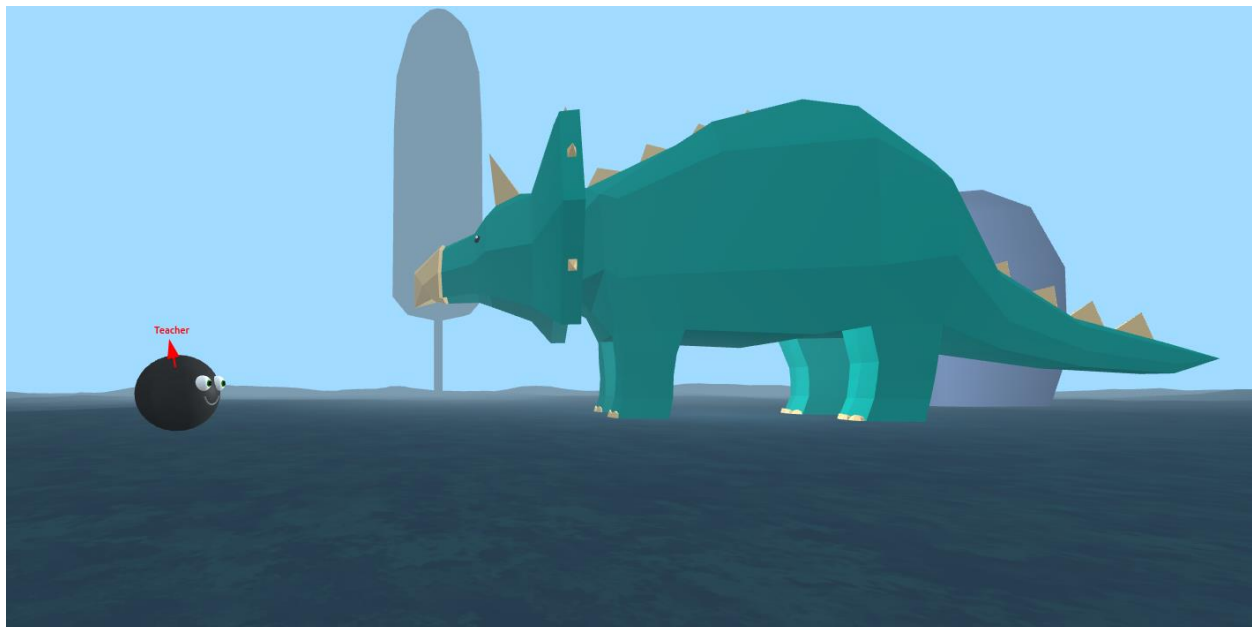
After entering the code and pressing continue the students gets taken to the session.



There he/she needs to allow through the browser the access to their microphones (different look depending on browser used). We present the user with a list of users currently in the room (in this case only the Teacher) and the ability to choose available microphone. After selecting and pressing next we show the user the avatar manipulation subsystem.



Here the user enters their name and customizes the avatar. After saving the student flies into the environment where he/she can interact with all the participants.



At a later stage we will include a waiting room functionality that will allow the teacher to manually include participants onto the session. In the above image you can see the student's view after entering the session, where the Teacher's avatar is always synchronized with the current teachers position in the 3d space. The student (same as the teacher) can move around



the scene and explore the environment. Back to the teacher's view we can see the student moving around and of course hear them through our speakers.



Backend

For the first prototype a big amount of work needed to be done on the Backend of our application.

Based on the work done on “User requirements analysis and project ethical issues” (IO1) of our project’s implementation we proceeded to implement the architecture.

Our backend during the first prototype stage includes the following subsystems all running inside an Ubuntu Linux powered VPS server instance (please refer to the Concept design Document for more details):

Implemented Subsystems

1. **MariaDB Database.**

We implemented this SQL DB to hold all the information for our application

2. **Headless CMS (Strapi)**

We implemented a headless Content Management System based on Node.js to be able to structure our DB properly, to handle our controllers, to enable Authentication and Authorization strategy, to clarify roles for the users, and to generate our REST API endpoints that we consume on the frontend.

3. **Websocket Subsystem**

To be able to handle the real-time interactions required for the first prototype we created a subsystem based on Websockets technology. This system is currently responsible for



exchanging information and state between the users through the use of Rooms. So each session is essentially a Room namespace inside our subsystem and once a user enters it (through the Authentication mechanism) they get access to a flow of information (such as each user's position in 3d space, avatar and name data , etc.) that allows us to render onto their browsers our 3d environments.

A special mention is needed to the Position exchange websocket subsystem , since it required a big amount of time to make sure the positioning of the avatars in 3d is accurate and that the movement of the avatars is as smooth as possible to get a good user experience.

4. Audio conferencing

We installed and setted up an SFU (Selective Forward Unit) subsystem that would allow us to handle audio exchanging inside our Sessions. The system was created with the help of an open source project called Jitsi.

5. Turn Relay subsystem

For the Audio conferencing system to work as intended, it was necessary to create a subsystem that would allow the connectivity and audio broadcasting even through extreme firewall protected systems. Our Turn server (based on Coturn) is able to Relay the traffic in all network environments , acting as a middleware between the client and our server's SFU

Future subsystems

Based on our concept design and the reports we got through our pilot testings and questionnaires we will proceed to implement the following subsystems:

1. Chatting subsystem

A subsystem to allow chat functionality during a session with the ability to exchange documents.

2. Conversion utilities subsystem

A system that will be able to convert uploaded documents by the teachers onto a format readable in our applications. We will use these converted documents to render content onto the 3d environment.

3. Quiz Creation and handling Subsystem

A system to allow the teachers to create quizzes of various types , make use of them in the 3d sessions , and the necessary actions to store the results onto the DB.

4. Emailing Subsystem

A system that will allow us to send out emails through the platform (inviting students, sending reports to admins and teachers, reminders for upcoming sessions etc.)