Expertise
and insight
for the future

Oskari Tamminen

Designing and building a dynamic learning platform for CS: GO players

Metropolia University of Applied Sciences

Bachelor of Engineering

Mobile Solutions

Bachelor's Thesis

5 November 2020

Author	Oskari Tamminen
Title	Designing and building a dynamic learning platform for CS: GO players
Number of Pages	33 pages + 1 appendix
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Professional Major	Mobile Solutions
Instructor	Kari Aaltonen, Principal Lecturer

The purpose of this thesis was to research and develop a dynamic and video-based learning platform. The target group of this application is Counter Strike: Global Offensive players. The platform will focus on one specific element of the game thus limiting the complexity of it.

The purpose of this project was to research possible users and other similar platforms and with that information, build an application. The results of user interview showed that over half of the users will be using the site on desktop and thus the application format of web application was chosen because it can be used to reach both mobile and desktop users. The user interface of the application was created with React JavaScript library. The backend of the application was done using Google's Firebase application development platform.

The end result is a working application that uses continuous deployment. The user experience was fine-tuned by conducting usability tests for target users. The application is not publishable at this point due to some security flaws.

Keywords	Web, web-app, UX, usability, Firebase, React, CS: GO

Tekijä Otsikko Sivumäärä	Oskari Tamminen Designing and building a dynamic learning platform for CS: GO players 33 sivua + 1 liite
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Mobile Solutions
Ohjaaja	Yliopettaja Kari Aaltonen

Insinöörityön tarkoituksena oli tutkia kohderyhmää sekä kehittää heille dynaaminen ja videopohjainen oppimisalusta. Kohderyhmänä toimii pelin Counter Strike: Global Offensive pelaajat. Oppimisalusta keskittyy vain yhteen pelin elementtiin ja näin rajoittaa hieman applikaation monimutkaisuutta.

Tarkoitus oli tutkia mahdollisia käyttäjiä sekä muita vastaavia alustoja ja luoda tämän perusteella dynaaminen verkkosovellus. Käyttäjätutkimuksesta selvisi, että yli puolet aikovat käyttää sovellusta tietokoneella ja näin sovellusmuodoksi valittiin verkkosovellus, sillä se toimii niin puhelimessa kuin tietokoneella. Verkkosovelluksen käyttöliittymä tehtiin käyttäen React JavaScript kirjastoa. Backend tehtiin Googlen Firebase-sovelluskehitysalustaa käyttäen.

Lopputuloksena saatiin aikaan toimiva sovellus, joka toimii jatkuvalla julkaisulla. Sovelluksen käyttökokemus saatiin hiottua hyvälle mallille suorittamalla kohdekäyttäjille käytettävyystestejä. Tässä vaiheessa sovellus ei vielä ole julkaisukelpoinen, sillä muutamia turvallisuuteen liittyviä asioita puuttuu.

Avainsanat	Verkkosovellus, käyttökokemus, käytettävyys, Firebase, React, CS: GO

Contents

List of Abbreviations

HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
CS: GO	Counter Strike: Global Offensive
JSON	JavaScript Object Notation
UI	User Interface
UX	User Experience
URL	Uniform Resource Locator
PWA	Progressive web application
API	Application programming interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Sercure
iOS	Apple hardware exclusive mobile operating system created by Apple
Android	Mobile operation system created by Google
Sharding	The process of breaking up large database tables into smaller chunks
Repository	Storage location for software packages
FFMPEG	A cross-platform solution for manipulating audio and video
XML	Extensible markup language

JSX	JavaScript XML
DOM	Document Object Model
SSL	Secure Sockets Layer
reCAPTCHA	A security service that protects websites from spam and abuse
OS	Operation system
PC	Personal computer
DOM	Document object model

1 Introduction

The goal of this study is to research and implement a web application that would work as a learning platform for Counter Strike: Global Offensive players. The research will be conducted by interviewing and studying some players ranging from total beginners to an almost professional level of players. The players' insight and ways of learning new techniques help to narrow down the application to a level that matter how skilled you are, you would still find some use of this application.

CS: GO is a game where two teams compete against each other taking turns attacking and defending. The aim for the attacking team is to plant a bomb on a marked spot while the defenders try to stop them. The bomb has a 40 second counter before exploding. During that time the defending team may attempt to defuse the bomb and thus winning the round. If either team is eliminated and bomb is not planted, that team loses.

There is one gameplay element of Counter Strike: Global Offensive that will be the core of this application. It is the ability to throw different types of grenades or utilities in-game. It is a major component of the game at its competitive level. The most tactical utility is probably the smoke grenade (Figure 1). It is used to distract, to hide your movement and to block your opponent's vision. With a bit of practice, you can learn to throw these from cover to precise locations. This application will work as a library for tactical utility throws and other tactical elements.

Figure 1. An in-game image of a utility (smoke grenade) being used. Here the defending team is using the smoke to cover their movement while they try to defuse the bomb.

The solution of the platform Holysmokes is to provide a tool for the users, so that they can create and provide the content for each other. This way the application would work dynamically, and the creator would not have to create all the content. This approach would mean that the platform would change overtime just as the game changes. As the developer and not a professional player, it would be impossible to know each and every quirk and feature of the game, and that is why it is better to leave it to the hands of the community.

The learning platform is aiming to be a simple and easy to use web application. The technology behind the application would be a frontend created with a React.js combined with a Google Firebase backend. The marketing strategy is to spread information about the application on social media platforms and generate ad-based income with ads.

2 Web applications

A web application is a software that is running on a server and can be accessed using a browser. The part of the application that gets delivered to the user consists of three things. These are HTML, CSS and JavaScript. All major browsers support these languages. HTML marks up the order of the visible elements while CSS styles them. JavaScript is the part that makes a website powerful and its content interactive.

2.1 Static web application

A static web application is the simplest form of a website. When a user enters a URL for a static website in his browser, the server responds by sending an HTML file just as it is stored. The file is not altered in any means, but it can have styling provided with CSS. JavaScript can also be used in a static website, but its functionality must be run on the client side. A static website is usually hosted on a server that does not provide the function of running server-side code.

2.2 Dynamic web application

A modern dynamic web application is a one paged application, that works in the user's browser and it is usually made using some kind of JavaScript framework. A one paged application means that when the site is browsed, the browser dynamically changes and edits things that are happening instead of redirecting to a new page like a traditional web page. Because a dynamic web application works in a browser, it does not need to be downloaded separately. Dynamic web applications can be accessed using a browser that meets a modern browsers criterion. Examples of such browsers are Google Chrome, Mozilla Firefox, Safari and Microsoft Edge.

A dynamic web application does not require any device specific hardware. A web application can be made to look like a native application so that a user familiarizes with it. Even if the web application looks like a native one, it responds more slowly to user actions than a native application thus feeling foreign to the user. The slowness of the application is due to the fact that it is accessed through a browser and has to download all

the functionality and data from the web. However, modern browsers make this a bit easier, as they store part of the site in the browsers cache. [1.]

2.3 Progressive web application

A progressive web application is a mixture of a traditional mobile application and a web application. Even though progressive web applications area fairly new thing, the technology behind them is essentially the same as in web applications. A PWA can be accessed through a browser and it does not need to be installed. If you have a PWA capable device such as an Android phone, the website will prompt you to install the application to your home screen. A PWA must meet a few criteria based on user experience in order for you to be able to install it on your phone. These criteria include:

- The site must use HTTPS-protocol to transfer data.
- The site must be responsive to any screen size.
- All of the applications URLs must be available in offline mode.
- A user must be able to install it to his home screen.
- The application needs to load fast. There is a 10 second time limit for using a 3G connection.
- The application must work on all major browsers (Chrome, Edge, Firefox, Safari)
- The user experience must be fast no matter how slow the connection is.

These criteria can be checked with the help of Lighthouse. It is a tool for evaluating PWAs. [2.]

The only thing slowing down the growth of PWAs is that big companies such as Apple have not joined in on it. This hinders companies to develop PWAs since they would have to make a separate application for Apples iOS if they want their application to be installed on the home screen. Apples reason for this is strictly economic. They would lose a lot of

money if people installed their applications straight from the browser instead of their own digital distribution platform App Store. As long as a company the size of Apple ignores this type of application software, it has an inevitable obstacle on its development path. [3.]

3 Technologies used

There were a couple of good JavaScript framework options to make the frontend of the application. Vue.js and React.js seemed both good options to build a PWA. The decision to choose React.js as the frontend solution was that there is a huge community behind React, much larger than behind Vue. The React community is a huge help when starting out or when encountering a technical problem while developing. Firebase was the backend choice because it serves all the functionality that was needed. The main function of the backend is to handle authentication, database and hosting the application.

3.1 ReactJS Frontend

React.js is Facebooks JavaScript library that is designed to create interactive UIs. It can be used to develop web- or mobile applications. At first React is quite overwhelming, because it has so many different parts and new things such as components, props, JSX, elements, virtual DOM, React hooks and more. Consequently, it is quite hard comprehend what React actually is.

React is technically not a framework because it does not rely on an ecosystem of tools. This means that React can be used on top of other frameworks such as Angular. React can be used to create a complete application or you could just place some React functionality to your non-React web app. This is achieved by React components, which are independent and reusable JavaScript code that return HTML with a render function (Listing 1). [4.]

```
class Bubble extends React.Component() {
      render() {
            return <p> I am a bubble! </p>
      }
}
```

Listing 1. An example of creating a simple React component called Bubble.

React can place this component in the HTML through DOM (Figure 2). DOM is an API that can be searched and manipulated with JavaScript. DOM is meant to define how elements in the document share data to each other. It also defines how these elements can be referred to.

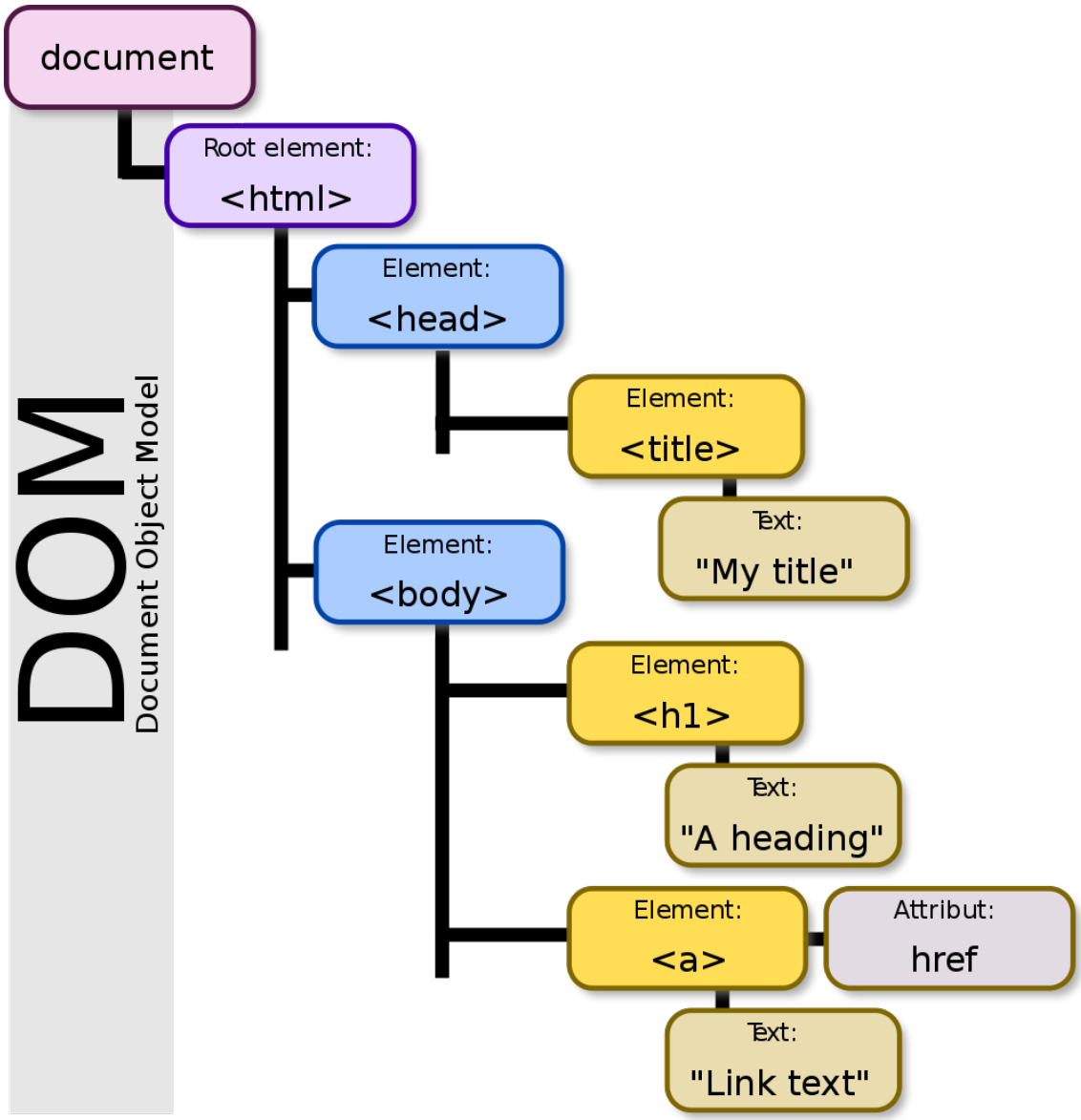


Figure 2. An overview how the DOM looks like. [5.]

React does not directly manage the DOM. It creates its own version of it called the React virtual DOM, which is not the actual DOM. That is why components are rendered to the ReactDOM (Listing 2). React compares the virtual DOM to the actual DOM every time a user interacts with a component. If React notices no differences, it does not do anything. If differences are found, React replaces the DOM with the virtual version thus updating the UI. [5.]

```
ReactDOM.render(<Bubble />, document.getElementById('root'));
```

Listing 2. This renders the Bubble component into React's virtual DOM

React components have a built-in state object. State is an object where you can store different values that belong to that component. A state can be updated by calling a “setState” function which updates the state asynchronously. When a component notices a change in its state, it re-renders itself meaning that the output will match the new value. An example of this could be a logged in users name that is displayed in the component. The component is same for all users, but the state is different.

3.2 Firebase Backend

Firebase is a backend solution which was eventually chosen for this project. It is a backend server designed for Android, iOS and mobile web users. The most recent addition to firebase is that it supports Unity projects as well. Firebase is an engaging backend solution that covers the need for our database, storage and hosting problems. The Firebase console filled with services that helps to build, improve and grow your application (Figure 3). Holysmokes is an application that needs to use many of these, and the aim here is to focus on the ones that are most vital.

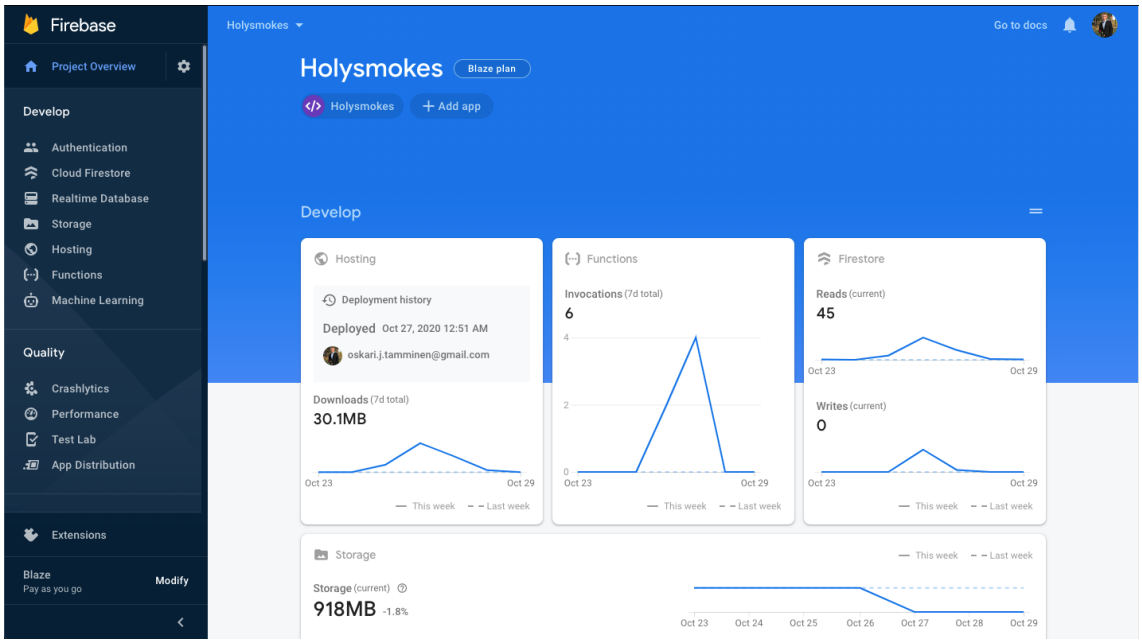


Figure 3. Overview of the Firebase console of an individual project

3.2.1 Authentication

Authentication will be handled by Firebase Authentication which integrates with the other services. Anonymous users are part of our application but there is some functionality like uploading content which is restricted to having a confirmed identity of the user. This content will then be reviewed by an admin user that will be configured in Firebase Authentication which supports role-based user permissions.

3.2.2 Storage

Firebase Cloud Storage is a core part of our application. It will store and serve all the content that is provided to the user. The upload process is robust which means that if something disturbs the upload process, the upload will resume from where it stopped. With Firebase security rules access can be granted or access to some areas of Cloud Storage to different user groups can be restricted. Firebase Cloud Storage is also infinitely scalable so there is no way it would perform slowly or run out of space. [6.]

3.2.3 Database

There are two database options in Firebase. The options are Realtime Database and Cloud Firestore. They are both NoSQL databases designed for mobile use and they both offer client-first SDKs, real-time updates and they are free to use up to a certain point.

Realtime Database is the original database that Firebase has had since the beginning of its time. Realtime Databases stores its data in JSON trees, does not offer local data storage for web clients and requires more complex queries with limited sorting. Realtime Database scales poorly as it requires sharding the data across multiple databases. It is mainly used for simple data.

Cloud Firestore is the successor to the Realtime Database and it is used in the backend since it offers so much more for this application. It stores data in JSON like documents arranged in collections which are easy to read. It offers better offline support for web applications which is important for this application if it is going to be transformed into a PWA at some point. Cloud Firestore also offers much more complex querying which is helpful when building a tool for the user to browse and filter through content and it scales

automatically when it detects its limits. The only downside in Cloud Firestore is that it has limits on the write rates to individual documents. Luckily, there are only a few occasions of needing to modify individual documents in our application. [7.]

3.2.4 Cloud Functions

Cloud Functions is a serverless environment where you can connect your Firebase services. Cloud Functions are snippets of custom code that can be triggered with events that are happening in the cloud infrastructure. Cloud Functions can be written in a variety of programming languages such as:

- Java
- Python
- Go
- Node.js

Our application uses Node.js environment to run Cloud Functions. We have need for a couple of custom functions. One function is triggered when a user uploads a clip, creating a thumbnail image in the storage with the help of FFMPEG. The other function is deleting a clip's Firestore document and thumbnail image after deleting a video file from the storage.

3.2.5 Hosting

Firebase Hosting is a hosting service for which provides content delivery for web applications. It is a secure way to deliver and retrieve data to the user with its built in zero-configuration SSL. The delivery of assets is backed by an SSD storage and a global CDN, which means that wherever the user may be, the content is delivered quickly. Firebase Hosting also provides you with a sub-domain which can be changed to a custom one if needed. You can also set up workflows with Travis or GitHub integration. [8.]

3.2.6 Pricing plans

Firebase features two pricing plans. A free one called Spark and a pay as you go plan called Blaze. There is one huge difference in these two plans which is why we can't use the free one. It is because the paid plan has the ability to use a Node.js 10 runtime with our Cloud Functions. Firebase will charge money from stored data in the Blaze pricing plan and an admin will not have the time to review the potential rate of unlimited videos provided by the users. This is why we need to add limitations to how many unreviewed video clips a user can have at a time, so that no user can cause intentional economic harm by uploading tons of useless video clips.

3.3 GitHub

GitHub is used for the project's version control. GitHub started in 2008 and it is a software development platform designed to host code. GitHub is nowadays owned by Microsoft and there are over 40 million users on the platform. [9, 10] GitHub supports version control, project management and task automation which makes it perfect for our application's version control platform. When being done with a specific task, that version of the application can be stored on GitHub to its own repository and it can be retrieved at any time.

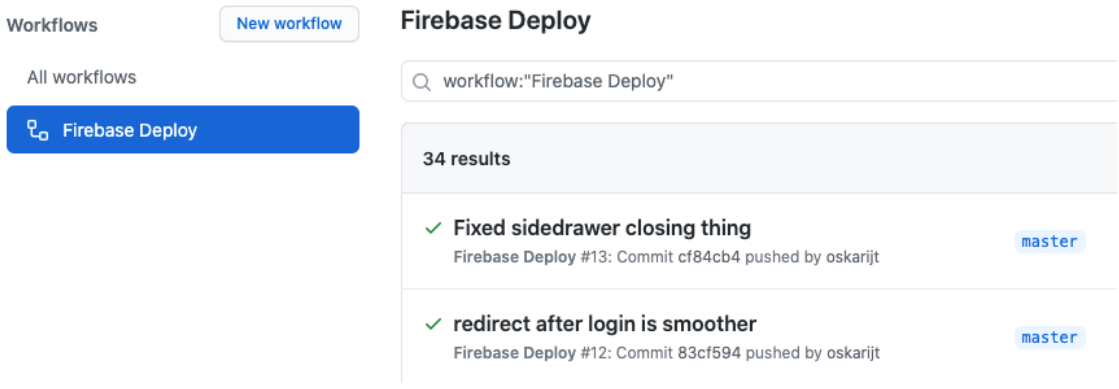


Figure 4. Commits on the master branch are deployed to Firebase after they run testing and building processes.

GitHub Actions is a feature that you can enable in your repository. It is a feature that performs automated tasks based on what the developer needs. This project uses it for continuous deployment. In our use case, when the code is pushed on a specific branch

in the repository, GitHub Actions will perform automated tasks such as run tests and build the application (Figure 4). After completing these tasks, it will deploy the build to Firebase Hosting.

4 Benchmarking other online video platforms

Benchmarking is used to compare performance of different things. Benchmarking is performed to gain new insights and to improve processes. In this benchmarking test the aim is to compare the different processes other online video platforms have. The main things we are looking for are things like landing page, authorization, upload process and the video feed. In this benchmark we compare 3 different video platforms where one is a competitor of ours.

- Youtube.com
- CSGONades.com
- Vimeo.com

The only page where interaction with videos can be made as soon as the page is loaded is CSGO Nades. CSGO Nades is a similar platform to the platform we are creating. YouTube comes second wanting to interact with a video. It only prompts you to sign in, but it does not force it, and with one click you are already browsing. Vimeo has the dullest landing page of them all. It does not show any video thumbnails and only shows pricing plans for content creators. I did not find a way to browse videos from the landing page in Vimeo without logging in.

Authorization is something that all of these provided. In Vimeo it is very highly required as you have no means to browse anything without authorizing. YouTube prompts you to do it but there is no requirement to do it if you are only watching videos and not planning to interact with them. CSGO Nades does not prompt or require authorization if you are not going to upload videos. The only problem with CSGO Nades authorization is the fact that you have to sign in with your Steam credentials. Steam is the platform where you can buy games and where you have your game library as well as your in-game inventory. It is well known that some sites use a fake Steam link to fish for your credentials so that they can steal inventory items from you. Some collectors in game inventory can have a value of over a thousand euros or more and this is why Steam login may scare away some users.

The upload process requires authorization in all of these platforms. YouTube and Vimeo have somewhat similar upload processes. They require the same additional information to be submitted with the video such as:

- Title
- Description
- Maturity rating
- Tags
- Language
- Category selects
- Private or Public video

CSGO Nades upload process is very different. It has some of the same options as Holysmokes but there is no direct video uploading. What you have to do instead is upload the video on another platform and then link the video URL to CSGO Nades upload form. This means that we have to authorize on two platforms to upload a video and this makes the process very complicated. This also means that their site is not hosting the videos. CSGO Nades is also missing thumbnail generation. You have to provide the thumbnail yourself in the upload form. This is where our platform Holysmokes has an advantage, because all videos are uploaded to and hosted by our Firebase backend where the thumbnail for the video is also generated.

This part of the benchmarking has the least amount of differences. The video feed is something where all of these platforms share the same UI principles. All of them show a gallery of thumbnails with the titles of the video and some additional information such as view count. This UI principle is something that cannot be strayed away from in order for a user to be familiar with it from the first moment he lands on the page.

5 Design & Usability

The application is designed on the fly. There is no clear shell how the application should look like. When building a feature such as registering a user, the design around that will focus on the stuff that the interface needs. There are a few major features that are going to be provided for users. All of these features revolve around the clips of techniques that the application consists of. These are the main features:

- To browse and find new techniques
- To share techniques with others
- To be able to provide your own techniques
- To be able to search techniques

The goal is to make the application as approachable as possible thus allowing anonymous users to browse through content. If a user wants to provide content, he needs to make a user and confirm his email. To prevent mischief, all provided content will be reviewed by an admin user before it is available for everyone. You can also have a maximum of 5 clips under review before you can post a new one. This prevents a malicious user from filling up our backend with useless content.

5.1 Responsive design

It is important to consider different screen sizes and layouts if the aim is to have a web-based application, that is meant to work in a browser. There are different people with different taste. Some like to use applications with their computer on a big screen while others may prefer to use their smartphone. This means that all screen sizes and aspect ratios must be supported.

When smartphones became common and people started to browse the web with them, designers started to make mobile friendly websites. A mobile friendly website is a regular website that squeezes all the functionality on a mobile display. This was a good step

towards a mobile world, but the user still has to pinch or double tap to zoom in on the elements to be able to read them. [11.]

In 2018, over half of the searches made online were made with a smartphone. That means that no matter what type of website you are building, mobile users must be paid attention to. That is why developers and designers talk about “mobile first”-method when building a website. Which means that the mobile user comes before anyone else in the design process. [12.]

Nowadays responsive design is the most common way to design a modern web-based user interface. Responsive design means that the user interface is able to recognize the device it is on and adjust to its screen size. Functionality of a website can also be controlled with responsive design. If the mobile and desktop versions of our application are to be compared, we can see that the header or top bar changes when the screen size changes. On a big screen it is filled with buttons, but the mobile version hides the buttons behind a hamburger menu button (Image 7).

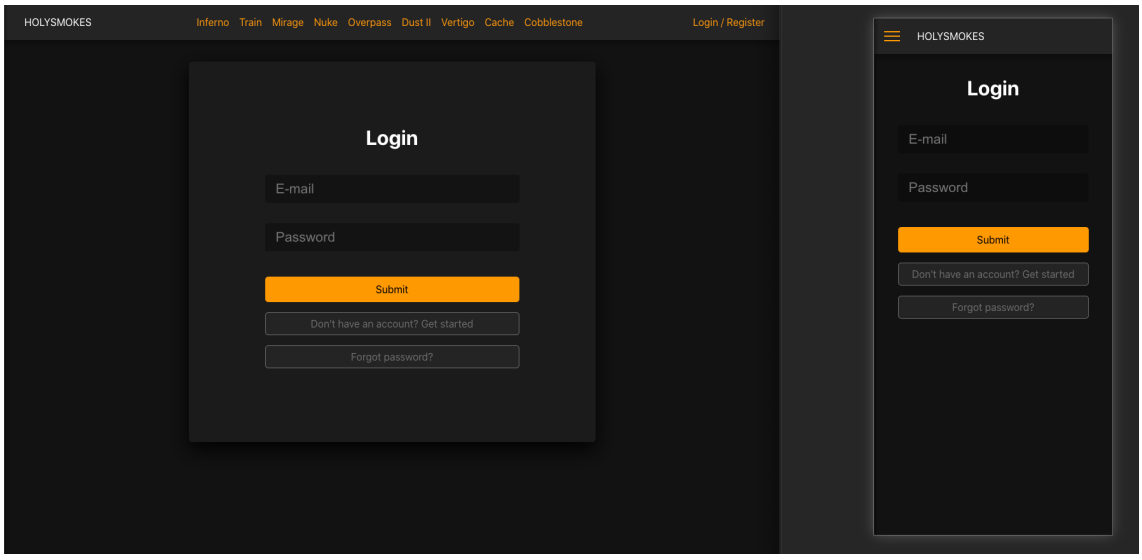


Figure 5. Comparison of the two header types. Here you can see the mobile version’s hamburger icon replacing the button row that is visible on a larger screen.

The responsiveness is made possible thanks to modern CSS: media queries. A media query consists of a media and of one or more prerequisites (Listing 1). If the prerequisites are met, the rules of that media are applied. With these you can create infinite rules which the user interface will obey.

```
@media only screen and (min-width: 767px) {
     h1 {
          color: black
     }
}
```

Listing 3. A simple media query where the color of Heading 1 is changed to black when the size of the browser window is at least 767 pixels wide

5.2 Design elements

The application takes form as an online video platform. The design of the application takes much inspiration from other online video platforms such as YouTube.

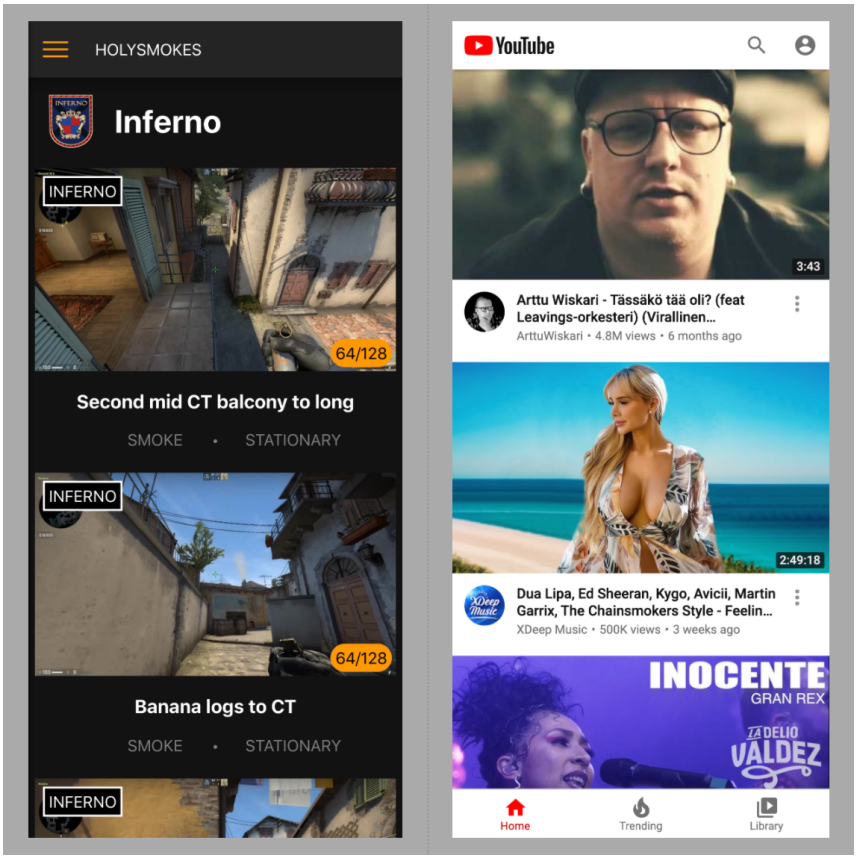


Figure 6. Holysmokes and YouTube mobile feed compared

As you can see (Figure 6), the wheel does not need to be reinvented. The purpose is to somewhat follow in the footsteps of the king of online video platforms. On the mobile version we mimic the following two trends that YouTube has. There are no clips next to

each other, just a vertical feed of infinite videos which can be browsed by scrolling. An individual video can be opened up just as in YouTube.

YouTube shows the length of the video in the corner of the thumbnail. There is no need to show that in our case as the clips that are uploaded are shorter than 20 seconds. What is shown in the corner is the important stuff that the player wants to see. A player wants to know if the specific technique works on the server that he is playing on. What is shown in the corner is viable information to the player as there are two types of servers. 64 and 128 tick servers. In-game objects behave a little differently on these two server types. That is why we need the uploader to tell which server environment is required for the type of technique.

5.3 User research

Some of the design choices were formed with the help of potential users. Google forms was used to conduct a user interview survey. An amount of 28 responses were formed at the time of writing. The variety of skill between the players who answered ranged from total beginners to an almost professional level.

The game in its competitive level is fast paced and if a potential user wants to check up on a technique mid game, he should be able browse through content as soon as he lands on the page. There is no need to authorize yourself if you are not planning to provide content, thus making the application ready to use immediately.

The plan is for the uploaded videos to be quick and informational, so we asked players about the duration of a perfect video about the content. Majority of the responses stated that 10 seconds is the optimal maximum length for an in-game smoke grenade throwing video. Some responded with 15 seconds but only one response thought that 5 seconds would be the optimal maximum length of the videos. The 10 second limit makes the user experience marginally faster when browsing through content and this way we also minimize in storage space per video. Manually testing of this led to a maximum length of 20 seconds, because 10 to 15 seconds was not enough in some cases.

5.4 Usability testing

Usability testing was performed on multiple users. Half of the test participants were on mobile while the other half were using the desktop version. They were given the URL to the first version of the site so that they could access it via the web. This was also a test to see how well the hosting service would do. When the user landed on the page, they were given a list of tasks to complete. These tasks included:

- Create a user and login
- Browse through at least 5 different video clips
- Upload a video clip of your choice (didn't have to be game related)
- Request a new password
- Logout

The upload task was only completed by a couple of people due to not having any short video clips on their device. The other tasks were done by everyone. Almost all of the tasks were completed seamlessly except a couple of them.

One of the tasks that proved to be difficult and almost everybody had something negative to say about it. It was the task of browsing through video clips. This felt difficult since the users felt that there was no way of returning from a video clip after they clicked and opened it. This is true since the only way to back from a clip is to use the dedicated back button of the browser the user is using. This was a severe problem since the aim is for the user to browse through multiple clips when he lands on the site. Opening a video clip is also an opportunity to show an ad an generate income. Many of the users suggested that the video clip page should open in a modal and not in its own page.

The other task that had some issues was the upload task. This was not nearly as bad as the other problem. Some of the users complained about not seeing an upload counter. This usually led the user to being confused and they started to refresh the page, thinking that there was some problem on their end.

5.5 Dark UI

A dark user interface is a no brainer in this case. CS: GO players like to setup their brightness and contrast to a very high level to make it easier to spot enemies in dark places. We will have a simple dark design because when a user switches to a browser and opens this application, we do not want to blind them with a light-colored theme. Another use case is that if a user has the application open on a secondary monitor, a dark theme helps them to focus on the main monitor and prevents blinding them from the side.

6 Marketing

6.1 Advertising Revenue

The main business plan of the software is to generate advertising revenue. That means that the main source of income would be to earn money by displaying advertisements on the site. There are many advertisement platforms provided by huge companies such as: Google, Facebook, Amazon and Microsoft.

Our application will go with Googles AdSense since our backend is already on Googles platform. That way we can fully embrace the Google ecosystem. AdSense is a service that Google provides for generating income for website publishers. AdSense matches the ad content based on your website and visitors. Google handles all the relationships with advertisers and takes 32% of the money made from your site's ads. There are four types of ways of getting paid for the ads that are displayed. These are:

- Cost per click
- Cost per thousand impressions
- Active view cost per thousand impressions
- Cost per engagement

Figure 7. Visualization of how Ad Rank is calculated. [13.]

Google has a way of determining which ads to show on your website. This tool is called Ad Rank (Figure 7). Ad Rank is a value used and determine ad position or whether not to show ads at all. Ad Rank is calculated by multiplying your cost per click value with a quality score which is a prediction of clickthrough rates and ad relevance. Quality score takes keywords also into account. [13.]

6.2 Fighting Ad Blockers

AdBlock can be an inhibition of ad revenue. According to Doc Searls' blog post "Is ad blocking past 2 million worldwide?", 11% of all devices connected to the world wide web are using some kind software to block ads. [14.] The majority of our userbase consists of young males and according to Joseph Johnsons online survey in the UK, these results show that around 40% of people from 15 to 35 year olds use ad blockers. [15.] Ad blockers need to be taken into consideration if maximum income with ad revenue is to be created.

There are different approaches to dealing with adblockers. Some say that you should block all content from them before they are allowed to interact with the website. Others think that they should be ignored and some just want to notify the incoming user with non-hostile message to disable the ad blocker, without closing the site from them. Our approach is going to be the one notifying the user about his ad blocker but not forcing them to close it, because according do Mathew Boughton's news article, 74% of US ad block users leave sites where content is hidden before you turn off your ad blocker. [16.]

7 Results

In this study, some UI elements from big online video platforms were mimicked, because they share some of the same UX and UI principles. User interview (Figure 8) and testing led to small changes and improvements that I had not thought of. The following observations were made as a result of this research. The application's user experience was working for the users. Many of the users liked the look of the first version and only had a few issues that are easy to fix. They also gave valuable insight on how they would like the UX to work in some of the cases that did not feel so natural for them. Many shared the same opinion on these things. Many of them also praised the simplicity of the application, which was a huge target of mine, since I wanted to create a simple and easy to use application that could be interacted with as soon as you land on the page. There are also some huge advantages on the platform that the competitors are missing, and that could lead into success. I also found out that I should prompt ad block users to remove their ad block to support the creator and not hide and disable content from them. Overall, I think the process was a success.

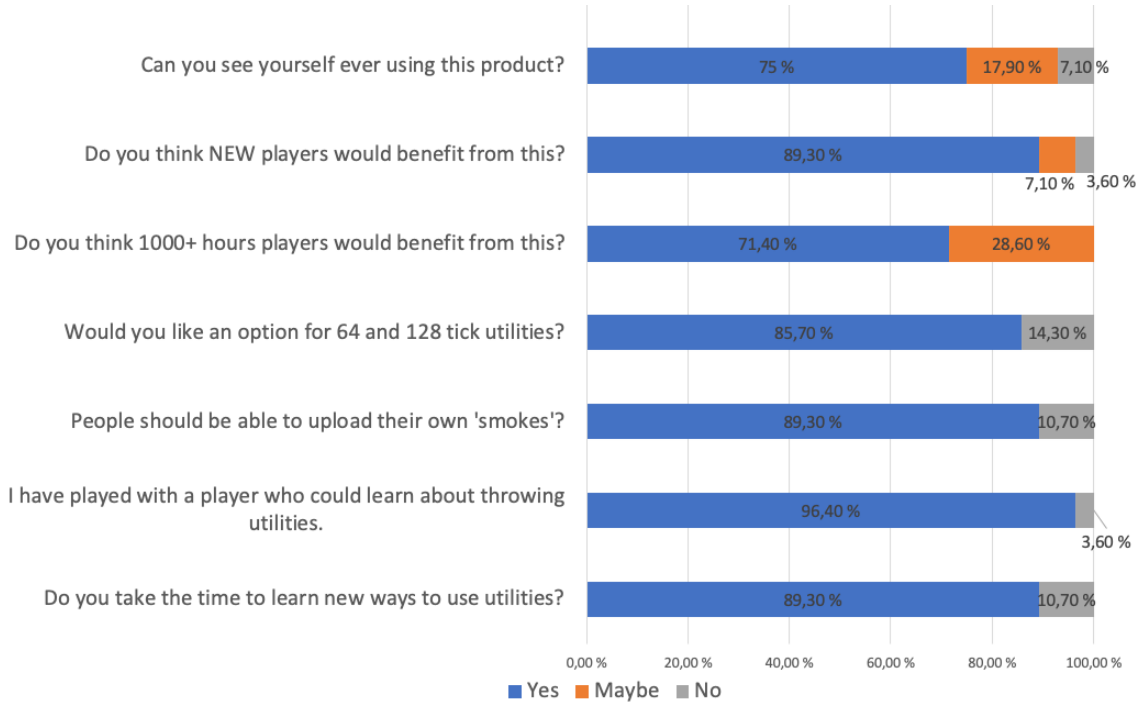


Figure 8. Results for some of the questions of the user interview

8 Conclusion

The purpose of this thesis was to develop a dynamic and video-based learning platform to the web for Counter Strike: Global Offensive players. The aim of this project was to create a web application with the help of user research. Web application was chosen since the aim is to reach desktop and mobile users. The platform focused on one specific element of the game and thus limiting the complexity of it. This gameplay element is the ability to throw utilities to specific tactical positions such as a smoke grenade to cause a diversion or to block vision.

The frontend was designed with the help of a user interview which had 28 participants that were all players of CS: GO. The applications frontend has a fully responsive UI built with Reactjs and after some usability testing with potential users, the UX proved to be good after a little bit of polishing. The applications code is hosted on GitHub and is continuously delivered to Firebase Hosting service. The full scope of this was not met as it is not ready for public use and there are no ads on the site as of yet.

The application is meant to be released in the coming months as it is not yet ready for the public use. After integrating an ad service and Googles reCAPTCHA it should be ready. The UI needs a little fine tuning, but it is not the worst of problems since that can be polished until forever. After release, the possible PWA future for the application is to be considered. Windows 10 supports PWA and that is one of the main reasons the applications future has PWA on its path. Almost all of CS: GO players have Windows 10 as their operation system since it is the go to OS when building or buying a gaming PC.

References

1 A Guide to Mobile App Development: Web vs. Native vs. Hybrid. Clearbridge. Online material <https://clearbridgemobile.com/mobile-app-development-native-vs-web-vs-hybrid/> Read 30.10.2020.

2 Progressive Web Apps. Google. Online material <https://developers.google.com/web/progressive-web-apps> Read 10.10.2019.

3 Ronsse, Johan. 2020. Apple doesn't care about your PWA. Online material <https://johanronsse.be/2020/08/30/apple-doesnt-care-about-your-pwa-and-a-little-rant-about-holding-back-the-future-of-computing/> Updated 30.8.2020. Read 10.10.2020

4 Macoveiciuc, Andreea. 2020. Why Responsive Design is Important and Google Approved. Online material <https://freshsparks.com/why-responsive-design-is-important/> Updated 1.8.2019. Read 10.10.2020.

5 Garner, Bennet. 2019. The Key Concepts You Need to Understand About React. Online material <https://levelup.gitconnected.com/new-to-react-you-need-to-understand-these-key-concepts-before-anything-else-2247efc1eaac/> Updated 29.1.2019. Read 30.10.2020.

6 Cloud Storage. 2020. Google. Online material <https://firebase.google.com/docs/storage/> Read 10.8.2020.

7 Choose a Database: Cloud Firestore or Realtime Database. 2020. Google. <https://firebase.google.com/docs/database/rtdb-vs-firestore/>

8 Firebase Hosting. 2020. Google. <https://firebase.google.com/docs/hosting/> Read 10.8.2020

9 Huddleston, Tom Jr. 2018. How this 33-year old college dropout co-founded GitHub, which just sold to Microsoft for $7.5 billion. <https://www.cnbc.com/2018/06/04/chris-wanstrath-co-founded-github-which-microsoft-bought-for-billions.html/> Updated 4.6.2018. Read 1.11.2020.

10 Octoverse. 2020. GitHub. <https://octoverse.github.com/> Read 1.10.2020.

11 Gregory, Sonia. 2020. Why Responsive Design is Important and Google Approved. <https://octoverse.github.com/> Updated 1.8.2020. Read 15.10.2020.

12 Lynkova, Darina. 2020. 35+ Website Design Industry Statistics: All You Need to Know in 2020. <https://techjury.net/stats-about/website-design-industry> Read 15.10.2020. Updated 11.9.2020.

13 Lofgren, Lars. 2020. How to Add Adsense to Your Website. <https://www.quick-sprout.com/how-to-add-adsense-to-your-website/> Read 19.9.2020. Updated 18.3.2019.

14 Searls, Doc. Is ad blocking past 2 billion worldwide? <https://blogs.harvard.edu/doc/2019/03/23/2billion/> Updated 23.3.2019. Read 19.9.2020.

15 Share of individuals who use add blocking software in the United Kingdom (UK) in 2020, by age group. Statista. <https://www.statista.com/statistics/875604/ad-blocker-users-in-the-united-kingdom-by-age-group/> Updated 12.9.2019. Read 19.9.2020

16 Broughton, Mathew. The Fight Against Ad Blocking: Why Blocking the Blockers Is Not Enough. <https://www.exchangewire.com/blog/2019/06/12/the-fight-against-ad-blocking-why-blocking-the-blockers-is-not-enough/> Updated. 12.6.2019. Read 19.9.2020.

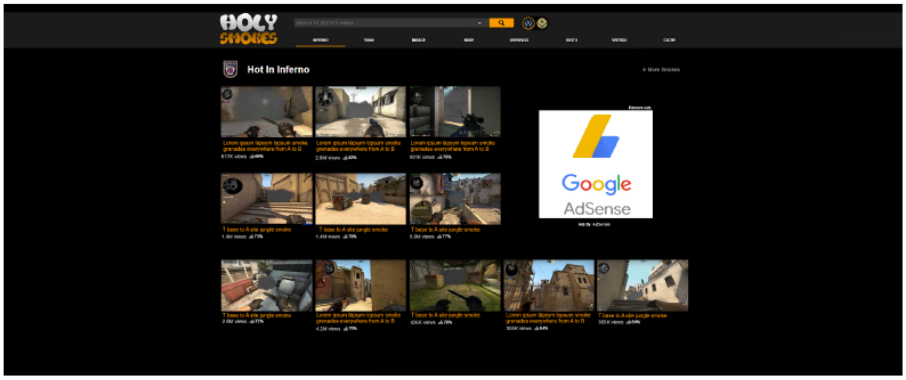
Holysmokes user interview

Holysmokes user interview

Holysmokes is aiming to be a dynamic and fast platform for CS:GO players to learn how make the most out of your in-game utilities such as smoke grenades. The site consist of user created videos of throwing grenades in-game. This question sheet has questions about your CS:GO gaming habits as well as some questions regarding the application.

* ~~Required~~

Holysmokes 'mockup' layout



1. Do you take the time to learn new ways to use utilities?

   Mark only one oval.

   - ◯ Yes
   - ◯ No

2. Best way/place to learn a new smoke.

Check all that apply.

- [ ] Practicing alone in-game
- [ ] When playing a fun match
- [ ] When playing a competitive match
- [ ] Clips from Youtube/Twitch etc.
- [ ] I don't practice
- [ ] https://sothatwemaybefree.com/
- [ ] https://www.csgonades.com/

Other: [ ] _____

3. I have played with a player (who I know IRL) who could learn a thing or two about throwing utilities.

Mark only one oval.

- ◯ Yes
- ◯ No

4. I have played with a player (random) who could learn a thing or two about throwing utilities.

Mark only one oval.

- ◯ Yes
- ◯ No

5. Optimal MAX length for one smoke throw video?

Mark only one oval.

- ◯ 5s
- ◯ 10s
- ◯ 15s
- ◯ 20s

6. Do you think people should be able to upload their own 'smokes'?

Mark only one oval.

- ◯ Yes
- ◯ No

7. What about the maps?

Mark only one oval.

- ◯ Competitive map pool only
- ◯ Competitive maps + some of the most played maps.
- ◯ All maps
- ◯ Other: _____

8. There should be an option to hide advanced smokes (jump throw bind, skybox bounce etc.)

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Other: _____

9. Would you like an option for 64 and 128 tick utilties? (64 is the official server tickrate while 128 is used in advanced matches, has an effect on the throw)

Mark only one oval.

- ◯ Yes
- ◯ No

10. Do you think 1000+ hours players would benefit from this?

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

11. Do you think NEW players would benefit from this?

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

12. Can you see yourself ever using this product? *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

13. On what device would you use holysmokes? *

Check all that apply.

- ☐ Desktop
- ☐ Mobile
- ☐ Tablet

14. Would you add holysmokes.gg to your phones home menu as an app icon? *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

15. I am interested of becoming an admin on holysmokes.gg

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Maybe

16. A user must be registered before he can upload a smoke/nade. Every upload is being reviewed by an admin so that we wouldn't have duplicates or awful videos. What do you think? *

Mark only one oval.

- ◯ Sounds good.
- ◯ Other: _____

17. If you have further development ideas please provide them here

_____

_____

_____

_____

_____