MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

REST WEB APPLICATION FOR GENERATING 3D MODELS USING AUGMENTED REALITY

DIPLOMA PROJECT

Author: **Cristian Alexandru RUSU**

Coordinator: **S.l dr. ing. Dan Ioan GOȚA**

2021

MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Authorized,

DEAN
Prof.dr.ing. Liviu MICLEA

AUTOMATION DEPARTAMENT DIRECTOR
Prof.dr.ing. Honoriu VĂLEAN

Author: **Cristian Alexandru RUSU**

REST Web application for generating 3D models using Augmented Reality

1. **Project proposal:** *Posibilitatea de a imbunatati sistemul educational prin intermediul unei aplicatii web, cu ajutorul realitatii augumentate.*

2. **Project contents:** *Abstract, Introduction, Bibliographic Study, Analysis, Design and Implementation, Conclusion, Bibliography.*

3. **Place of documentation:** *Tehnical University of Cluj-Napoca*

4. **Consultants: -**

5. **Date of issue of the proposal:**

6. **Date of delivery:** 08.07.2021

Author signature _____

Coordinator signature _____

MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Declaraţie pe proprie răspundere privind

autenticitatea proiectului de diplomă

Subsemnatul(a) **Cristian Alexandru RUSU**,
legitimat(ă) cu CI seria XD nr. 003046, CNP 1981211204497,

autorul lucrării:

REST Web application for generating 3D models using Augmented Reality

elaborată în vederea susţinerii examenului de finalizare a studiilor de licenţă la **Facultatea de Automatică şi Calculatoare**, specializarea **Automatică şi Informatică Aplicată (în limba engleză),** din cadrul Universităţii Tehnice din Cluj-Napoca, sesiunea Iulie 2021 a anului universitar 2020-2021, declar pe proprie răspundere, că această lucrare este rezultatul propriei activităţi intelectuale, pe baza cercetărilor mele şi pe baza informaţiilor obţinute din surse care au fost citate, în textul lucrării, şi în bibliografie.

Declar, că această lucrare nu conţine porţiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislaţiei române şi a convenţiilor internaţionale privind drepturile de autor.

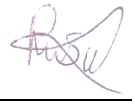
Declar, de asemenea, că această lucrare nu a mai fost prezentată în faţa unei alte comisii de examen de licenţă.

În cazul constatării ulterioare a unor declaraţii false, voi suporta sancţiunile administrative, respectiv, *anularea examenului de licenţă*.

Data

07/07/2021

Cristian Alexandru RUSU



(semnătura)

MINISTERUL EDUCAȚIEI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

ABSTRACT

Of diploma project entitled:

REST Web application for generating 3D models using Augmented Reality

Author: **Cristian Alexandru RUSU**

Coordinator: **S.l dr. ing. Dan Ioan GOȚA**

1. Project requirments: Developing a web application using HTML, CSS and JavaScript programming knowledge and a Cloud Firestore database.

2. Solutions: For the front-end part of the application I used: WebStorm, React 17.0.2 as framework, HTML5, CSS3, JavaScript. For the back-end I used: Cloud Firestore and Google Sheets.

3. Obtained results: With the help of frameworks and development environments, we created an application that helps teachers manage students and students have a more interactive learning environment.

4. Testing and validation: Following the tests and verifications of the application, the requirements of the theme were observed. At the same time, I noticed that the application can be developed and improved.

5. Personal contribution: Designing and implementing a web application and a database, but also integrating augmented reality to provide students with a more modern learning environment by studying their needs.

6. Sources of documentation: Online guides, profile books and specialized sites.

Author signature: _____

Coordinator signature: _____

Table of contents

1 INTRODUCTION 3

1.1 General Context 3
1.2 Objectives and specifications 4

2 BIBLIOGRAPHIC STUDY 5

2.1 The history of browsers 5
2.2 Browser wars 6
2.3 HTML and CSS 6
2.4 JavaScript 7
2.5 React 7
2.6 Cloud Storage 8
2.7 Firebase 8
2.8 Google Sheets 9
2.9 Node.js 9
2.10 Augmented Reality 10

3 ANALYSIS, DESIGN AND IMPLEMENTATION 11

3.1 Analysis and Theoretical Fundamentals 11
3.1.1 How are the websites created 11
3.1.2 System Arhitecture 12
3.1.3 Progressive Web Application (PWA) 13
3.1.4 REST API 14
3.1.5 HTTP Requests 15
3.1.6 HTML and CSS 16
3.1.7 JavaScript 16
3.1.8 React 16
3.1.9 React Lifecycle Hooks 18
3.1.10 Firebase 19
3.1.11 Tabletop.js 19
3.1.12 Augmented Reality 20
3.1.12.1 The use of Augmented Reality 20
3.1.12.2 Augmented Reality in education 21
3.1.12.3 Augmented Reality vs Virtual Reality 22
3.1.12.4 A-Frame 23
3.1.12.5 A-entity 23
3.2 Detailed Design 24
3.2.1 Unified Modeling Language (UML) 24
3.2.2 Use Case Diagram 25
3.2.2.1 Use case for an abstract user 25
3.2.2.2 Use case for a student 26
3.2.2.3 Use case for a teacher 27
3.2.3 Sequence Diagram 28
3.2.4 Architecture of Cloud Firestore 31
3.2.5 Google Sheets 33
3.2.6 Tabletop.js 33
3.2.7 Create React Application 35
3.2.8 Test results for PWA 38
3.2.9 Use of A-frame in the application 41

3.3 TESTING AND VALIDATION 45

4 CONCLUSION 51

4.1 CONCLUSION 51
4.2 FUTURE IMPROVEMENTS 51

5 BIBLIOGRAPHY 53

1 Introduction

1.1 General Context

How can we improve the learning methods? That's quite a valid question, considering that the whole educational system had to migrate online due to the current pandemic context we're all going through. This great and unexpected shift has made it very difficult for pupils and teachers alike to adapt. The learning and teaching processes were both hindered by the lack of live interaction.

This has forced the educational system to adjust to the demands created by the pandemic quickly. In other words, a brand new curriculum had to be designed, one that would fit all needs to the best extent possible. However, at the same time, the declining interaction volume between the teachers and students has led to a loss of interest and attention on both sides.

Even though many educational institutions are fighting with these issues daily, the advanced stage in technology can prove to be a significant factor in rectifying the current predicament, not only for present times, but also for the future. The internet and technology can help the pupil obtain all pertinent educational materials through a simple clicking action. Simultaneously, they offer a wide variety of teaching methods, one more interactive than the other, making the workload and information easier to understand and assimilate.

To create such learning spaces, we have first to consider the fact that everyone understands things differently or at an individual pace. What matches ones learning pattern and process will fail to satisfy the other's educational needs. In this case, assuring an inclusive online curriculum for all peers will improve the scholar's results.

A good example and solution for this problem would be the introduction of AR (Augmented Reality) in education. AR allows the user to interact with a computer-generated 3d model or with a virtual environment. Thus, the information can be explained and understood much more quickly by representing it visually.

It can be applied in education for courses such as chemistry, biology or architecture, to gain an easier understanding of complex subjects by looking into the virtual representation of the model. While this is more of an unconventional method, it can prove to be the most efficient.

This paper aims to offer an alternative to conventional learning, where the pupils can see a 3D representation or an animation of the current lecture's topic through AR technology.

We strive to offer the teachers the possibility to upload the lectures in PDF format, which will have a QR code attached to it. This, in turn, will be scanned through a Web application that I created, after which the child will be enabled to visualize the 3D animation about the uploaded document.

By adding an animation along with the source lesson, the level of interest in students can be increased. At the same time, a more detailed perspective on what they should understand is brought to them in a compelling and reasonably new way.

1.2 Objectives and specifications

Education is an essential part of everyone's life. It can be a key to success and a big help in finding opportunities. People benefit from education in a variety of ways. It brightens a person's thoughts and thinking, for example. It assists students in preparing for a job or further study after graduation from university. Education in a particular field helps people think, feel, and act in ways that contribute to their success and increase not only their own happiness but also their community's.

One thing that can be observed in Romanian's educational system is that teachers did not change the way they try to pass the information to pupils in most cases. Since we live in the technology era, multiple methods of delivering information have been discovered. A lot of them can help in the efficiency of the educational system.

According to several studies, visual formats represent around 75 percent of the information processed by the brain. In addition, visual information is better mapped in the minds of pupils. The absorption of knowledge from visual representations is referred to as visual learning.

Visual learning aids students in developing visual thinking, which is an important and efficient way of learning. In this method, the learner associates words, ideas, or concepts with visuals to better grasp and retain knowledge. With the application that I created, I want to make a change in the educational system by using Augmented Reality to a better understanding of school subjects.

Using this web application, every lesson can have a 3D visual representation to make the information easier to assimilate for every student and a short description of it. In the teacher's case, this application aims to help make a more exciting and unique way to present the material to the class. Also, the Attendance part of the web application can show in real-time if everyone has seen the lesson, has been present to it, and the results for quizzes regarding the topic.

2 Bibliographic Study

2.1 The history of browsers

First, I would like to make a quick introduction to what a browser is. A browser is a software application that allows you to access websites and send and receive information on the World Wide Web.

Nowadays, a significant majority of people use them on a daily basis, making life easier and more accessible from everywhere. With the technology that we have, you can find the information you are looking for by taking your phone out of your pocket and searching it on your browser. They work on multiple devices such as phones, tablets, or laptops.

Although we use it so often, not everyone knows how a browser works. It is a server-client type of model, the browser being the client that runs on a device, sending requests to the server. Then, the server comes back with the client's information, showing it on the browser from the device that we are using.

The first time we hear about browsers was in 1989 when Burners-Lee created The World Wide Web and the first browser in 1990, called WorldWideWeb, which was later changed in Nexus. It could display style sheets, send and receive files and let you check the news. [1]

Mosaic was invented by the University of Illinois' National Center for Supercomputing Applications in 1993, coming with the possibility to show images and text on the same page. One year later, a big part of the team that made Mosaic worked and released Netscape Navigator. It was owned by Mozilla, which we know for the Firefox browser.

Then, we have Opera in 1994, developed by Jon Stephenson von Tetzchner and Geir Ivarsoy, being one of the oldest browsers that are still massively used, getting new updates in the present.

The big sensation, Internet Explorer, was started by Thomas Reardon using Mosaic's source code. It was a part of the Jumpstart Kit from Microsoft Plus! for Windows 95. In 2005, it came in a new form as Microsoft Edge becoming the best version of it.

Firefox came in 2002 with version 0.1 because of Mozilla's community members' desire to take privacy to a new level, being an open-source browser.

Steve Jobs announced in 2003 that Apple had its own browser, Safari. The beta version of it came only for Mac OS X as the default browser, making Internet Explorer an alternative.

In 2008, Google made its entrance into the browsers' world being a combination of components from Apple WebKit and Mozilla Firefox.

2.2 Browser wars

At the beginning of the 1990s, Netscape was the only way to access the World Wide Web. Later, with the apparition of Internet Explorer from Microsoft, the browser wars began. Since then, both of the competitors have tried to get as many users as they can.

To make each of them more appealing to the world, they started to add more features such as Netscape, Javascript, and Microsoft, with Cascading Style Sheets (CSS).

The ultimate factor that made Microsoft have 99% of the market by 1999 was that Internet Explorer was free and pre-installed on Windows. Meanwhile, for Netscape, you had to pay, so it became an alternative.

After a couple of years, Netscape made their codebase open source, which took to the apparition of Firefox in 2002. This time, the main target was to make it as suitable as possible for the users instead of focusing on the war with Microsoft.

In our days, Chrome is the market leader for web browsers and Safari for the mobile ones showing that it does not matter who came first but who made it best. [2]

2.3 HTML and CSS

HTML or HyperText Markup Language was created in 1993 by Tim Berners-Lee. The first version of it, HTML 1.0, made it possible to access and read the information on web browsers. It had 18 tags initially, but as time passed by, new versions came with significant improvements and features.

HTML 2.0 came with new tags, bringing the concept of forms and tables in 1995. In this time, browsers started to create specific layers of tags. Because there was a desire to keep rendering tags similar, the World Wide Web Consortium was born having the mission to improve web development.

The following two versions, HTML 3.2 and 4.01, made a big difference, mainly because of the Cascading Style Sheet, making each webpage more alive.

The last version, HTML 5 is by far the best choice for websites, having around 140 tags, such as <article>, <footer> or <header>. These new features and tags allow creating the ideal website in the easiest way possible. [3]

For developers to create unique visual data, they had to develop something to style the elements and pages, so they created Cascade Style Sheet. It was conceived in 1996 by Håkon Wium Lie which was working at CERN. This technology gives a set of rules on how the HTML should look on a website.

From the first version, CSS released in 1996, the user could change the styles and colors for text and background. The second version, CSS2, became a recommendation from the World Wide Web Consortium. It gives the opportunity to users to conceive the page layout.

In 1999, CSS3 was developed, giving developers new features to create even more exciting and interactive websites. Now, it can help create advanced animations, using multiple backgrounds making the styling of HTML easier. [4]

2.4 JavaScript

Besides HTML and CSS, JavaScript is one of the main scripting languages to create a website. Although, on the one hand, the first two give style to the website, on the other hand, JavaScript permits the developer to provide the functionality to it.

Brandan Eich managed to develop Mocha in 1995, a scripting language that we know today by the name of JavaScript. He presented it to ECMA International, which created standard specifications or ECMAScript. [5]

The first four versions of JavaScript appeared between 1997 and 1999, without many improvements from one to another. Finally, after ten years, the next version, ES5, was supported on most browsers, bringing new features such as object or array methods, strict mode, and supported JSON.

The following progress from JavaScript is ES6, brought in 2015, which made coding more efficient and modern. The apparition of 'let' and 'const' arrow functions or the creation of classes simplified the way of writing the code and improved its performance. Since the apparition of ES6 until today, there is a version for each year adding more improvements in helping developers.

2.5 React

ReactJS is an open-source JavaScript framework for quickly and easily developing rich and engaging online applications with minimum code. ReactJS's primary goal is to provide the greatest possible rendering performance. Its strength stems from the fact that it concentrates on individual components. ReactJS allows a developer to split down a web project into smaller components rather than working on the entire program.

In 2011, Facebook's developers began to have difficulties with code maintenance. The team required extra personnel to maintain the Facebook Ads app functioning, and the number of features increased. As a firm, they were slowed by the increasing number of team members and app features. As a result of the numerous cascade changes, their app grew harder to manage over time.

So, Jordan Walke, one of the developers from Facebook, created FaxJS, which will later become React.js. This helped the company manage the Facebook Ads app, and it became open-source when Facebook acquired Instagram in 2012.

In 2014, React became known and started to build a reputation with new features and an extension for Google Chrome called React Developer Tools which is helping developers with their projects. [6]

It also has two valuable and essential features JSX and Virtual Dom. JSX or JavaScript Syntax Extension allows users to add HTML and JavaScript code in the same file. This feature makes the code easier to understand and modify.

Then, the virtual DOM (Document Object Model) represents a copy of the actual DOM. The Virtual DOM is easier to manipulate than the real DOM because it makes the changes in the virtual one, and then it is compared with the original one to update just some of the content of the page.

2.6 Cloud Storage

When we hear about the concept of cloud storage is back in 1963 when the Defense Advanced Research Projects Agency financed the Massachusetts Institute of Technology for the project on Mathematics and Computation to find a way for more people to use the same data simultaneously.

In the late 1990s, many companies began to work with cloud services. First, Saleforce.com gave the opportunity for users to access multiple applications on the internet. Then, Amazon did the same thing but over the name of Elastic Compute Cloud in 2006, which also rented computers. Finally, three years later, Google created Google Sheets or Docs, making people's daily work more accessible.

Cloud storage is a convenient way to securely store or move data. It lets consumers and companies store their files in the cloud and access them from any device on demand.

It allows a client to send and retrieve files from a remote data server through the internet. Clients can always access their data even if one server is offline or misses data because the same data is frequently kept on multiple servers at the same time.

The main concern was the security of those cloud storage. Now, access control, user authentication, and data encryption are all standard security features provided by most cloud storage providers. However, when it comes to private business papers or personnel information, it is exceptionally vital to ensure these safeguards are in place. [7]

2.7 Firebase

Firebase is a software development platform developed by Firebase Inc. in 2011, which three years later was bought by Google. It began as a real-time database and has now grown to include 18 services as well as specialized Application Programming Interfaces.

Everything started from a startup which was created by Andrew Lee and James Tamplin over the name of Envolve. It was allowing users to incorporate an online chat feature on websites. They discovered a little later that people use it in order to send information, so they split the API from the chat feature.

In 2011, Firebase was founded, coming with a product a year later, which can synchronize data throughout web and mobile applications named Firebase Realtime Database. [8]

After that, in 2014, they released another two applications: Firebase Authentication and Firebase Hosting, which are back-end tools to create the application wanted. In the same year, Firebase became the newest acquisition of Google, expanding more in the upcoming years.

Now, it is a full Backend-as-a-Service or Baas, which is free to use. The user gets a lot of tools to develop and expand their application and to obtain profit.

The main features of Firebase are:

- Authentication - Firebase has an email/password authentication system, but it also gives you the option to log in using Facebook, Google, GitHub, or Twitter accounts.
- Database - it allows every user to access information synced at any moment, even if an application is not opened.
- Hosting – it lets the user host web applications quickly.
- File Storage – the platform stores files on Google Cloud Storage right from the client.
- Notifications – your application can send notifications without having to add more code.

2.8 Google Sheets

It all started from 2Web Technologies, which created XL2Web. This application was later acquired by Google, and it was renamed Google Labs Spreadsheets. They decided to try it firstly just for a small number of users as a test but ended up being available for anyone that has a Google Account.

Google Sheets is a spreadsheet software that is included with Google's free Google Docs Editors suite. Google Docs, Google Slides, Google Drawings, Google Forms, Google Sites, and Google Keep are all part of the service. Google Sheets is accessible as an online application, a mobile application, or a desktop program.

The program supports Microsoft Excel file types. People may create and modify files online while collaborating in real-time with other users. The user's edits are logged, and a revision history shows the changes. A particular color and cursor indicate the position of an editor, and a permissions system limits what users may do. [9]

2.9 Node.js

Node.js is an open-source JavaScript environment built on top of the Google v8 engine. It made it possible to write JavaScript code on every platform that it is installed on. It is free to use on any device that runs Windows, Linux, or macOS.

Ryan Dahl wrote the framework in 2009 because Apache HTTP Server, the most popular server at that time, could not handle many connections simultaneously, making the user experience unpleasant.

Then, he presented the project at the Inaugural European JSConf at the end of 2009. At the beginning of the following year, Node.js receives a package manager to help programmers share and publish the Node.js package's source code, and it can install, uninstall or update packages. [10]

The enthusiast developers worldwide started to create packages (libraries) that helped the growth of Node.js. Some of the mentionable packages created are Express.js, helping with the API endpoints, Friend Connect, to keep the connection alive between API and database, and Sales.js, which is a library that helps creating e-commerce fast and easy.

2.10 Augmented Reality

Augmented reality (AR) is a technologically augmented representation of the actual world produced via digital visual components, music, or other sensory inputs. It is a rising trend among businesses that deal with mobile computing and commercial applications.

The first sign of this technology appeared in 1962 when Sensorama was invented. It was a booth that included the five senses using a stereo, screen, fans, changers of smell, and a moving chair. All of these items work in concomitant with the movie that could be seen on the screen to make the viewer feel like in the film's story.

Six years later, Ivan Sutherland created Sketch. It was a part of his thesis at MIT, which made him a pioneer of Augmented Reality. It was a pair of glasses made to let the consumer see 3D images. The device was hanged from the ceiling because it was too heavy, being called "The sword of Damocles".

In 1980, the first wearable device was produced by Steve Man. It was a helmet made to show virtual data to the user named EyeTap. The device is still used, but it came in different and simplest forms, such as a pair of spectacles.

After the term Augmented Reality was coined by Tom Caudell in 1990, many new software and devices started to become popular. In 2000, ARToolKit was designed for developers to create augmented reality software programs by Hirokazu Kato.

Then, Augmented Reality was introduced in advertising. Big companies such as Coca-Cola used it in order to create unique and exciting commercials. They used big screens that were put in crowded places to animate the world and advertise their products.

In 2012, the Google Glasses were released, giving access to this technology at a larger scale. It did not resonate well with the audience, so Google stopped producing them in 2015. Four years later, the same product with extra features came into the market but just for professionals. [11]

3 Analysis, Design and Implementation

3.1 Analysis and Theoretical Fundamentals

3.1.1 How are the websites created

FrontEnd and Backend are two key concepts in website building. To increase the site's operation, the backend and frontend must interact efficiently with one another.

FrontEnd development is concerned with a website that the user interacts with directly, which is also the application's client side. This includes fonts, colors, pictures, tables, buttons, and the navigation menu, as well as anything the user may directly encounter. HTML, CSS, and JavaScript are the languages utilized on this side. [12]

Although all websites are built using HTML and CSS, content management systems, blogging software, and e-commerce platforms frequently incorporate other technologies.

When we visit a website, our browser most likely receives HTML and CSS material from the server that hosts the site. To produce the page we're looking at, the web browser reads the HTML and CSS code.

HTML uses elements to describe the page's structure. HTML tags work in the same way that containers do. They provide insight into the material contained between the start and end tags. Attributes, which are found at the start of a tag and consist of two parts: a name and a value separated by an equal sign, provide extra information about the content of an element.

Thanks to CSS elements, you can create rules that specify what the content of an element should look like, making the web page much more appealing and engaging for the user. CSS associates rules with HTML elements to make it function. A selector and a declaration are the two pieces of a CSS rule. These can be set in a line, or in an external stylesheet, and assigned to the HTML element.

The FrontEnd developer must make sure that the site is responsive, meaning that it looks well on any platform, whether mobile, tablet, or desktop, and on any search engine the user uses.

The backend of a website is the server side, which the user cannot view or interact with. Only a frontend application allows the user to access the backend. Writing APIs, establishing libraries, arranging system components, creating and managing databases, designing software architecture, and internal program design are all part of the backend. It organizes and saves data, while taking care that everything is fine on the client's side.

An important role of the backend is in web servers. Web servers are computer programs that store, process, and deliver web pages to users. When you visit a website, the server on which it is hosted can be anywhere in the world. The browser must first connect to a DNS (Domain Name System) server in order for the user to be able to visit the site and determine the server's location.

A local development environment, called localhost, plays an important role in the web development process. The benefit of using a localhost is that it is only accessible to the developer, allowing him to test and experiment with the code before the site goes live.

Two other fundamental concepts that come up when we talk about websites are UI and UX.

UI, or user interface, is the graphical aspect of an application. It consists of the buttons that users click, the text that they read, the pictures, the text input forms, and any other components that the user interacts with once they have entered the site. The user interface also includes the screen's look, transitions, animations, and other micro-interactions.

UX or user experience is determined by how the user interacts with the web page. The user experience should be cursive and intuitive, the application's navigation should be logical, and the user's contact with the program should give the user the impression that the site's objective has been effectively accomplished. [13]

3.1.2 System Arhitecture

The application that I created has an architecture of client-server type. In architecture, both the client and the server play an important part. The server is responsible for storing data and processing and delivering information and reacting to consumer requests. Likewise, it is the main piece that controls access to the system database in addition to these features.

The presentation is handled by the client via a user-friendly interface. In this situation, we are talking about a sort of client known as a web application designed specifically for teachers and students.

Among the most common client-server architectures is the two-tier architecture due to the two types of components present: client and server. As previously stated, specific duties are shared by both the client and the server.

This job separation brings various benefits to the surface, including simplicity of maintenance and rapid modifications without disrupting other functions.

Because the server will keep the data and have its own authentication mechanism, the user who chooses to work with such an architecture will not have to take on data security. The server must be developed and deployed to accommodate a higher number of simultaneous requests in order to retain the integrity of the database's contents.

The HTTP protocol is used to communicate between the client and the server in this application, which includes particular methods like GET, POST, WELL, and DELETE. The client makes a 'request' to the server, and the server responds with a 'response.' This information is supplied in JSON format to make the content of the apps easier to parse.

Therefore, through the web application used by teachers or students, they can send data to the server through HTTP requests. These requests are received by the server,

which then responds to the client. The server in this example is Firebase, which will save the data in a relational database. [14]

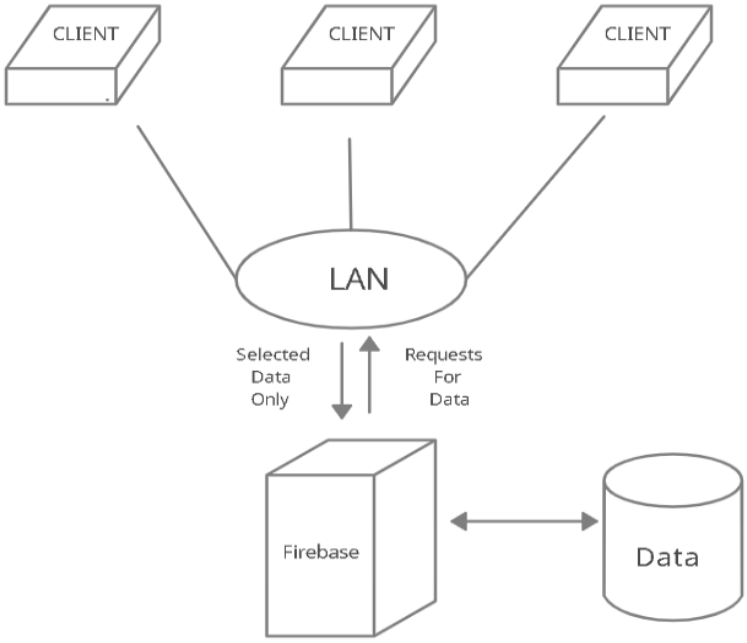


Fig. 3.1. Server-Client Arhitecture

3.1.3 Progressive Web Application (PWA)

Progressive Web apps are a growing trend that has been embraced by major corporations such as Instagram, Twitter, and Google, who all believe that this is the next natural step in the evolution of the web. They were first released in 2015 and have gained popularity, particularly in regions with weak internet connections, because they consume far less data than a standard smartphone application.

These are web-based software applications that are created with popular web technologies like HTML, CSS, and JavaScript. Among the most attractive features of these applications are offline functionality, specific mobile browsing, notifications push, direct access from the home screen, the fact that it is not necessary to download the application, are linkable and shareable, indexed by Google, the connection requirements are small or the fact that it does not require periodic updates.

There are also extra skills on both iOS and Android. PWAs on iOS have access to geolocation, camera, sensors, and Apple Pay. In addition, Android supports data storage of up to 50 MB, integrated Bluetooth access for BLE devices, speech recognition, background synchronization, and web banners to attract other users to use the app.

Progressive Web apps work offline, allowing users to browse them even when they are not connected to the internet.

Because there is no need to construct another distinct application, online stores developed as PWAs save up to 75% of the expenses of a native application, both in terms of development and maintenance.

Google has made it possible to install these apps directly from the Play Store by introducing the Trusted Web Activity feature in Chrome 72. Apple has added support for a new set of technologies underpinning progressive web applications with the release of iOS 11.2.

Using App Shortcuts, we can now render an app without any other Chrome UI components.

Service Workers, Manifest.json, and HTTPS are the three main technological components of PWAs.

A service worker is a technical component of JavaScript code that works as a proxy between the browser and the network, and it provides one of the major aspects of progressive web applications. Using the browser-cached API, it also handles push notifications and aids in the offline development of the web application.

The manifest file is a JSON configuration that contains vital information about the application, such as the icon displayed on the homescreen, the short name, the background color or its theme. At the same time, a splashscreen can be defined by the web developer.

Because a service worker can intercept network requests and change their replies on the client side, progressive web applications require the HTTPS security protocol. HTTPS not only protects the website's integrity by preventing intruders from tampering with communication between the website and users' browsers, but it also protects users' privacy by preventing attackers from passively listening to site-to-user interactions. [15]

3.1.4 REST API

API or Application Programming Interface is a collection of definitions and protocols that enable technology products and services to connect with one another through the Internet, allowing interaction between two software programs but not between users.

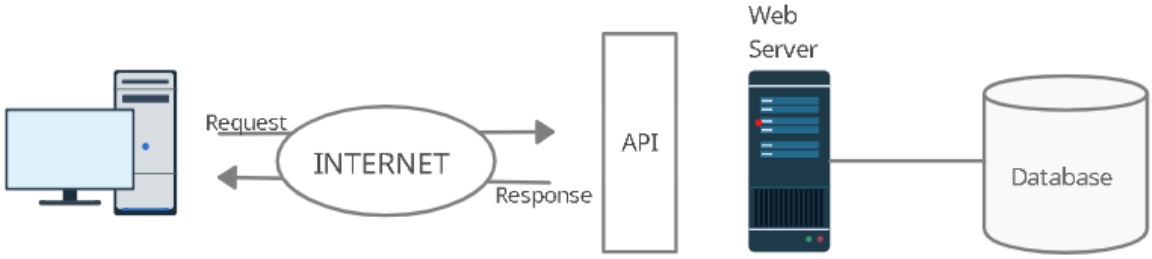


Fig. 3.2. How does API work

For an API request to work, this request must contain 4 items. The first element, the endpoint consists of two parts. The first parameter of the ednpoint is the URL, and the second is the path. The path varies depending on what you are trying to achieve. By combining these two components, we obtain a final result.

The second element, the header, contains information about the client and server. The authentication token and the customer id are two of the most frequent types of data put in the header. When an API account is created, these credentials will be automatically given.

The third element is called the method and is an action taken when sending a request. The last item, the data. Request data, often found as "body", is information that will be sent or returned by a server.

The most commonly used HTTP methods are:

- GET to provide read only access to a resource
- PUT to create a new resource
- DELETE to remove a resource
- POST to update an existing resource or create a new resource

REST (Representational State Transfer) is a web-based architecture, which uses the HTTP protocol, being introduced by Roy Fielding in 2000. [16]

Like any other architectural style, REST also has the 6 guidance restrictions that must be met for the interface to be RESTful:

- Client server, assumes that the application part of the client and the server must be able to evolve separately, the two segments being independent
- Stateless, assumes that each request received from the customer must contain all the information necessary to be able to process the request
- Cacheable, assumes that if a response is cacheable, then the client's cache is allowed to reuse the information later
- Uniform interface means that once a developer is familiar with an API, then they should be able to follow the same principles for other APIs.
- Layered system, assumes that REST allows the developer to use a layered architectural system, ie it can be deployed on server A, data can be stored on server B and requests sent to server C
- On-demand code is optional, as static resource representations are often sent as XML or JSON files, but can be returned, if needed, an executable code to support part of the application.

3.1.5 HTTP Requests

An HTTP request is a request to execute a specific action on a resource indicated by a Request-URL. Request methods are case-sensitive and should be written in capital letters. Although there are several HTTP request methods, each one has a unique function.

HTTP requests serve as a medium of communication between a client and a server or an application and a server. The client sends an HTTP request to the server, which the server responds to after internalizing the message. The status of the request is included in the answer.

3.1.6 HTML and CSS

HTML (HyperText Markup Language) is the programming language used to create web pages. HyperText is the term used to describe how web pages communicate with one another, as well as the available connection to another web page. Markup Language, on the other hand, considers how to organize HTML code, since tags inform the web browser how to structure what they want to be shown.

There is a strong relationship between HTML and CSS that results in a user-friendly page display. So CSS is the component that specifies how web pages are styled. Styles may be applied to HTMML code in a variety of ways, but the most common is through an external file. Developers' job is made simpler by the separation of design content, and the code seems clean and legible. [17]

3.1.7 JavaScript

JavaScript is a scripting language being specific in creating web pages and more. It was created to animate web pages, and programs written in this language are called scripts. Because these scripts do not require special training before running, JavaScript is completely different from the Java language. Runs on the client side of the web page, the user interacting directly with the application. [18]

With the use of JavaScript, it brings a number of benefits to users such as:

- Reduced server load;
- Possibility of working with complex and friendly interfaces;
- Increased interactivity - the ability to react to events such as mouse movement or keyboard access;
- Modulation and encapsulation capability - the ability to have scripts written by different teams, but which work the same on the same web page;
- Full integration with HTML / CSS;
- Working speed;
- Simplicity - relatively simple to learn and implement;
- Interoperability - JavaScript can be inserted in any web page regardless of its extension;
- Supported by all major browsers and enabled by default;
- Extended functionality;

3.1.8 React

In recent years, single page applications have become increasingly popular. Frameworks such as Angular, Ember and Backbone have helped developers of JavaScript-based applications build modern web applications, leaving behind the use of the jQuery

library. From the point of view of single page applications, the list can go on, there is a wide variety of frameworks. When we consider the dates on which they were released, many of them are in the first generation of single page applications: Angluar 2010, Backbone 2010, Ember 2011.

In 2013, Facebook launched React. The whole concept is based on XHP, a PHP framework. It was designed to render the entire page each time a request was made. Analyzing the design pattern MVC (Model view controller), we can say that the React initially dealt with the View part, ie the user interface. The view displays to the user information coming from the Model, but also gives him the freedom to modify the received information. It also lets the user render components as items that can be viewed in the browser.

React.js is a state machine, which helps to manage the complexity of the state over time. It achieves this due to the limited purpose, which is concerned with updating the DOM and responding to events. The DOM (Document Object Model) is the representation of object information that compromises the structure and content of a web document. This is a programmable interface for HTML and XML documents, allowing programming languages to connect to the page by representing documents as nodes and objects.

The entire React ecosystem makes it possible to build a single page application. As mentioned above, the React only delivers the View layer, but its ecosystem is flexible and adaptable to change. It has a simple API, a well-developed ecosystem and a vast community, which is why it is constantly updated and modernized.

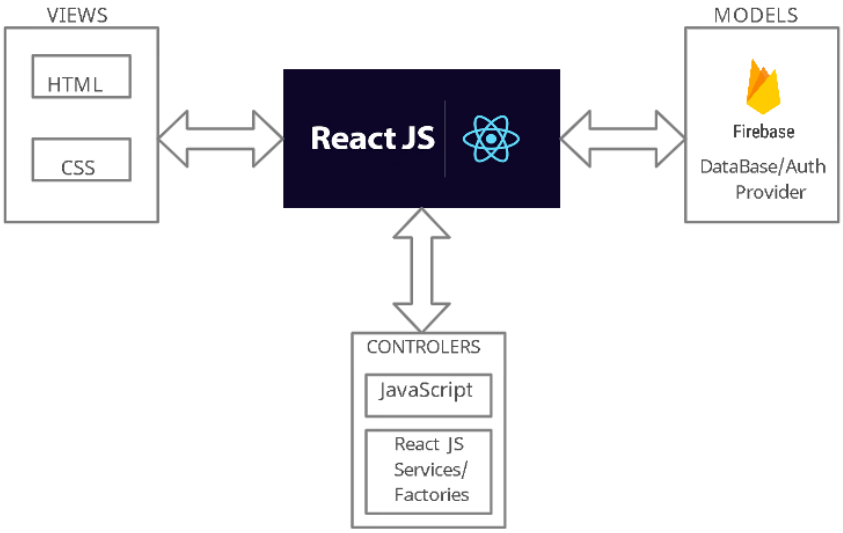


Fig. 3.3. Diagram of the MCV design pattern

3.1.9 React Lifecycle Hooks

As React mounts our application, it provides hooks for us to add our own code at various points during the component's lifetime. To hook into the lifecycle, we'll need to specify methods on our component that React will run at the right moment for each hook.

Because we're building virtual nodes, we can't rely on a component being present in the DOM instantly when it's declared on a page in our application. Instead, we must wait until the component has been successfully configured in the browser. We have two distinct hooks that we may write for functionality that we need to run once it has been mounted. One is called immediately before the component is supposed to be mounted on the page, componentWillMount() , and the other is called soon after it's been installed, componentDidMount().

We're not really creating DOM nodes since we're using React to provide virtual representations of nodes in our DOM tree. Instead, we'll create an in-memory view that React will maintain and administer. When we talk about mounting, we're referring to the process of turning virtual components into DOM elements that React can place in the DOM.

Now, we'll want to send an HTTP request when the component is about to be mounted or shortly after it has been mounted. React performs the method componentWillMount() or componentDidMount() defined in our component right before it mounts in the DOM. This is an ideal location for a GET request.

Before or after changing the actual rendering, we may wish to update some data in our component. Let's assume we want to call a function to set up the rendering or to call a function when the props of a component change. The componentWillUpdate() function is a good place to start when it comes to prepping our component for a change.

React will call a function when the component is about to receive new props. When a component receives a new set of props, this is the initial method that is called. It's a good time to search for modifications to particular props while defining this function since it allows us to compute changes and update our component's internal state.

It's important to remember that even when the componentWillReceiveProps() function is called, the props' value may not have changed. Checking for changes in the prop values is always a good idea.

React will call the componentWillUnmount() function before the component is unmounted. This is the moment to perform any necessary cleanup tasks, such as clearing timeouts, deleting data, disconnecting websockets, and so on.

These are a handful of the React framework's lifecycle hooks with which we may interact. As we develop our react app, we'll be utilizing a lot of them, so it's a good idea to be familiar with them, know that they exist, and know how to hook into the life of a component. [19]

3.1.10 Firebase

Google's Firebase platform allows developers to build mobile and online apps. In 2011, it began as a stand-alone business. Google aquired the platform in 2014, and it is now the company's main app development platform. It has more than 15 products in the Firebase package, but for this application I used only Authentication and Cloud Firestore.

To authenticate users to your project, Firebase Authentication delivers backend services, easy-to-use SDKs, and ready-to-use UI frameworks. It accepts passwords, phone numbers, and third party application accounts like Google, Facebook, and Twitter, among other methods.

To sign up or sign in a user into an app, you must first obtain the user's authentication credentials. The user's email and password, or a third party application account, can be used as these credentials. The credentials are subsequently sent to the Firebase Authentication SDK. The credentials will then be verified by our backend services, and a response will be sent to the client.

After the user managed to sign in, the developer can access the user's basic profile information and control the data that they can access in other Firebase products. The supplied authentication token can also be used to validate the identity of users in the application's backend services. [20]

Cloud Firestore is a No-SQL database that uses SDKs in order for the application to interact with it. The data is carefully stored and synced even for mobile or web applications through real-time listeners. [21]

Firestore is easy to use and has a lot of advantages that make the developers choose it, such as:

- It has an offline feature, which lets the user store data and sync it when the connection is online even for mobile or web applications;
- It works hand in hand with other Firebase products and Google Cloud becoming easier to manage the data;
- It supports indexed queries;
- The developer pays for the features that are used not for the full package;
- Since it uses SDKs, there is less code to worry about;
- Because it is a Google platform it gives excellent security features for data;

3.1.11 Tabletop.js

Tabletop.js is an open-source JavaScript library that developers use in order to integrate information through Google Sheets on an application or project. It is easy to set it up and to modify the data. The library allows you to retrieve JSON from a Google Spreadsheet without having to go through hoops.

The main advantages of using Google Sheets as backend through tabletop.js are:

- Super accessible from everywhere; all that is needed for that is a Google Account and a browser;

- It allows you to revert to an earlier version if an error has been found;
- The developer can share them easily, having three ways to do that, as a viewer, letting the user just to see the data, as a commenter, you can see the data and also to let comments in regards to it and as an editor that gives you full control of it;

3.1.12 Augmented Reality

Augmented Reality provides us with unique benefits by merging the actual and virtual worlds. This is a new method of controlling how we interact with the environment. This technology augments virtual information on top of the actual environment with modifications made by user which can control the point of view, without replacing the real world completly. [22] [37]

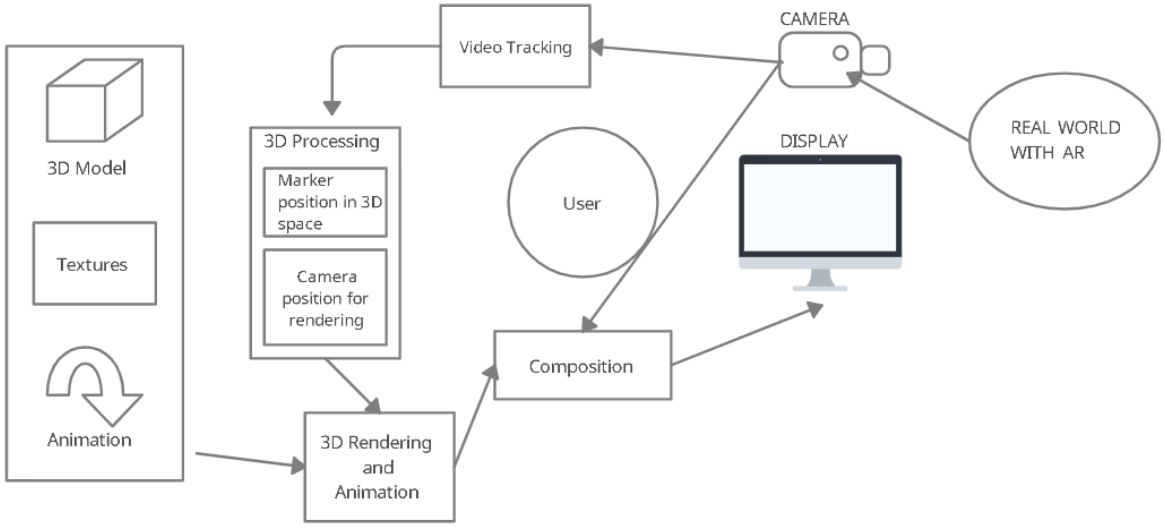


Fig. 3.4. How does AR work?

3.1.12.1 The use of Augmented Reality

Augmented Reality provides us with unique benefits by merging the actual and virtual worlds. This is a new method of controlling how we interact with the environment. This technology augments virtual information on top of the actual environment with modifications made by user which can control the point of view, without replacing the real world completly.

It gives the viewer a composite vision that includes both real-world and computer-generated virtual scenes. This is a method of augmenting the actual world by interacting with an everyday location, environment, item, or event in a somewhat unmediated manner. By integrating augmented reality interfaces with instructional material, we can provide learners with seamless interaction between the real and virtual worlds. [33]

This revolutionary method improves the efficiency and appeal of teaching and learning. The capacity to superimpose computer-generated virtual objects onto the actual

environment alters how we interact, and trainings become real-time rather than static experiences.

Augmented Reality enhances the user's vision and engagement with the actual world by bringing virtual information or objects to any indirect view of the user's real-world surroundings. Augmented Reality aims to overlay virtual items or scenes on real-world objects or situations in order to give a more intuitive user experience in real time.

It is an interactive environment in which virtual objects are used to improve real life. Augmented Reality must have three characteristics: it must combine the actual and virtual worlds, interact with the user in real time, and be registered in 3D. [34]

3.1.12.2 Augmented Reality in education

People can be educated and trained in a variety of methods depending on the knowledge and skills they require. Classroom lectures using textbooks, laptops, portable devices, and other electronics are examples of these methods.

The ability to learn innovation is determined by a person's access to numerous technologies and the infrastructural environment in which he or she lives. In a fast changing culture with a wealth of available information and expertise, it is critical to accept and use information at the appropriate time and in the right place to maintain efficiency in both educational and corporate contexts.

Augmented Reality is a technology that made big changes in regard of education and not only. It has been applied in military, medical, and engineering design applications.

Using virtual objects that the user cannot directly detect with his own senses to display information allows a person to engage with the actual environment in ways that were not previously imaginable. With the help of augmented reality interaction methods, we may modify the location, form, or other graphical aspects of virtual objects. [23]

We may manipulate virtual things as well as actual objects in the real world by using our fingers or actions on portable devices such as shake and tilt. By improving a user's view of and interaction with the actual environment, Augmented Reality can be used for learning, entertainment, or edutainment. Like a genuine item, the user may move around the three-dimensional virtual picture and see it from any angle.

Users can conduct real-world operations using the information provided by virtual items. Collaborative work can also benefit from augmented reality. Innovative computer interfaces that blend the virtual and real worlds can be created to improve face-to-face and remote communication.

The use of this technology in augmented reality textbooks is another intriguing potential. These books are printed regularly, but when you aim a webcam at them, they come alive with visuals and interactivity. Installing specific software on a PC, utilizing particular mobile apps, or visiting a website are all options. After a book is published, this technology allows it to be turned into an augmented reality version. The simplest method

to connect the two separate worlds is to use 3D objects and perspectives, diverse and imaginative media, simulations with various sorts of interactions. [24]

3.1.12.3 Augmented Reality vs Virtual Reality

Virtual reality or VR is a technology that allows us to immerse oneself in an artificial environment; this world might be fully fictional or simply a replica of reality. Visual, auditory, and tactile experiences are all possible. This is accomplished through the use of a virtual reality headgear with a stereoscopic 3D display system placed in front of the eyes. Some versions have sensors that detect head tracking so that the user may gaze around. The pictures are then recalculated in real time to match the head's or gaze's orientation.

The ability to manipulate the items of the synthetic environment with controllers like the Oculus Touch is a significant advancement in the area of virtual reality. This allows students, for example, to practice and learn in a more engaging way by interacting with items in the virtual world.

The possibilities for this technology are endless: training with simulators, medical procedure simulations, architecture, archeology with site reconstructions, virtual museum tours, phobia therapy, and different forms of learning.

Virtual reality's value in education and learning represents a shift from its ability to improve and promote learning, expand memory capacity, and make better judgments while working in an enjoyable and engaging environment. In reality, when we read textual information, our brain goes through a process of interpreting what we read, which puts more strain on our cognitive abilities. Because there are less symbols to analyze and the comprehension is more direct when using virtual reality, the process of interpretation is sped up.

Despite their comparable designs, virtual reality and augmented reality achieve two very distinct things in two very different ways. AR allows users to interact with the actual environment that has been digitally altered or augmented in some way. Virtual reality, on the other hand, takes the user out of the actual world and replaces it with a totally synthetic one.

Because VR needs total immersion, VR gadgets entirely block out the outside world. The smart glasses with AR capabilities, on the other hand, have clear lenses. Understanding these distinctions is crucial in identifying which use cases are suitable for each. [35]

AR applications are mainly used when the user has to be connected to the present in the actual world. Remote help, on-the-job training, remote collaboration, and computer-assisted activities are all examples of AR enterprise solutions. Researchers discovered that AR is well-suited for industrial use cases, notably workforce training and product maintenance. Companies experiencing knowledge gaps and expertise loss are digitizing that information and sharing it with less-experienced personnel using AR technologies. [36]

Since VR is mostly used to not interact with the real world, but to create virtual environments, such as remote collaboration with 3D features, point-of-view teaching, and virtual tours. Virtual reality software can be developed to help orthopedic surgeons and nurses enhance their training. Other businesses utilize virtual reality to create both improbable and common circumstances in order to provide colleagues with hands-on training without disturbing operations. [25]

3.1.12.4 A-Frame

The A-Frame web framework is used to create virtual reality experiences. Because it is built on top of HTML, it is easy to get started. However, A-Frame is more than just a 3D scene graph or a markup language; it is also a solid entity-component framework that gives three.js a declarative, extensible, and composable structure.

A-Frame was created to be an easy yet powerful approach to generate VR content. It was first designed within Mozilla and is currently maintained by the co-creators of A-Frame within Supermedium. A-Frame has evolved to be one of the largest VR communities as an independent open source project.

Most VR headsets, including the Windows Mixed Reality, Vive, Rift, Daydream, Cardboard, GearVR, and Oculus Go, are supported by A-Frame, which may also be used for augmented reality. A-Frame seeks to create truly immersive interactive VR experiences that go beyond simple 360° material and make full use of positional tracking and controllers, despite supporting the whole spectrum. [26]

A-scene comes with great features such as:

- WebVR is built into A-Frame from the ground up. While the DOM is used by A-Frame, its elements are not affected by the browser layout engine;
- You can build applications for Rift, Windows Mixed Reality, GearVR, making it cross-platform;
- Easy to use and understand because it is based on HTML;
- It has an Entity-Component Arhitecture;
- It has a variety of components;

3.1.12.5 A-entity

A-Frame represents an entity via the <a-entity> element. Entities are placeholder objects to which we plug in components to provide them with look, behavior, and functionality, as specified by the entity-component-system design.

To make it render or perform something, we may connect components to it. We may attach the geometry and material components to give it form and appearance:

<a-entity geometry="primitive: box" material="color: blue">

In order to modify its appearance, we have the following parameters:

- Components - This offers us access to the entity's components, including data, state, and methods for each component.
- isPlaying – If the entity is active, and we pause it, isPlaying becomes false
- hasLoaded – It checks if all the components were initialized
- object3DMap – It lets you use THREE.Object3D
- seceneEl – It sets a reference to a scene element

3.2 Detailed Design

3.2.1 Unified Modeling Language (UML)

The Unified Modeling Language is a visual representation language for the architecture, design, and implementation of large software systems. There are many code lines in an application while you are developing code, and it is tough to keep track of the links and hierarchies inside a software system. The software system is divided into components and subcomponents using UML diagrams.

UML diagrams allow software models to be generated, reviewed, and changed during analysis and design, similar to how drawings show an architect the exact design of a structure. Modeling is a technique used in a variety of areas, notably engineering, because it assists in the understanding of more complex system processes and allows for project evaluation based on a variety of criteria such as flexibility.

Because UML is a standardized modeling language that can be used to a variety of programming languages and development methods, the vast majority of software developers will be familiar with it and able to utilize it in their work.

We create UML diagrams to help us comprehend the system more clearly and simply. A single graphic is insufficient to represent all features of the system. UML specifies a variety of diagrams to address the majority of a system's features. Diagrams are split into two major categories, each of which is further broken into subclasses.

Structural Diagrams are the static part of the system. These static features are the components of a diagram that make up the primary structure and are hence stable. Classes, interfaces, objects, components, and nodes are examples of static parts. The following are the 7 structural diagrams: Component Diagram, Deployment Diagram, Object Diagram, Class Diagram, Package Diagram, Profile Diagram and Composite Structure Diagram. [27]

Behavioral diagrams are used to show the dynamic nature of a system. The moving elements of a system are referred to as the dynamic aspect. These diagrams are : Use Case Diagram, Activity Diagram, Sequence Diagram, Communication Diagram, Timing Diagram, State Machine Diagram and Interaction Overview Diagram.

I used some of the previous mentioned diagram types in order to organize and structure my application better. They are : use case diagram, database diagram and sequence diagram.

3.2.2 Use Case Diagram

The primary form of system/software requirements for an undeveloped software application is a UML use case diagram. The desired behavior is specified in use cases, not the specific technique of achieving it. Once defined, use cases can be represented both textually and visually. A fundamental notion in use case modeling is that it assists us in designing a system from the standpoint of the end user. It's a good way to communicate system behavior to users in their own words by describing every externally observable system activity.

A use case diagram's syntax is quite simple, and it does not use as many symbols as other UML diagrams. In order to be a good diagram it has to have use cases and actors linked by associations.

The traditional method of portraying actors is using stick figures. Actors can represent non-human components such as an email system, despite the fact that they are depicted as humans. An actor might be an active user of the system, triggering use cases that way, or a passive user of the system, allowing use cases to be realized. When an actor is created, it must be associated with at least one use case. Because an actor represents a role, it may be filled by many individuals or systems.

Ovals are commonly used to represent use cases. A use case defines the aims of a user's use of a system's capabilities from the user's perspective. To cover alternate routes, the action order may and should be entered here. When a use case is only described through another use case, it is said to as abstract and is labeled as such in the use case. A use case's multiplicity can also be defined. It specifies the number of times a use case can be carried out.

A communication connection, also known as an association, determines whether or not an actor is participating in a use case. For exemple, a simple line is used to identify it in the use case diagram. The types of associations are : generalization, dependency, include and extend. [28]

3.2.2.1 Use case for an abstract user

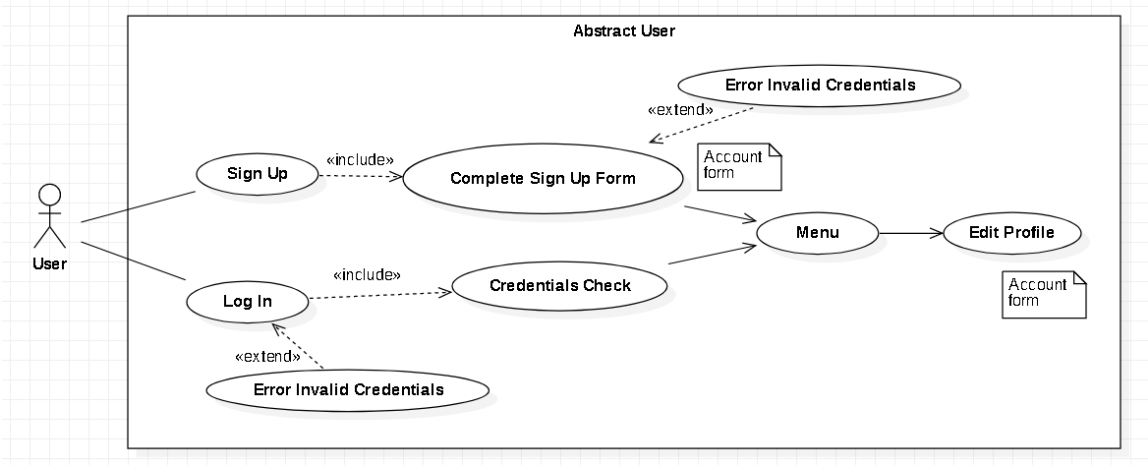


Fig. 3.5. Use case for an abstract user

In the figure above the abstract user has the general cases that any user can have on my web application.

Firstly, when a user enters my application on a mobile or computer, he is required to sign in or sign up in order to use the functionalities of it.

If you are a new user, you will have to press the sign up button and you will be redirected to a sign up form to complete using your basic information and also to check if you are a teacher or not. All the inputs on the form has to be filled, each of them having specific requirements. After you managed to do that, there will be two scenarios: it will either let you sign in into your new account, or display an error regarding the requirements for the fields.

For signing in, you will have to fill out the two inputs for the e-mail address and password that you set up for your account. Then, either an error can be seen on the screen saying that the e-mail or password is invalid, or you will manage to get to the home page.

After you are logged in into your account, another common action that you can do is to edit your profile. This can be achieved by pressing the edit profile button on the header, which will send you to a form that has the same fields as the one from the sign up part. After completing it, you can finish editing it or just cancel in case you do not want to change it anymore.

I used the include association, firstly for the sign up part, because in order to sign up you will have to complete a form, so everytime we want to do one thing, the other will have to be done as well. The same thing applies also for the second time I used it, for the sign in part, where in order to log in it will have to check your credentials.

The extended relationship was used in both sign in and sign up in the case where the fields were not filled out properly, so the user will receive an error. It is an extended associations because the error supplements the sign in or sign up but it is not completely necessary in the process.

3.2.2.2 Use case for a student

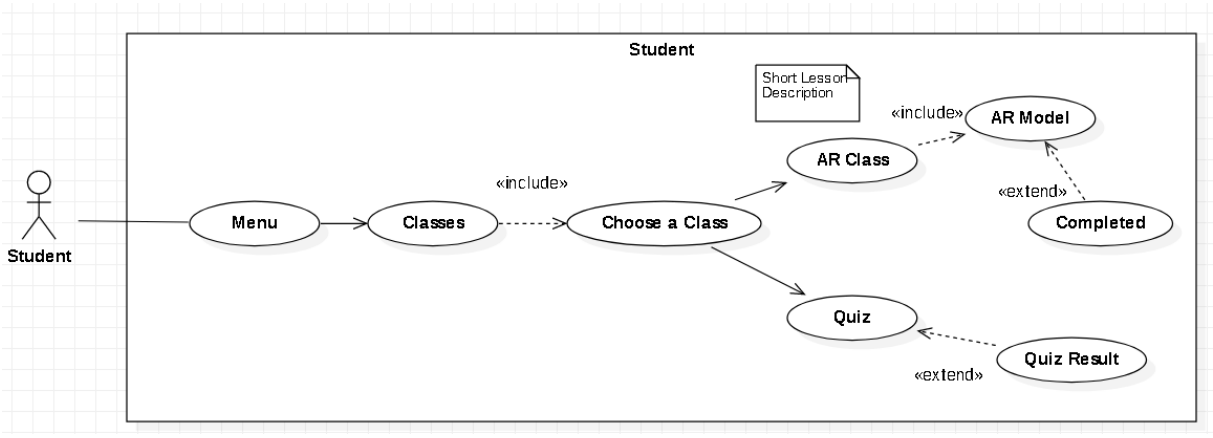


Fig. 3.6. Use case for a student

After we managed to create or access our account, we find that there are two different types of users: student and teacher.

For a student, when you sign in to my application, you will be on the page called Classes, where you will have to choose which class you want to check, by using the dropdown. After choosing a class, the user will see multiple cards, each of them being a lesson for that specific class. After you choose one of them, a short description will appear and a button to get you to see the AR model. After you saw it you can click to complete the assignment, which will turn the lesson to be completed for the student.

Another feature that the student has, is the quiz for every subject. After pressing the button to go to the quiz, he can complete the multiple choice quiz and then the result for it will come up instantly, showing the user how well he performed.

In this figure I used include for the fact that, you have to choose a class in order to see more info about it, or to choose a card in order to get that AR model for it. This shows that both of the actions will be done.

For the extend association, you can either finish your assignment or quiz in order to get you to the use cases completed or quiz result but at the same time you can choose to go back. This means that the cases does not have to be completed if the users choose not to.

3.2.2.3 Use case for a teacher

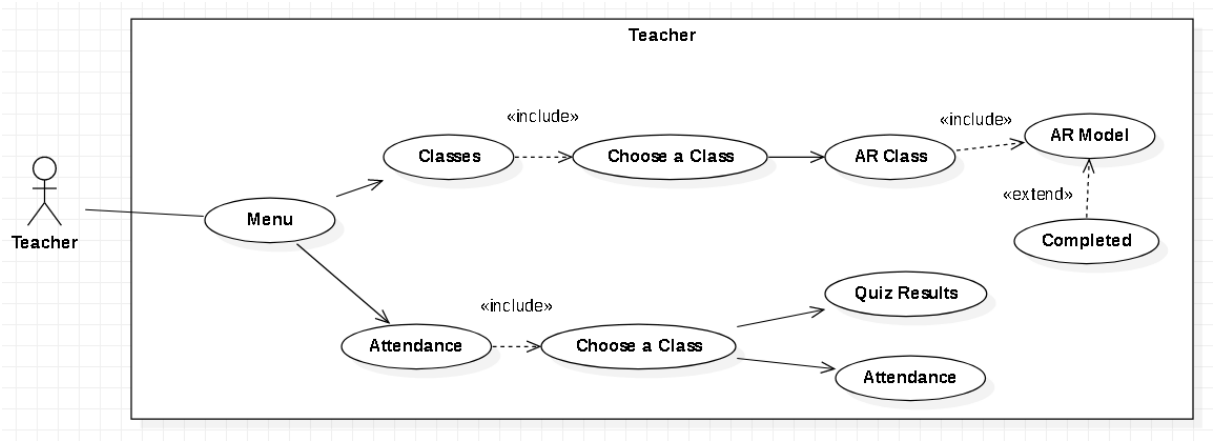


Fig. 3.7. Use case for a teacher

In this case, where the user that signed in is a teacher, he or she will be redirected to the homepage, Classes, but this time on the header we will see next to classes the Attendance page.

For the Classes page, the teacher can do the same actions as a student can, but just for the class accordingly to his job description. There, he can check the classes, their status and also the AR models attached to each of the class.

When the user changes to the Attendance page, he will have for each of his students the attendance for each class and the quiz results. In order for the teacher to see if a class was completed by the student, the student has to check the AR model from the assignment and then to complete it. The quiz results are shown under the names of the students by showing how many points they did, above a graphic.

3.2.3 Sequence Diagram

In UML, sequence diagrams describe how items interact with one another and in what order they interact. It's important to note that they only display interactions for a certain scenario. The processes are shown vertically, with arrows indicating interactions. [29]

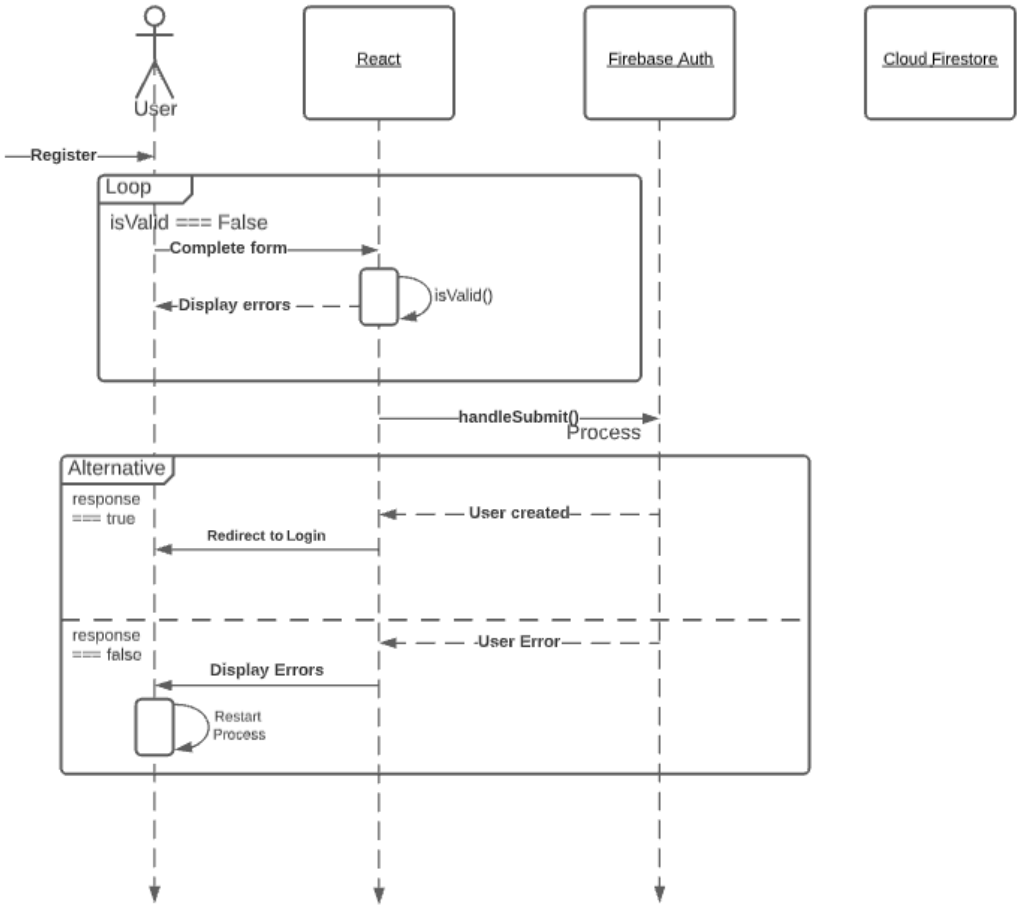


Fig. 3.8. Sequence diagram for Register

For the register part, we will have a loop that is going to repeat until all the requirements from each field is valid. After that, the user will press register button and we have two cases, one where the response is true and the user was created and redirected to the Log in page, or the response is false and the user will receive the error that the register did not succeed and the process starts again .

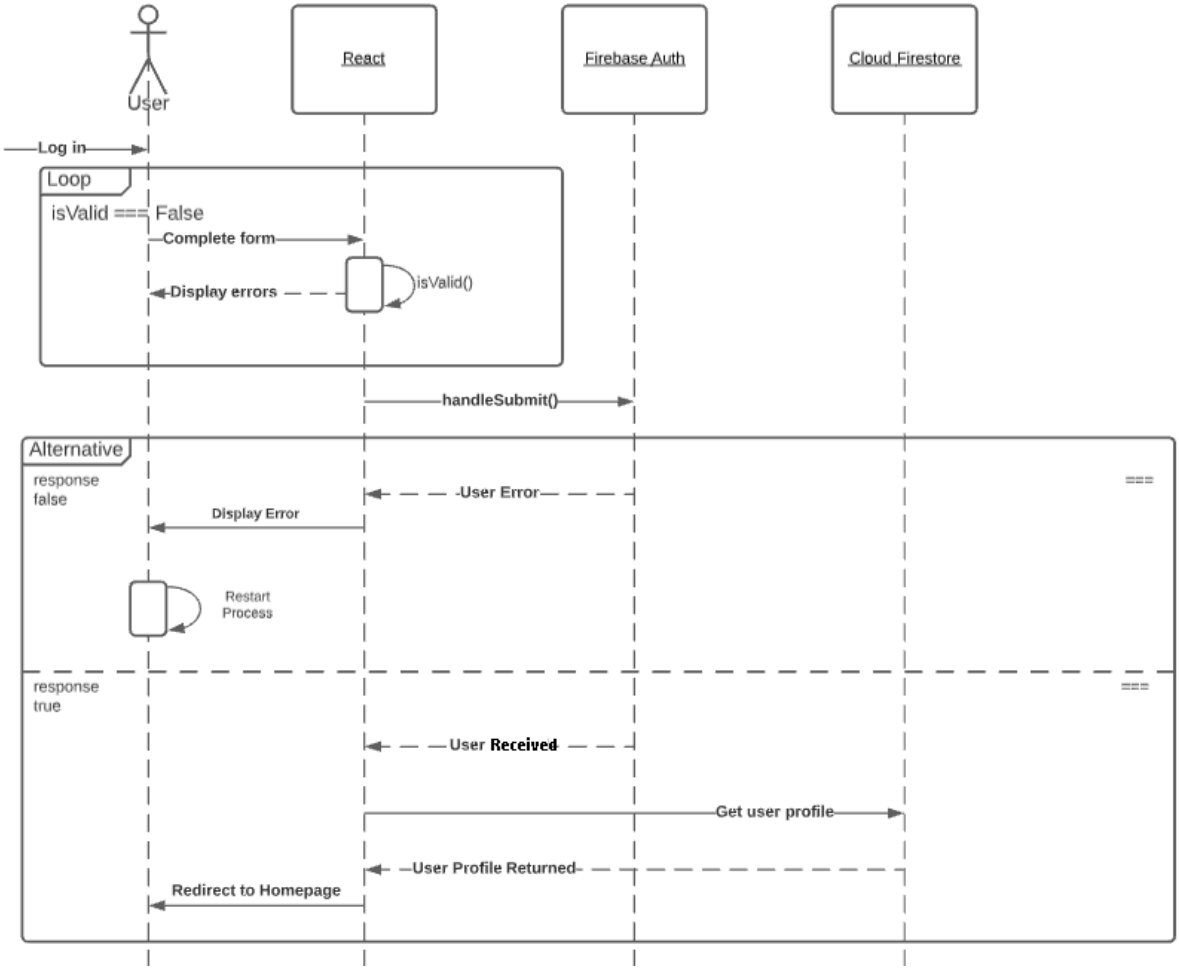


Fig. 3.9. Sequence diagram for Log in

In the sequence diagram for the Log in page, we will have the same loop as we have for the Register page that is going to repeat until all the fields are completed.

Then, after pressing the button to Log in and the handleSubmit() is called, we will have another if statement.

The first part is going to be the case when the response is false, displaying the error message because the log in failed and the user will start the process again.

For the other case, the user was received and then React will make a call to Cloud Firestore to get the user's profile. Firestore will return the data requested and then the user will be redirected to the Homepage.

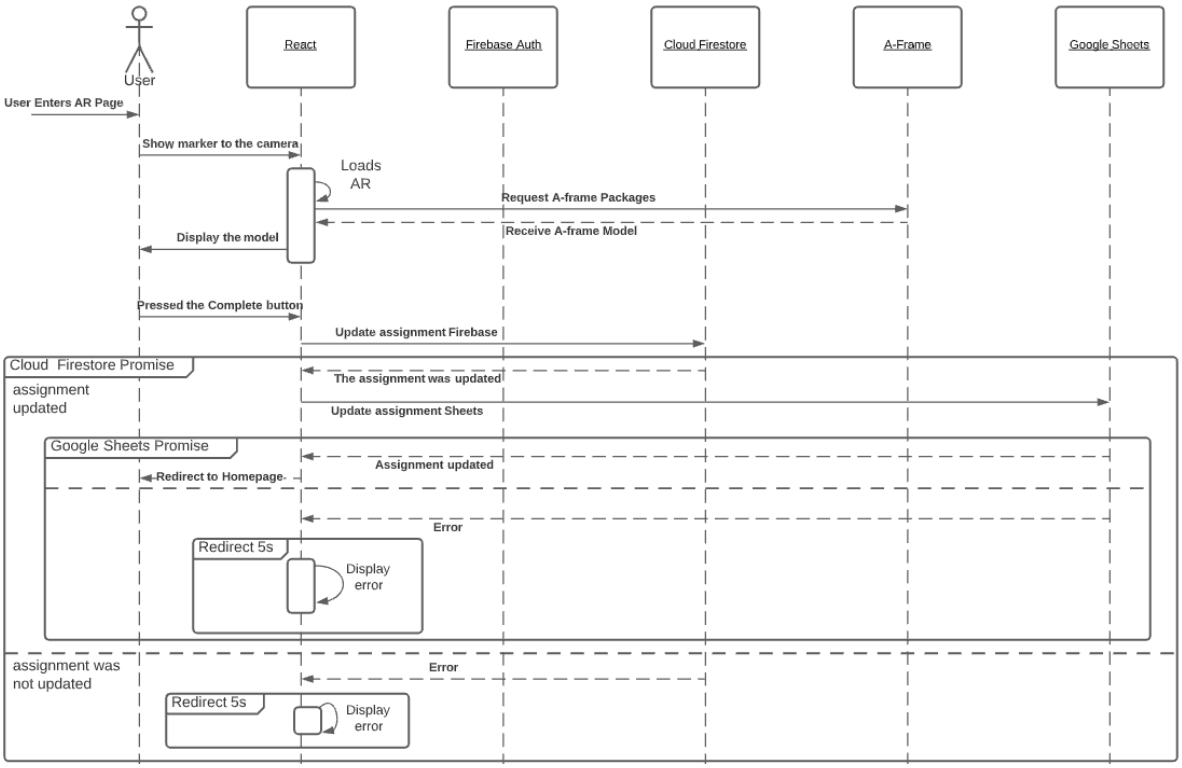


Fig. 3.10. Sequence diagram for the AR page

In the figure above we can see the sequence diagram that describes how an user enters the AR page and what is about to happen.

Firstly, the user shows to the camera the marker for the specific 3D model that he wants to see. Then, the AR is loading, requesting A-frame packages to display the model. The user sees the 3D representation and after some research he presses the complete assignment button.

Since the user finished the assignment, we want to update this in the Cloud Firestore and Google Sheets. For the Firestore we will update the data for the individual student, meanwhile in Sheets we upload data for the Attendance page.

The Cloud will store the fact that the assignment was completed for the student's Classes page. Google sheets on the other hand will get the data in order to show the teacher which students managed to complete the assignment.

If one of the calls for storing data did not succeed, after five seconds, the user will be redirected to the homepage.

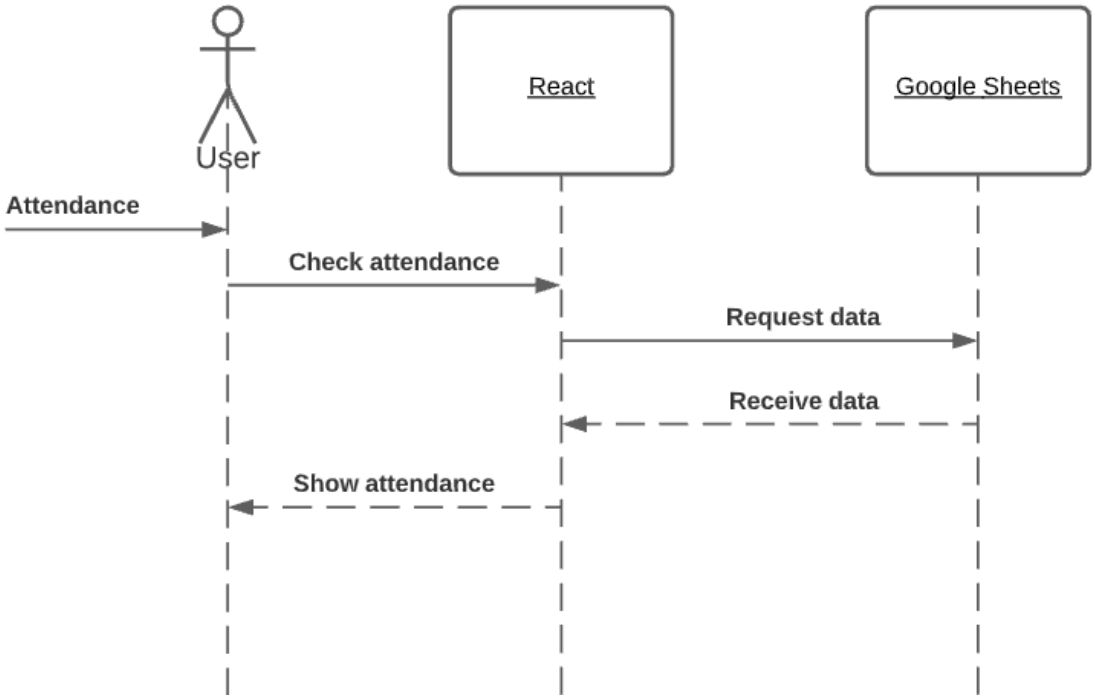


Fig. 3.11. Sequence for attendance

If a teacher wants to check the attendance list for a class, a request for it will be sent to the page. The page will request the data that we have stored in the spreadsheet Attendance.

The response from Sheets will be sent to the page, rendering it to the user. The list with attendance and quiz results will appear on teacher's screen.

3.2.4 Architecture of Cloud Firestore

Cloud Firestore is a document-oriented database that runs on NoSQL. There are no tables or rows in this database, unlike a SQL database. Instead, you save information as documents that are then grouped into collections.

A collection of key-value pairs may be found in each document. Cloud Firestore is designed to hold huge groups of tiny documents.

Collections must be used to hold all documents. Subcollections and nested objects, both of which can include simple fields like strings or sophisticated objects like lists, can be found in documents.

Cloud Firestore automatically creates collections and documents. Simply said, data may be assigned to a document within a collection. Cloud Firestore generates the collection or document if it does not exist.

The document is the storage unit in Cloud Firestore. A document is a simple record with fields that correspond to values. A name is assigned to each document. If an object is more complex , such as a user's name, it is called map.

It can be observed that the documents resemble JSON. They are, in reality, essentially the same. Although there are significant distinctions, documents may be treated as lightweight JSON records in general.

Documents are stored in collections, which are nothing more than storage containers for documents. You might, for example, have a users collection that contains all of your users, each of them being represented by a document.

Cloud Firestore gives you total control over the fields you include in each document and the data types you store in them. Various fields or different types of data can be stored in the same fields across documents in the same collection. However, using the same variables and data types across numerous documents is a smart idea so you can query them more quickly.

A collection consists solely of documents. It can't have any raw fields with values in it, and it can't have any other collections in it.

The names of the documents in a collection are all different. You may either give your own keys, such as user IDs, or Cloud Firestore can generate random IDs for you.

There's no need to add or remove collections. The collection exists after you create the first document in it. A collection is no longer active if all of the documents in it are deleted. [30]

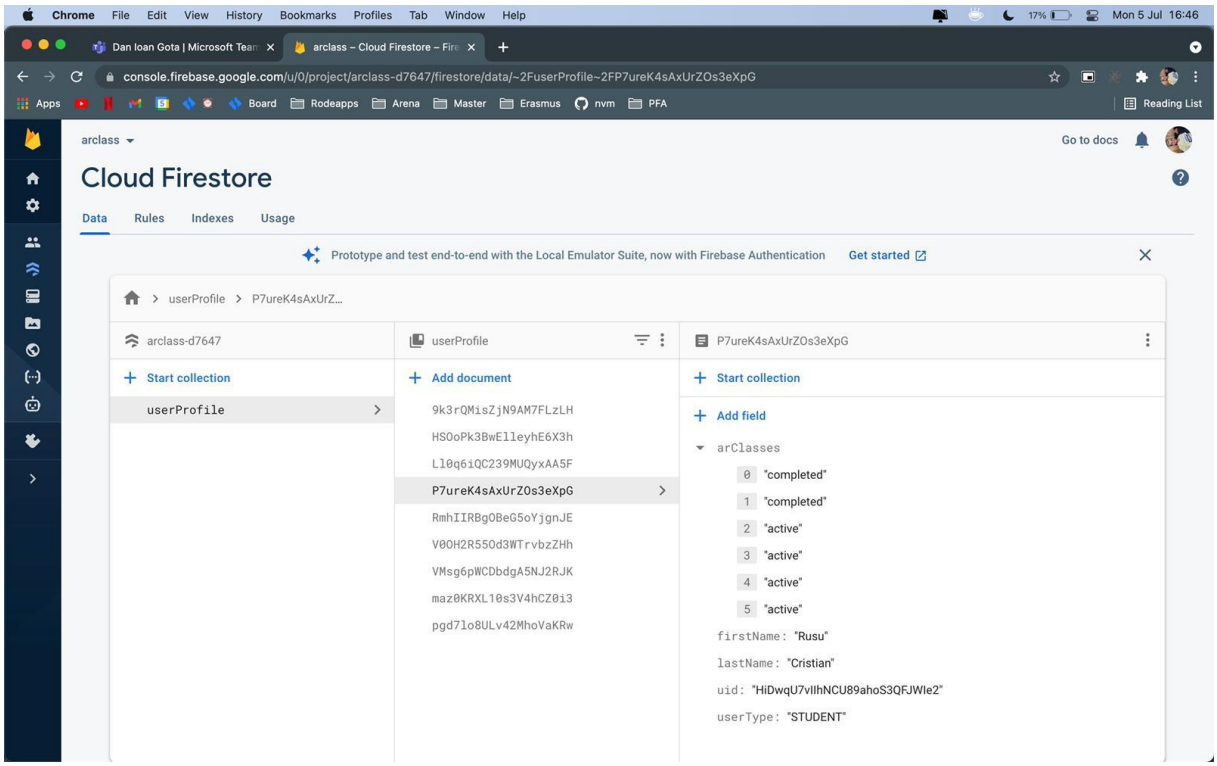
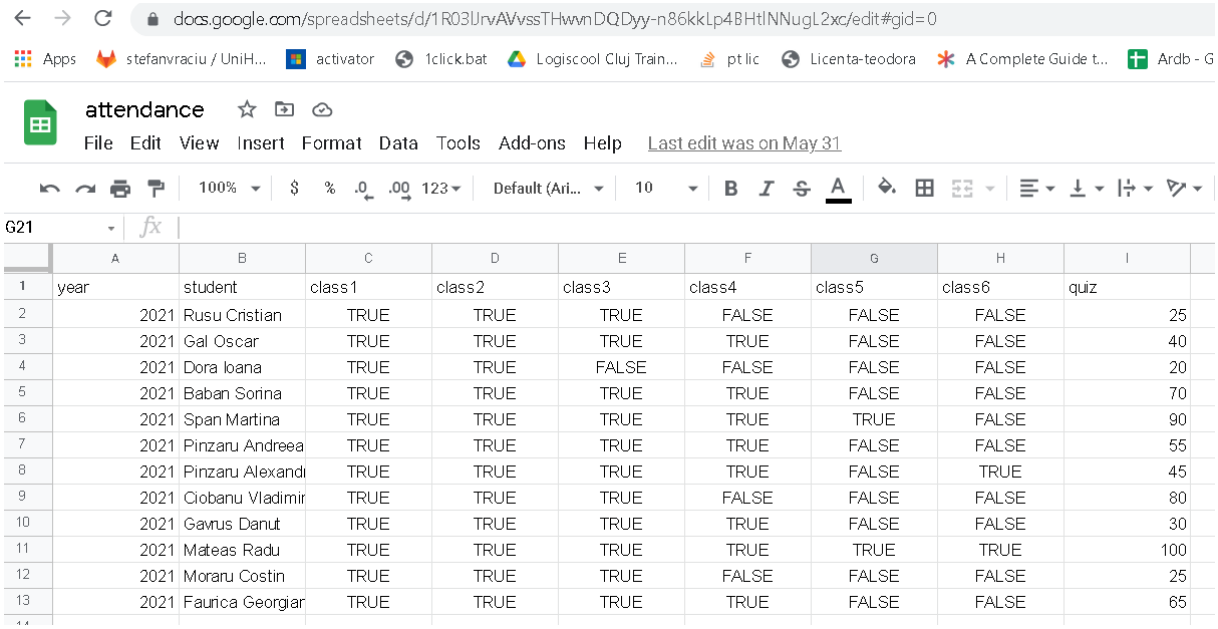


Fig. 3.12. Firestore Arhitecture

3.2.5 Google Sheets

In my application I chose Google Sheets to store data. I used three Google Sheets: one for the dropdown that I will use for classes, one for the Attendance page which stores every Student with his quiz result and assignments for that specific class and the last one for the cards that have the lessons prepared with the name, content, photo and status of it. In the next figure I will present you the spreadsheet made for the Attendance page.


docs.google.com/spreadsheets/d/1R03IJrvAVvssTHwvnDQDyy-n86kkLp4BHtlNNugL2xc/edit#gid=0

Apps | stefanvraciu / UniH... | activator | 1click.bat | Logiscool Cluj Train... | pt lic | Licenta-teodora | A Complete Guide t... | Ardb - G

attendance

File Edit View Insert Format Data Tools Add-ons Help Last edit was on May 31

100% $ % .0 .00 123 Default (Ari... 10

G21

	A	B	C	D	E	F	G	H	I
1	year	student	class1	class2	class3	class4	class5	class6	quiz
2	2021	Rusu Cristian	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	25
3	2021	Gal Oscar	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	40
4	2021	Dora Ioana	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	20
5	2021	Baban Sorina	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	70
6	2021	Span Martina	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	90
7	2021	Pinzaru Andreea	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	55
8	2021	Pinzaru Alexandi	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	45
9	2021	Ciobanu Vladimir	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	80
10	2021	Gavrus Danut	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	30
11	2021	Mateas Radu	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	100
12	2021	Moraru Costin	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	25
13	2021	Faurica Georgiar	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	65

Fig. 3.13. Attendance Spreadsheet

We have nine columns on the first row representing the data that we want to use for our application. The first one represent the year of study in order to represent the students that you want to check. Then we will have the full name of the student.

From column C to H we have the classes for that subject which has true or false telling the teacher if the student attended that specific class or not (and the 3D model for it) .

The last column represents the grade obtained by each student at the quiz regarding that subject.

3.2.6 Tabletop.js

Spreadsheets are popular because they function as a database that non-developers may easily change. Tabletop.js was used to retrieve data from Google Spreadsheets for my application. The library allows you to retrieve JSON from a Google Spreadsheet without having to go through a million hoops.

In the next part I will show you how I set up and use tabletop in order to integrate Google Sheets in the application.

The first step is to publish the spreadsheet. In order to do that you will have to press File and then Publish to the web. After choosing the options click on Start Publishing. By pressing it you will receive an url and a warning to make sure that the content is visible to anyone.

To share the document you will have to press the share link button followed by change button. There we have to make sure that anyone can view preference has been chosen the document and press save. The last step here is to copy the link that was shared.

Next, we will have to install tabletop by writing the command in the terminal:

```
E:\Licenta\frontend>npm install tabletop -save
```

In order to use it, you will have to write the react component that will get the data from the sheet. In that component we initialize the tabletop.js library inside the ComponentDidMount() lifecycle hook. The function init will take an object for the argument. The key represents a part of your URL for the Google Sheet. The code can bee seen in fig. 3.9.

After the tabletop library has finished initializing, it will retrieve data from Google Sheets. If the query is successful, the obtained data is saved in the data variable of the react hook. [31]

The data is shown on the screen in a simple manner. We're going to map the data into an array as seen on fig. 3.10.

```
componentDidMount() {
    Tabletop.init( options: {
            key: '1DeEOfHeq3FPtRV6A-6yhEUWr_xpy80Fvqwed5mvbJcU',
            simpleSheet: true
        }
    ).then((data) => {
        this.setState( state: {vrclasses: [...data], loading: false})
    })

    Tabletop.init( options: {
            key: '1s8cBN1wWcSLdRnQ_TFf5SbQAc90jfvyXXZMy9DZJS8w',
            simpleSheet: true
        }
    ).then((data) => {
        let ceva = data.map((el) => {
            return {...el, selected: el.selected === "TRUE"}
        });
        this.setState( state: {classes: [...ceva], loading: false})
    })

}
```

Fig. 3.14. React component that gets data from the sheet

```
renderClass(cardClass) {
    return (
        <div key={cardClass.id}
             className={`card-class ${['disabled', 'completed'].includes(cardClass.status) ? 'no-event' : ''}`}>
            {cardClass.status === "active" &&
            <div className={`card-class-status ${cardClass.status === "disabled" ? "disabled" : "completed"}`}/>
            }
            <div className='card-class-inner'>
                <div className='card-class-front'>
                    <div className='photo-card' style={{backgroundImage: `url("${cardClass.photo}")`}}/>
                    <div className='title-card'>{cardClass.title}</div>
                </div>
                <div className='card-class-back'>
                    <div className='content-card'>{cardClass.content}</div>
                </div>
            </div>
        </div>
    )
}
```

Fig. 3.15. The use of data from the sheet

3.2.7 Create React Application

The ReactJs library was chosen because of its capabilities, which make life easier for every developer. The usage of components, which was made possible by ReactJs, was a huge assistance. Create React App is a popular approach to build single-page React apps, with a current setup and minimal configuration needs. Certain tools, such as Babel or Webpack, are preset and hidden, allowing the developer to concentrate on the code.

All you have to do to create the project is install Node.js and run the NPX create-react-app command and the project name. The default script for running the project is accessed from the console by entering the NPM START command for development mode, which will automatically open the web browser in http://localhost: 3000. The page will reload automatically each time the source code changes. [32]

A component divides the user interface into separate, reusable elements. This accepts an optional input and renders a React element on the screen. A state may or may not exist for a component. Components having states should be declared as classes, whereas those without states should be declared as functions. If we use ES6, we may use the React.createClass () or React.Component method to create these components.

We picked the option in which the components are instantiated as classes because the majority of the components used in this project depend on their state. We may look at the Login component as an example. First and foremost, we must court imports.

```
import React from "react";
import {Link} from "react-router-dom";
import './LogIn.scss';
import {LocalStorage} from "../../utils/LocalStorage";
import {logIn} from "../../networking/Requests";
```

We import React and other files that will help us in the code later.

```
export default class LogIn extends React.Component {

    constructor(props) {
        super(props);
        this.state = {
            showPassword: false,
            emailInput: '',
            passwordInput: '',
            emailError: null,
            passwordError: null,
        }
    }
```

By class Login extends Component, we declare the component by creating a class. As you can see, this component contains five states, each of which is defined inside this.state and serves the purpose of storing data to be utilized later within the component. The component re-renders every time the state changes. The initial state is simply assigned by the constructor class.

```
handleInputChange(value, key) {
    if (key === 'email') this.setState( state: {emailInput: value});
    if (key === 'password') this.setState( state: {passwordInput: value});
}
```

After instantiating the state, we will write functions that decide what will happen to it along the way. The handleInputChange function triggers an event, which when the user enters his email and password in the form, the initial state for emailInput and passwordInput will change, receiving the value entered in the input. We have also created similar functions for all states.

```
handleSubmit() {
    this.setState( state: {passwordError: null, emailError: null});
    if (!this.isValid()) return;
    auth.signInWithEmailAndPassword(this.state.emailInput, this.state.passwordInput) Promise<UserCredential>
        .then((user : UserCredential ) => {
            db.collection( collectionPath: 'userProfile') CollectionReference<DocumentData>
                .where( fieldPath: 'uid',  opStr: '==', user.user.uid) Query<DocumentData>
                .get() Promise<QuerySnapshot<DocumentData>>
                .then((snapshot : QuerySnapshot<DocumentData> ) => {
                    localStorage.setItem('userProfileId', JSON.stringify(snapshot.docs[0].id))
                    localStorage.setItem('user', JSON.stringify(snapshot.docs[0].data()))
                    localStorage.setItem('userType', snapshot.docs[0].data().userType)
                    this.props.history.push('/')
                })
        }) Promise<void>
        .catch((err) => {
            this.setState( state: {emailError: err.message})
        })
}
```

The function handleSubmit() is essential to the component Log in. This triggers after the user fills the inputs for e-mail and password and presses the log in button. We are using from the auth module the signInWithEmailAndPassword which returns a promise. If there was a successful response, we are creating a query to get the user profile from Cloud Firestore and save the data retrieved from the servers in the localStorage in order to access it from everywhere. Then, we redirect the user to the Homepage. In case there was a problem, we display the error.

This function is linked to the submit button via the following line of code:

```
<div className='action-button login-button' onClick={() => this.handleSubmit()}>Log In</div>
```

Instead of loading complete new pages through the server, single-page apps rewrite parts of a page. The application uses a router to travel between components, altering the browser's URL, updating its history, and keeping the user interface in sync. The most popular routing library in React is React Router. It gives the developer the ability to design routes in the same declarative manner. To enable this routing mode, just execute the following command at the terminal: npm install --save react-router-dom.

In the proposed application, the routing takes place inside the App.js file, as follows:

```
<Router>
    <Route component={Header} />
    <Switch>
        {/* Authentication */}
        <Route exact path="/login" component={LogIn} />
        <Route exact path="/register" component={SignUp} />
        <Route exact path="/reset-password" component={ForgotPassword2} />
        <Route exact path="/forgot-password" component={ForgotPassword1} />
        <Route exact path="/registration-complete" component={RegistrationComplete} />
        <Route exact path="/edit-profile" component={EditProfile} />
        {/* Homepage */}
        <AuthRoute exact path="/" component={Homepage} />
        {/* Attendance */}
        <AuthRoute exact path='/attendance' component={Attendance} />
        {/* Quiz Page */}
        <AuthRoute exact path="/quiz/math1" component={Quiz} />
        {/*  404  */}
        <Route component={NotFoundPage} />
    </Switch>
    <Route component={Footer} />
</Router>
```

Routes declared in Route are accessible by any user, whether logged in or not, and those inside the AuthRoute tag are only accessible once the user has logged in.

3.2.8 Test results for PWA

Lighthouse audits built into Chrome Dev Tools can be used to determine your application score. To run such a PWA test, all you have to do is enter Dev Tools by pressing the F12 key, select the "Lighthouse" window, mark "Progressive Web App" and the mobile option, then click on the "Generate report" button.

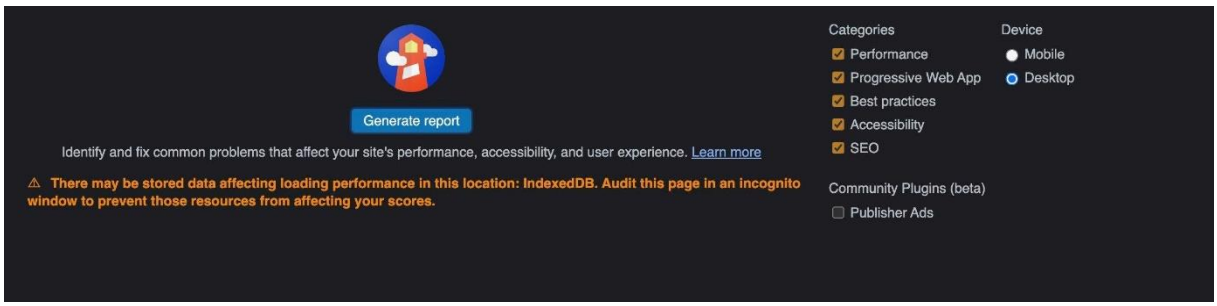


Fig. 3.16. Generating the report for PWA

After pressing the button, the next window will appear for a few seconds, while the tests take place in the background.

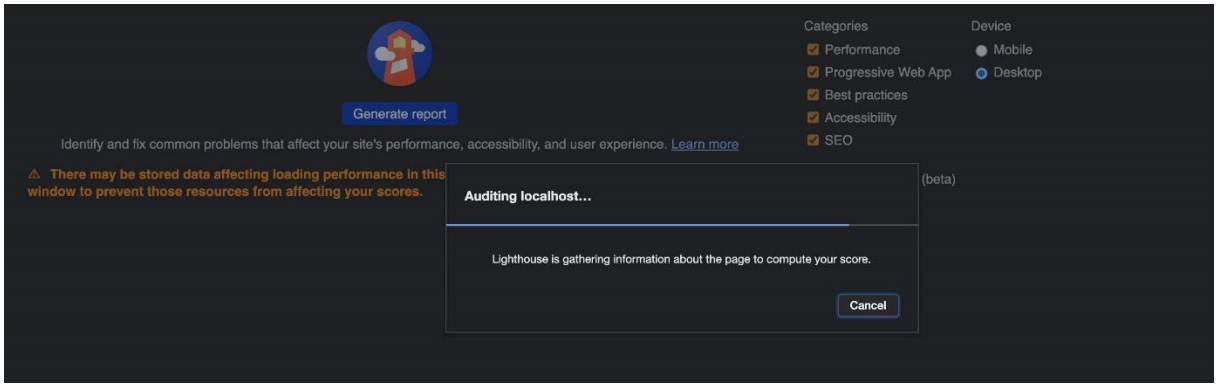


Fig. 3.17. Waiting window

The results of the tests run are as follows:

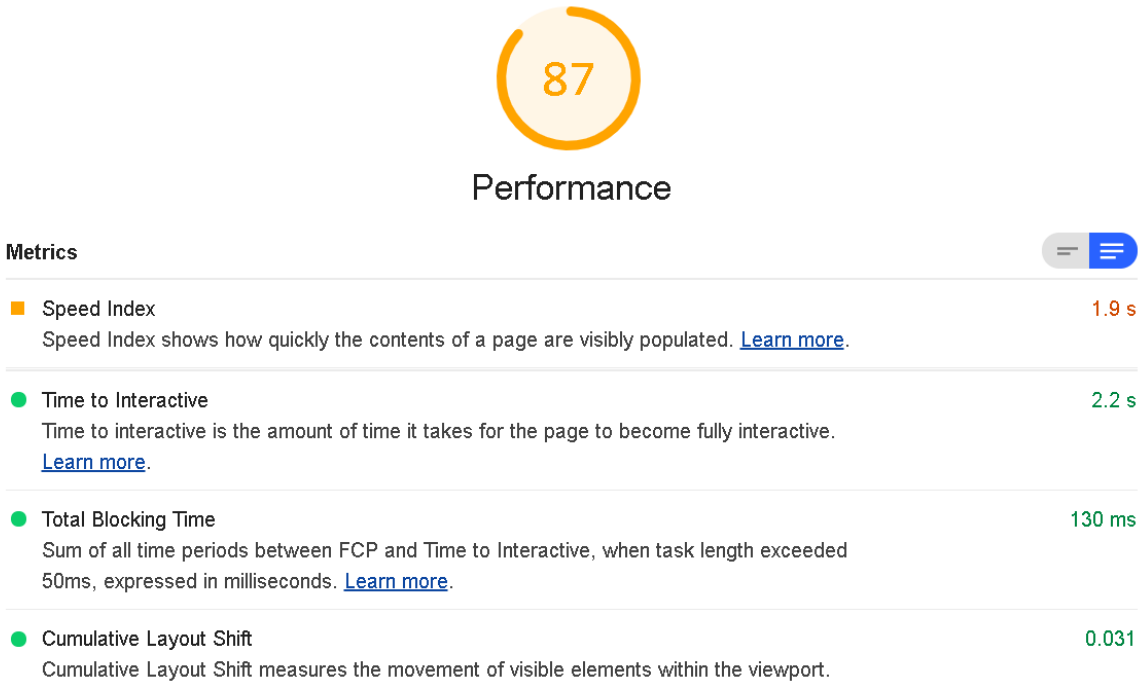


Fig. 3.18. PWA test results

Fig. 3.19. PWA test results

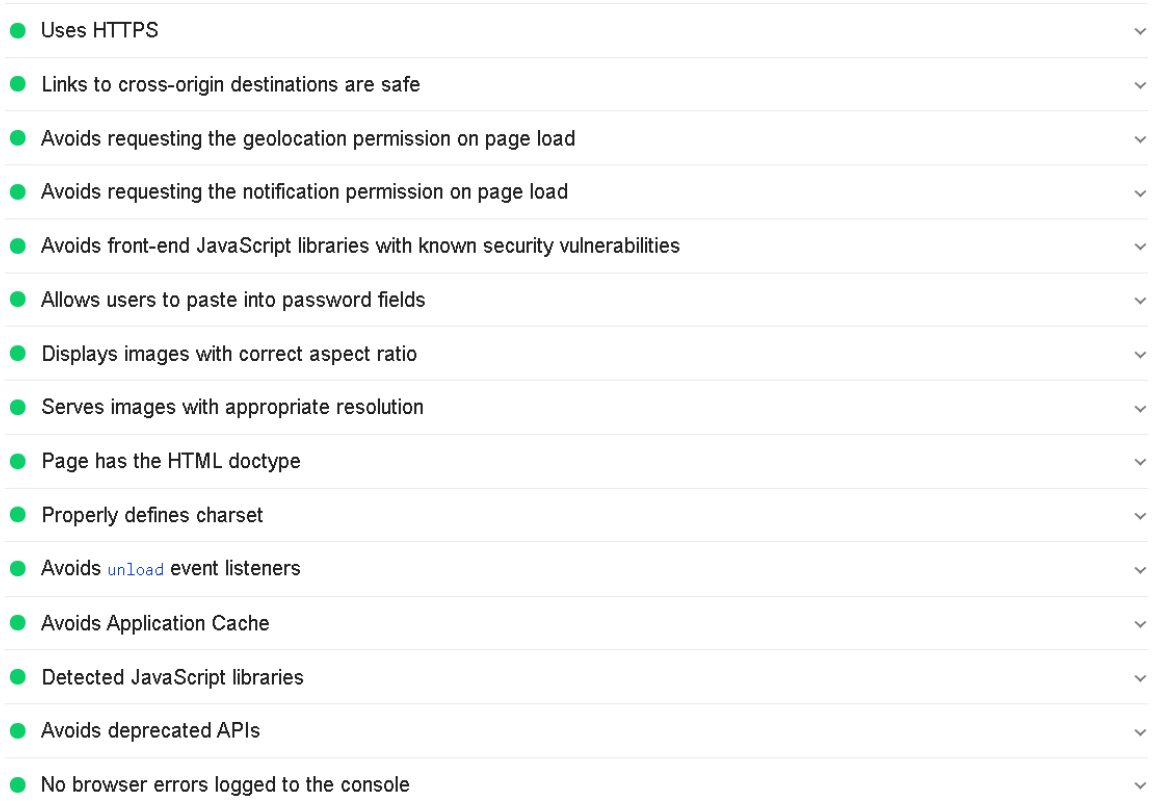


Fig. 3.20. PWA test results

3.2.9 Use of A-frame in the application

In this sub-section, I will describe how I managed to use A-frame to create a 3D model. We will start by adding in the head section of our index.html the following script:

```
<script src="https://aframe.io/releases/1.0.4/aframe.min.js"></script>
```

Then, we will create a new class for each 3D model. In this example I used the one made for the circle lesson. It will render a scene where we can find all the entities created. So, our scene will actually be what we can see on camera.

We use one of A-basic Frame's primitives, <a-entity>, to connect 3D things to our <a-scene>. We can use <a-entity> just like any other HTML element, creating the tag and customizing it using HTML attributes. <a-cylinder>, <a-plane>, and <a-sphere> are some of the other primitives included with A-Frame.

Using the right-handed coordinate system, we may transform an entity in 3D. The positive X-axis extends right, the positive Y-axis extends up, and the positive Z-axis extends out of the screen towards us with the default camera direction.

The distance and rotational units of A-Frame are in meters and degrees, respectively, however they will be internally changed to radians when passing to three.js. We may alter the position, rotation, and scale components to translate, rotate, and scale the box.

I chose the 3D model for circle to show the properties. Since we are working with geometry shapes, I used the geometry component that provides basic shapes for an <entity>.

In order to get a circle that is going to help me for this model, I used the primitive geometric shape of a ring. The geometry primitive ring, produces a flat ring, similar to a CD. Because the ring is flat, A-frame will only draw one face of the ring unless we specify side: double.

The properties for the ring are:

- radiusInner – sets the inner radius of the ring;
- radiusOuter – sets the outer radius of the ring;
- segmentsTheta – represents the number of segments. The bigger the number the rounder the ring will be;
- segmentsPhi – represents the number of triangles within each face defined by segmentsTheta;
- thetaStart – represents the starting angle in degrees;
- thetaLength – represents the central angle in degrees;

```
import React from "react";

export default class Circle extends React.Component {
    render() {
        return (
            <a-scene embedded asrjs='sourceType: camera;'>
                <a-entity
                    geometry="primitive: ring; radiusInner: 0.9; radiusOuter: 1"
                    rotation="90 0 0"
                />
                <a-entity position="-1 1 1" text="value: radius 5cm"/>
                <a-entity position="0 0 0" text="Compute the diameter of the circle"/>
                <a-marker-camera preset='hiro'/>
            </a-scene>
        )
    }
}
```

The first entity in our scene represents the ring with a small inner and outer radius, so the 3D model will look like a circle. I also used the property of rotation in order to put it in a better angle. Then, the second entity will tell us the radius of the circle in cm, in order to help the user calculate de diameter. The result will allow the user to complete the assignment by enabling the button which will send you to the homepage and update the status of that assignment. The last HTML tag will give the marker created for the 3D model named hiro. Now the camera will try to find that specific marker in order to render us the model created.

We can create unique markers for our models with the marker generator. The user must first upload their own image and create a pattern file called pattern-marker.patt. Second, he can save the trained marker to his computer. Finally, he might be able to print his marker. It includes a PDF file to make the developer's job easier.

The pattern that was created, can be added to your code by using the <a-marker-camera> entity. To put that specific marker, you have to specify the type and also the url provided by the website where you create the marker. For my application I used the Hiro marker, which is a basic marker recommended to use for A-Frame. In order to render the 3D model, the camera can move while the marker is going to be static, fixed at 0,0,0.

Fig. 3.21. Hiro marker used in my application

In the next figures I will present how a 3D model is displayed from different angles and distances.

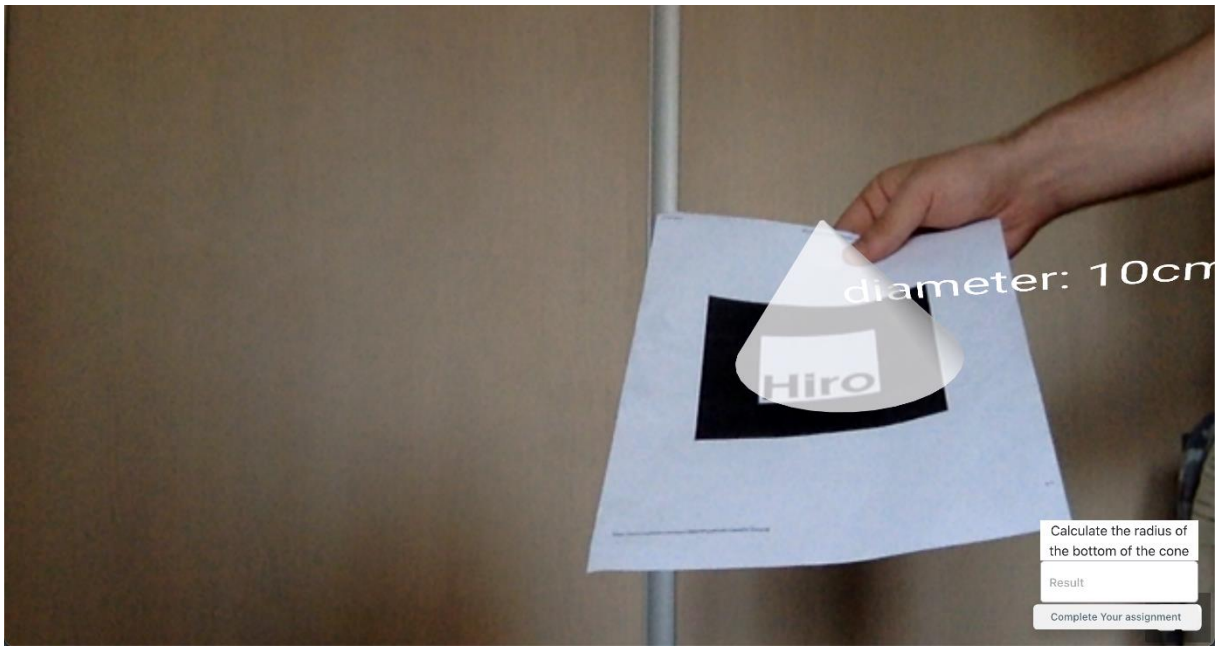


Fig. 3.22. The 3D model for a cone used in my application

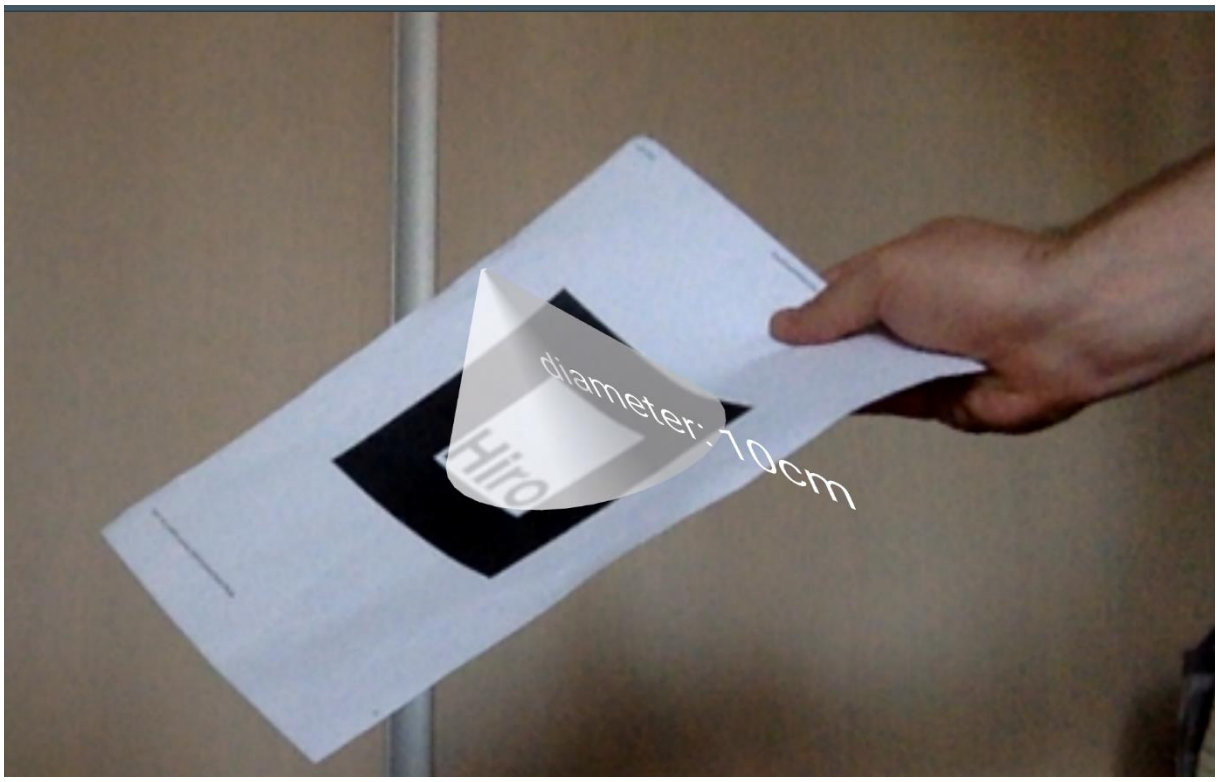


Fig. 3.23. The 3D model for a cone used in my application

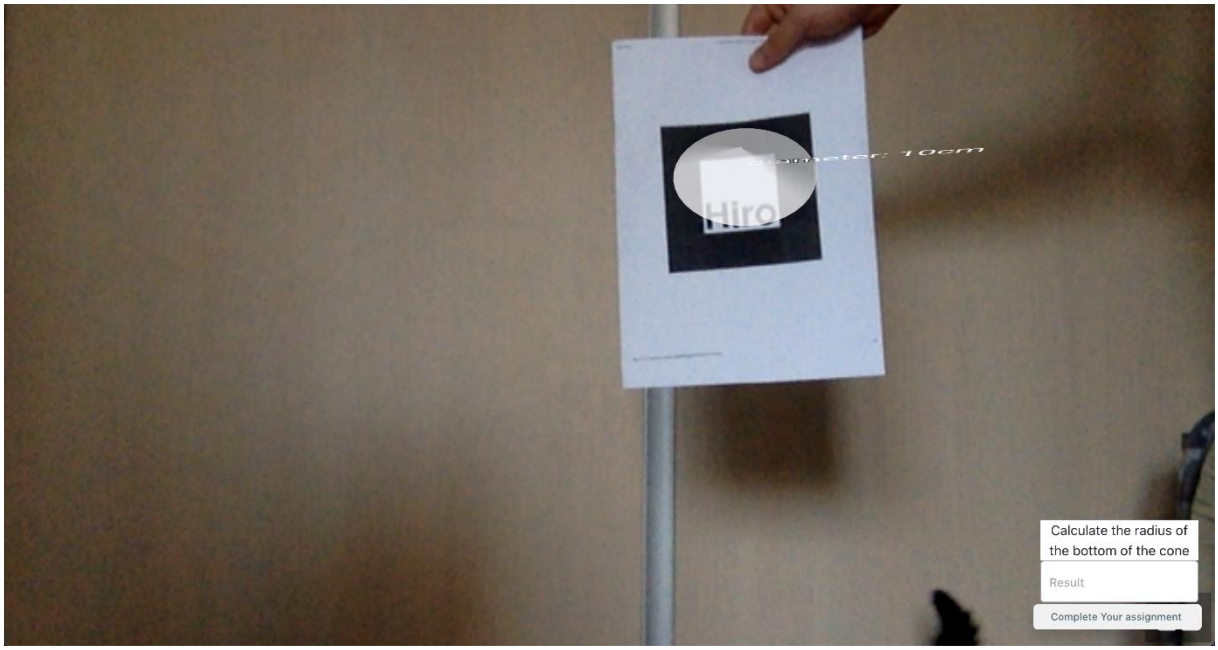


Fig. 3.24. The 3D model for a cone used in my application

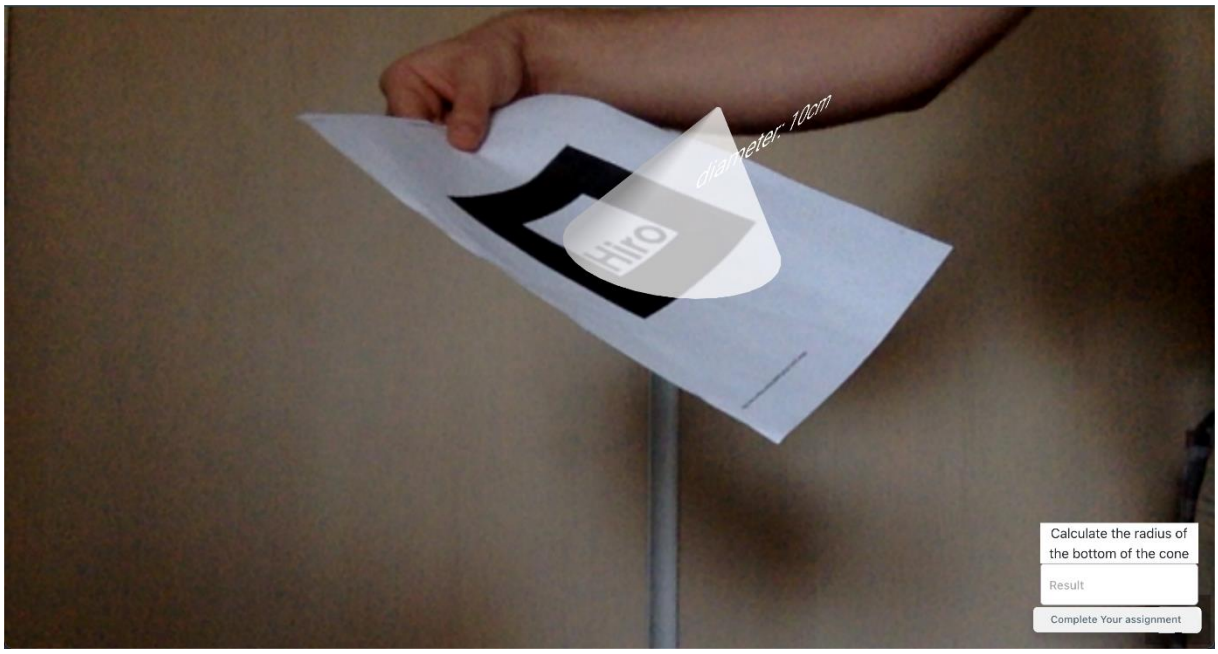


Fig. 3.25. The 3D model for a cone used in my application

3.3 Testing and validation

For the testing and validation I will present you the application. The next screenshots will show the flow of the user on the website.

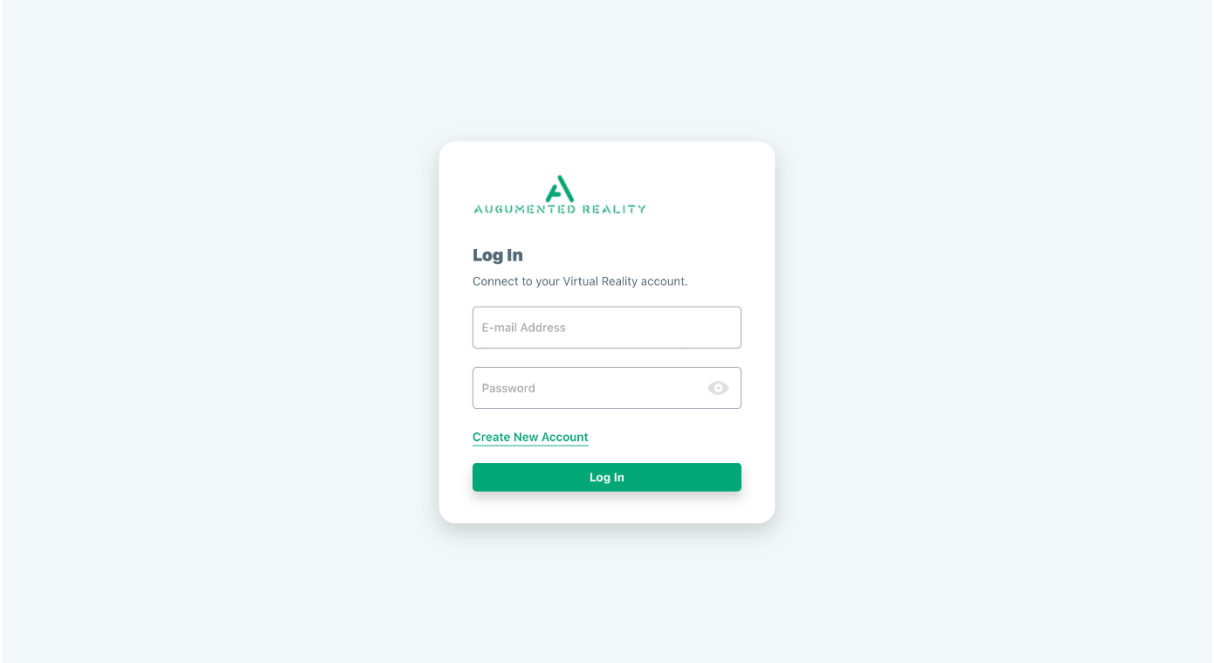


Fig. 3.26. Log In page

On the log in page represented in figure 3.26. the user has to enter the e-mail address and password used for his account and then press the log in button. If the log in was succsesful, the user will be redirected to the homepage. Above the button we can see the „Create new account" link that can redirect the user to the register page.

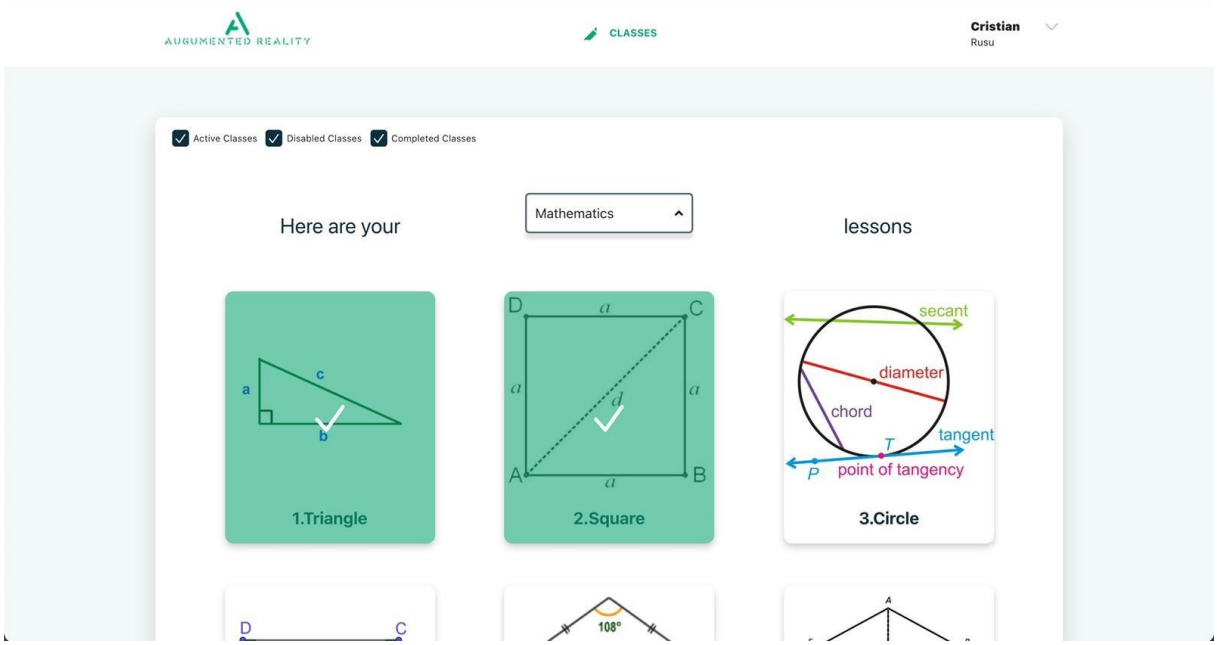


Fig. 3.27. Homepage of the app

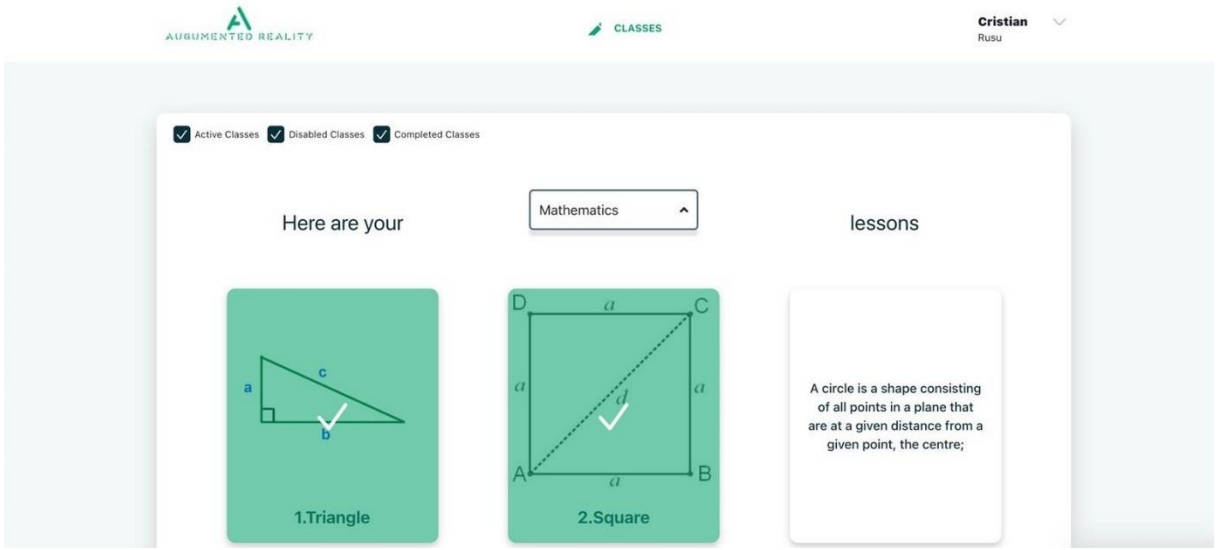


Fig. 3.28. Homepage of the app and hovered over a class

In figures 3.27. and 3.28. we can see that the user is on the homepage, after successfully logging in.

On the homepage, we have the header with the logo that sends you to the homepage if you press it, the Classes page and the first and last name for the user with a dropdown where we have the edit profile page and the log out button.

Next we have the filters for the classes, each of them showing if an assignment is active, completed or disabled.

Next we have the dropdown for each subject, which will present the lessons in cards. Each card represents a lesson, having a representative picture, the title and number for it.

If a card has a check and a green cover, it means that the user finished the assignment, meanwhile if it has an X with a grey cover it means that the assignment were not uploaded yet.

In the case where the assignment is still active, the user can hover over it, flipping the card that displays a short description about the lesson, and then click on it in order to send you to the AR page from figures 3.29, 3.30 and 3.31.

At the bottom of the page, above the footer that displays contact information and social media account, we have the button which will send the user to the quiz page.

Fig. 3.29. The AR page

Fig. 3.30. The AR page

Fig. 3.31. The AR page

When the user gets on this page, the camera will be active, and it will start searching for the marker that is going to display the 3D model. For my application I used the Hiro marker because it is the most precise and recommended for A-frame. On every model, the user will find a small task to complete in order to enable the complete the assignment button, such as how much is the radius of the circle from the bottom of the cone if the diameter of it is 10cm.

After writing the result, and if the result is correct, the user can press the button which will redirect him to the homepage, where the lesson that he entered will turn green and get the check meaning that the assignment was completed. In the same time, the fact that it was completed will get to the Sheets in order for the teacher to see.

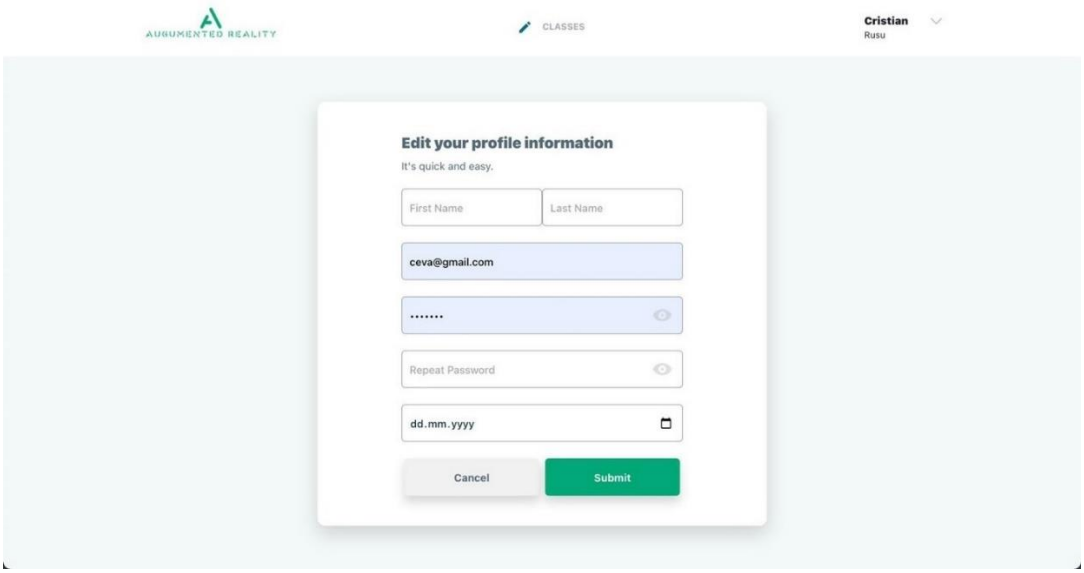


Fig. 3.32. Edit profile page

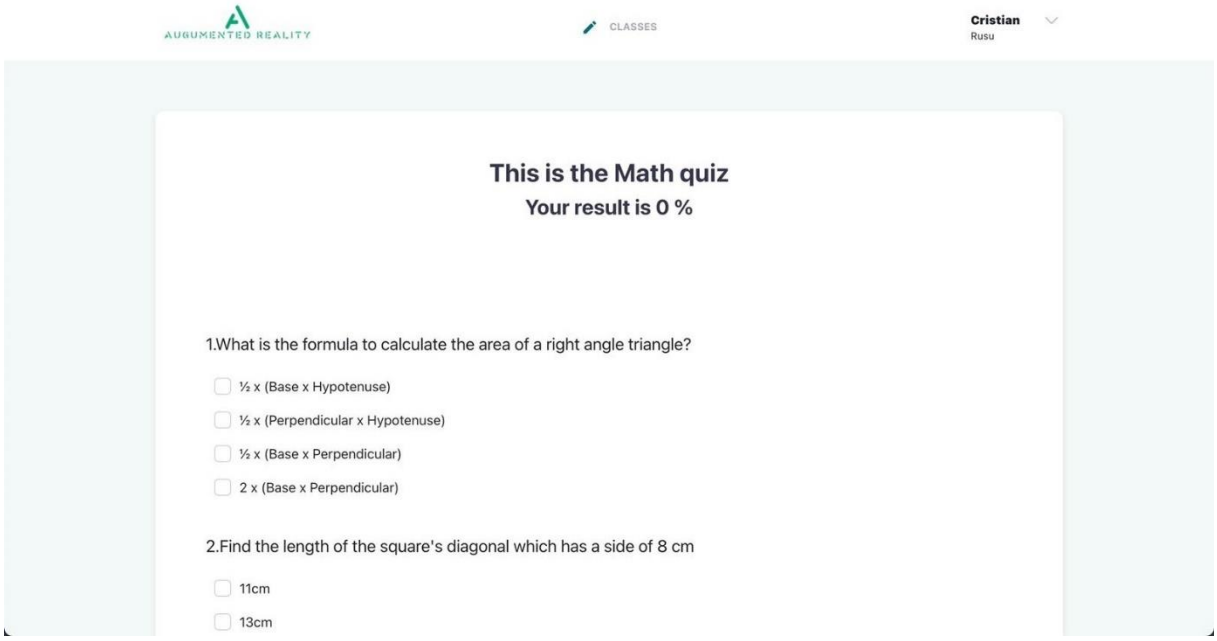


Fig. 3.33. The quiz page

In the figure above we have the quiz page. On this page, every student will have a multiple choice test with questions from each lesson. After the user will check all the questions, on the bottom of the page there is a submit button which will be pressed. By pressing it, the user will be redirected to the top of the page where the result for the quiz will appear. This result will be updated in the Sheet for attendance.

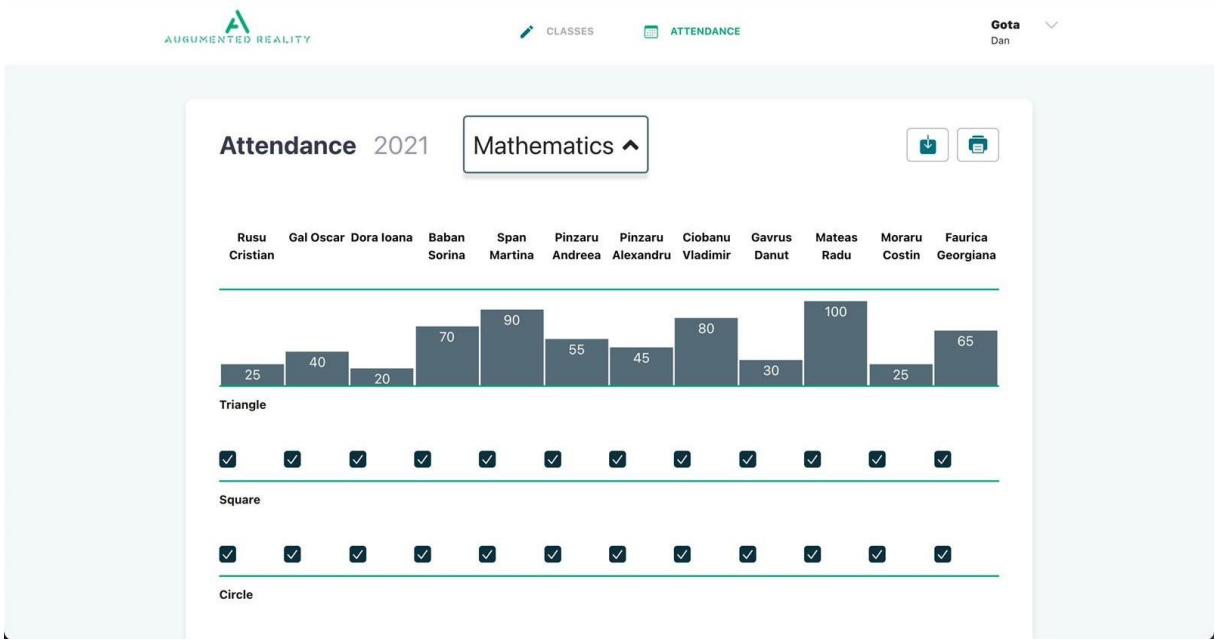


Fig. 3.34. The attendance page

If the user that logged in on the application is a teacher, on the header will appear the Attendance page as seen in the figure 3.34. There, the teacher can see for each student the quiz result displayed as a chart and a checked box or unchecked box for each lesson showing if the assignment was completed or not. Those things are displayed in a column under the student's full name.

4 Conclusion

4.1 Conclusion

When we think of education, the first thing that comes to mind is obtaining information. Education is a tool that provides individuals with knowledge, skill, technique, and information, as well as allowing them to understand their rights and responsibilities toward their family, society, and country. It broadens one's perspective and attitude on life.

We gain knowledge of the world around us through education. It cultivates in us a way of looking at life. It is the most crucial aspect of a person's development. One cannot explore new ideas without education. It implies that someone will be unable to grow the world because without ideas, there is no creativity, and without that there is no development overall.

With my application, I want to help improving the education system, giving a tool for both students and teachers to mantain a unique interaction. Using it, students will manage to get a better representation for each lesson and teachers can organize the information and control the attendance for each student, as well as quizzes for their subject.

The progressive web application was easily made through the following technologies: React.js for frontend and Google Sheets, Firebase for backend. The choice of a NoSQL database led to the simplification of the data structure. The main functionality of this website, the use of Augmented Reality to visually represent the lessons through 3D models, has been implemented through the A-Frame library, which has full support for most existing browsers. At the same time, the UX is intuitive, so the user should not encounter navigation problems.

Augmented reality has the potential to alter the way we interact with technology. The unthinkable becomes achievable with augmented reality, and its educational potential is only beginning. Augmented reality interfaces allow users to engage with the real and virtual worlds in a seamless manner. Learners engage naturally with 3D information, objects, and events using augmented reality technologies.

4.2 Future improvements

The prototype is in an initial stage, there is room for improvement. Thus, weaknesses can become strengths, the solutions being easy to find. In the future, based on this prototype can be added a wide range of extra features.

Because it is a web application, we can improve this by making it compatible for mobiles and tablets as well. This will make it more accessible and easier to use the AR feature.

Another huge update would be to make it work at a bigger scale, and also to make it more customizable for any school. This will require support for different languages as well.

Lastly, another great improvement would be for the teacher to create their own versions of 3D models and also to mark attendances for students and grade them separatly from the quiz.

5 Bibliography

[1] Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H.F. The world-wide web. Communications of the ACM, 37(8), 1994, pp.76-82.

[2] Kulkarni, Amogh, Jaison Salu John, Yohan John Thampi, Shravan Udaykumar, Gaurav Prasad, and Vrinda Halarnkar. "Web Browsers." National Institute of Technology Karnataka Surakhtal (2009).

[3] Raggett, Dave, Arnaud Le Hors, and Ian Jacobs. "HTML 4.01 Specification." W3C recommendation 24 (1999).

[4] Håkon Wium Lie. "Cascading Style Sheets." University of Oslo, Norway (2005)

[5] M. DiPierro, "The Rise of JavaScript," in Computing in Science & Engineering, vol. 20, no. 1, January/February 2018, pp. 9-10.

[6] Gackenheimer, Cory. Introduction to React. Apress, 2015.

[7] Varghese, Blesson. "A history of the cloud." ITNOW 61.2 (2019): 46-48.

[8] "What Is Firebase? The Complete Story, Abridged.". Medium, 2021, https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0.

[9] "What Is Google Sheets And How Is It Used?". Whatis.Com, 2021, https://whatis.techtarget.com/definition/Google-Spreadsheets.

[10] A. Sterling, "NodeJS and Angular Tools for JSON-LD," 2019 IEEE 13th International Conference on Semantic Computing (ICSC), 2019, pp. 392-395

[11] Loisier, Marie. "A Brief History Of Augmented Reality". Numerized, 2021, https://numerized.com/augmented-reality/history-augmented-reality/.

[12] H. M. Abdullah and A. M. Zeki, "Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example," 2014 3rd International Conference on Advanced Computer Science Applications and Technologies, 2014, pp. 85-89

[13] "What Is UI Vs. UX Design? What'S The Difference?". *Medium*, 2021, https://uxplanet.org/what-is-ui-vs-ux-design-and-the-difference-d9113f6612de.

[14] Oluwatosin, Haroon Shakirat. "Client-server model." IOSRJ Comput. Eng 16.1 (2014): 2278-8727.

[15] Thakur, Parbat. "Evaluation and Implementation of Progressive Web Application." (2018).

[16] Masse, Mark. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. " O'Reilly Media, Inc.", 2011.

[17] Larsen, Rob. Beginning HTML and CSS. John Wiley & Sons, 2013.

[18] Jensen, Simon Holm, Anders Møller, and Peter Thiemann. "Type analysis for JavaScript." In International Static Analysis Symposium, pp. 238-255. Springer, Berlin, Heidelberg, 2009.

[19] Bugl, Daniel. Learn React Hooks: Build and refactor modern React. js applications using Hooks. Packt Publishing Ltd, 2019.

[20] Moroney, Laurence. "Using authentication in firebase." In The Definitive Guide to Firebase, pp. 25-50. Apress, Berkeley, CA, 2017.

[21] Hajian, Majid. "Deploying to Firebase as the Back End." In Progressive Web Apps with Angular, pp. 9-27. Apress, Berkeley, CA, 2019.

[22] Carmigniani, Julie, and Borko Furht. "Augmented reality: an overview." Handbook of augmented reality, 2011, pp. 3-46.

[23] Kesim, Mehmet, and Yasin Ozarslan. "Augmented reality in education: current technologies and the potential for education." Procedia-social and behavioral sciences, 2012, pp. 297-302.

[24] Lee, Kangdon. "Augmented reality in education and training." TechTrends 56, no. 2, 2012, pp. 13-21.

[25] Jung, Timothy, and M. Cluaudia tom Dieck. "Augmented Reality and Virtual Reality." Empowering Human, Place and Business. Cham: Springer International Publishing, 2018.

[26] S. G. Santos and J. C. S. Cardoso, "Web-based Virtual Reality with A-Frame," 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), 2019, pp. 1-2.

[27] Fowler, Martin. UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional, 2004.

[28] V. Vachharajani, J. Pareek and S. Gulabani, "Effective Label Matching for Automatic Evaluation of Use -- Case Diagrams," 2012 IEEE Fourth International Conference on Technology for Education, 2012, pp. 172-175.

[29] Xiaoshan Li, Zhiming Liu and H. Jifeng, "A formal semantics of UML sequence diagram," 2004 Australian Software Engineering Conference. Proceedings., 2004, pp. 168-177.

[30] "Cloud Firestore Data Model | Firebase". Firebase, 2021, https://firebase.google.com/docs/firestore/data-model#:~:text=Cloud%20Firestore%20is%20a%20NoSQL,set%20of%20key%2Dvalue%20pairs.

[31] "Tabletop". Npm, 2021, https://www.npmjs.com/package/tabletop.

[32] Freeman, Adam. "Creating a React App." In Essential TypeScript, pp. 459-482. Apress, Berkeley, CA, 2019.

[33] Zlatanova, Sikya. "Augmented reality technology." GISt Report No. 17, Delft, 2002, 72 p.

[34] Chen, Yunqiang, Qing Wang, Hong Chen, Xiaoyu Song, Hui Tang, and Mengxiao Tian. "An overview of augmented reality technology." In Journal of Physics: Conference Series, vol. 1237, no. 2, p. 022082. IOP Publishing, 2019.

[35] Caudell, Thomas P. "Introduction to augmented and virtual reality." In Telemanipulator and Telepresence Technologies, vol. 2351, 2018, pp. 272-281.

[36] Yuan, Yu. "Changing the world with virtual\/augmented reality technologies." IEEE Consumer Electronics Magazine 6, no. 1, 2016, pp. 40-41.

[37] R. Aggarwal and A. Singhal, "Augmented Reality and its effect on our life," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2019, pp. 510-515